

MADCAP FLARE 2024

Branding

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
9171 Towne Center Drive, Suite 335
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Introduction	5
--------------------	---

CHAPTER 2

General Information for Branding	8
Branding Stylesheets	9
Branding and CSS Variables	11

CHAPTER 3

Main Activities for Branding	20
Creating Branding for a New Project	21
Downloading Templates	22
Opening the Branding Editor	23

CHAPTER 4

Other Activities for Branding	24
Adding a Branding Stylesheet	25
Associating a Branding Stylesheet at the Project Level	27
Associating a Branding Stylesheet at the Target Level	28
Sharing Branding Files With Others	29

CHAPTER 5

Custom Branding Project Templates	31
Step 1: Create or Open a Project Template	33
Step 2: Identify the Main Branding Elements and Values of the Project	34
Step 3: Rename and Move Image Files	36
Step 4: Copy Branding CSS Variables From a Factory Template	38
Step 5: Replace Values in Stylesheets	44
Step 6: Replace Values in Skins	46
Step 7: Replace Images in Template Pages	48
Step 8: Edit Page Layouts	50
Step 9: Copy and Paste Preview Folder	54
Step 10: Edit the Branding.css Preview File	57
Step 11: Edit the Branding-Template.xml Preview File	59
Step 12: Replace the Image Files in the Preview Folder	61
Step 13: Edit the Preview HTML and CSS Files	63
Step 14: Edit the FLPRJ File	73
Step 15: Remove Unnecessary Folders	74
Step 16: Test It	75

APPENDIX

PDFs	77
Tutorials	77
Cheat Sheets	78
User Guides	79

CHAPTER 1

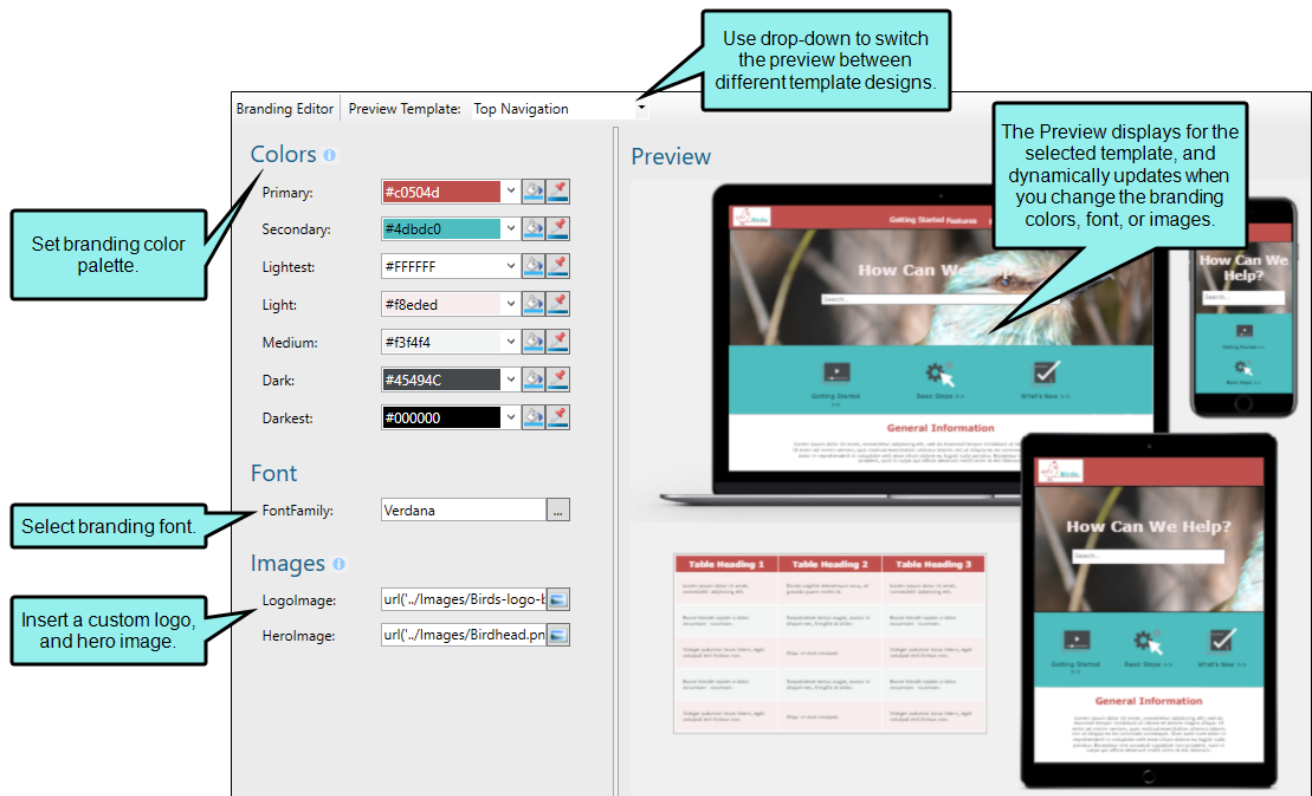
Introduction

Supported In:



This feature provides a convenient way to add branding to a project. Without much effort, you can set common branding elements (e.g., logo, hero image, font, color palette) to match the output with your company's brand.

- **If you are setting up a new project or are new to Flare** You can set up branding when creating a new project using the Start New Project Wizard. The advantage of doing this is that your basic branding design is established from the get-go. Alternatively, you can download a free project template (from MadCap Software's website), and then add your content and branding (via the Branding Editor).
- **If you already have an established project in Flare** You can leverage the benefits of the branding feature. For example, you can add a new branding stylesheet to a project or create and save a branding template to share with others.



General Information

- "Branding Stylesheets" on page 9
- "Branding and CSS Variables" on page 11

Main Activities

- "Creating Branding for a New Project" on page 21
- "Downloading Templates" on page 22
- "Opening the Branding Editor" on page 23

Other Activities

- "Adding a Branding Stylesheet" on page 25
- "Associating a Branding Stylesheet at the Project Level" on page 27
- "Associating a Branding Stylesheet at the Target Level" on page 28

- "Sharing Branding Files With Others" on page 29
- "Custom Branding Project Templates" on page 31

✔ **TIP** It is wise to add branding to your technical documentation or eLearning output because it reinforces your organization's identity and personality. A strong brand can stand out in a sea of competition. Good branding can enhance client and employee loyalty and build trust and credibility in your company or product.

General Information for Branding

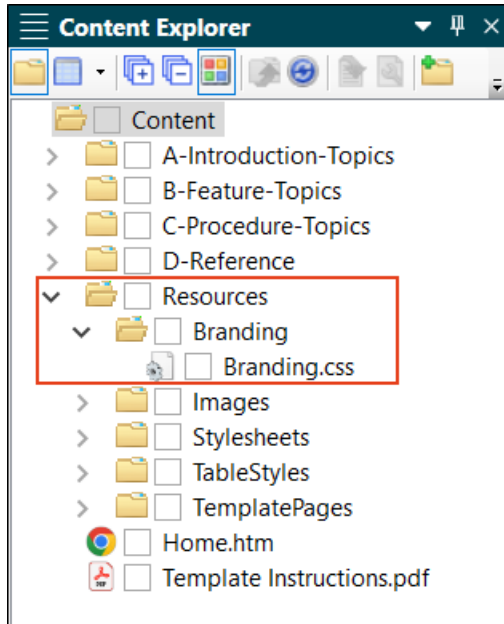
There are various pieces of general information you should know if you plan to use this feature.

This chapter discusses the following:

- Branding Stylesheets 9
- Branding and CSS Variables 11

I Branding Stylesheets

When a project is created a new branding file is automatically added that specifically identifies values for the branding elements. This file is a cascading stylesheet. Its default location is Content Explorer > Resources > Branding (Branding.css), but you can store the stylesheet anywhere. If a custom branding scheme is not defined, the stylesheet is still generated but with default branding values.



✓ **TIP** The branding stylesheet is an ideal place to consolidate CSS variables across your project into a single file (i.e., the Branding.css file). If you have CSS variables already set up in another stylesheet you can keep them there.

Branding Stylesheet Order of Precedence

When using multiple stylesheets with conflicting settings, the branding stylesheet has the lowest precedence.

FLARE INTERFACE

1. Local stylesheet associated with micro content file
2. Local stylesheet associated with topics
3. Primary stylesheet (styles from only one primary stylesheet can be used)
 - a. *Primary* Target
 - b. Project
4. Branding stylesheet

OUTPUT

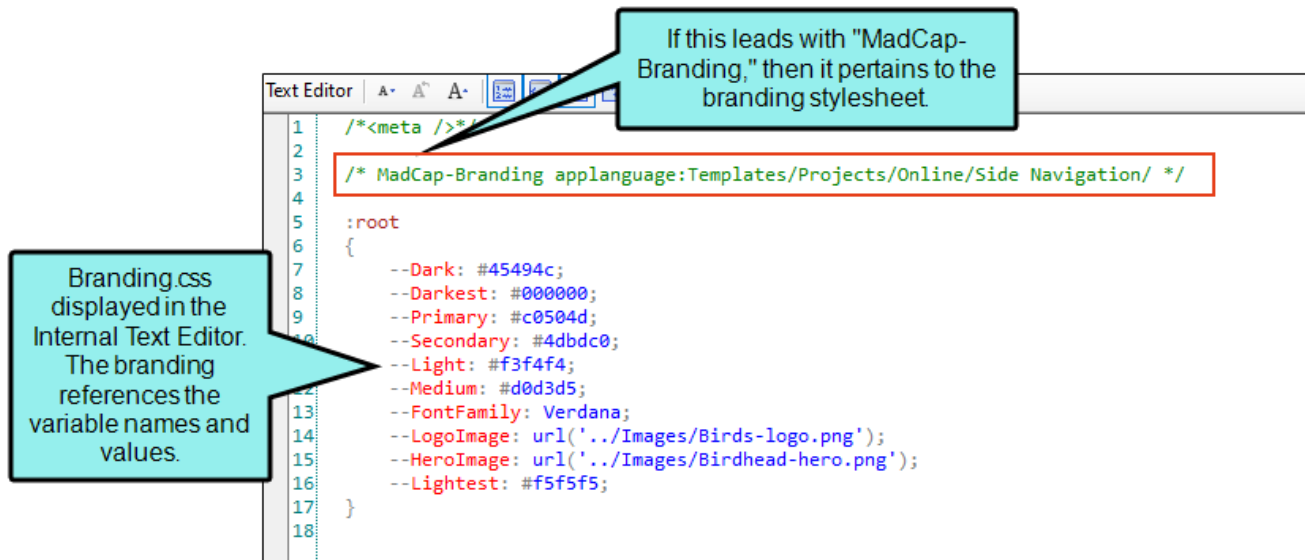
1. Local stylesheet associated with micro content file
2. Local stylesheet associated with topics
3. Local stylesheet associated with template pages
4. Primary stylesheet (styles from only one primary stylesheet can be used)
 - a. *Any* Target
 - b. Project
5. Branding stylesheet



NOTE If branding values are assigned in the Start New Project Wizard or after, those settings would propagate to other stylesheets of higher priority in the project. But you can override the branding styles in stylesheets with higher precedence.

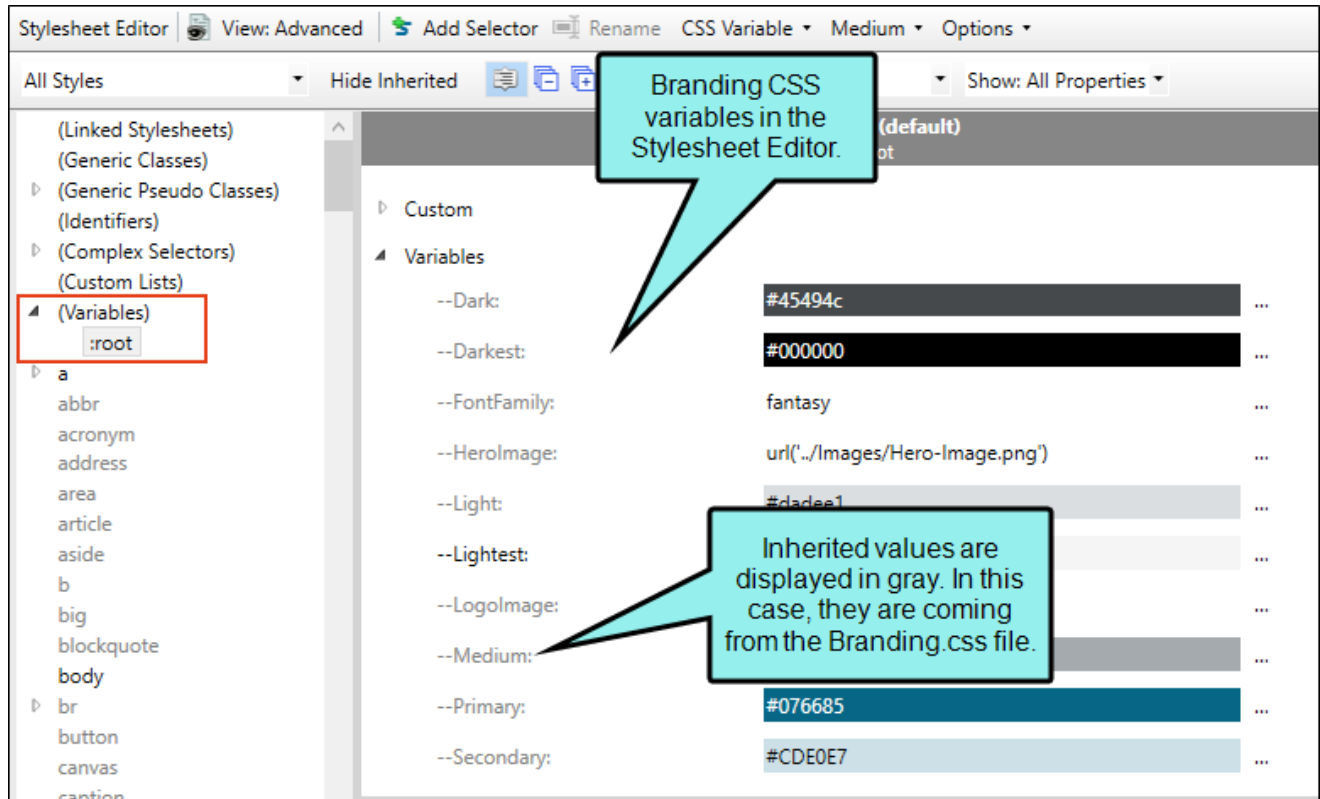
I Branding and CSS Variables

CSS variables are the foundation of branding functionality. The Start New Project Wizard and the Branding Editor group together certain CSS variables (e.g., logo, hero image, font, color palette) that relate to branding. The branding values you identify in the Branding.css file are those associated with those CSS variables in the project. If you start your project with the wizard, the CSS variables are automatically linked to various places throughout the project where they point to the branding stylesheet.



Managing CSS Variable Values

Use the Branding Editor to manage your project's CSS variable values. If you open the regular stylesheet, the branding CSS variables are shown as inherited (i.e., displayed in gray). If you change the values in the Branding.css file, those changes will automatically be shown in the regular stylesheet.

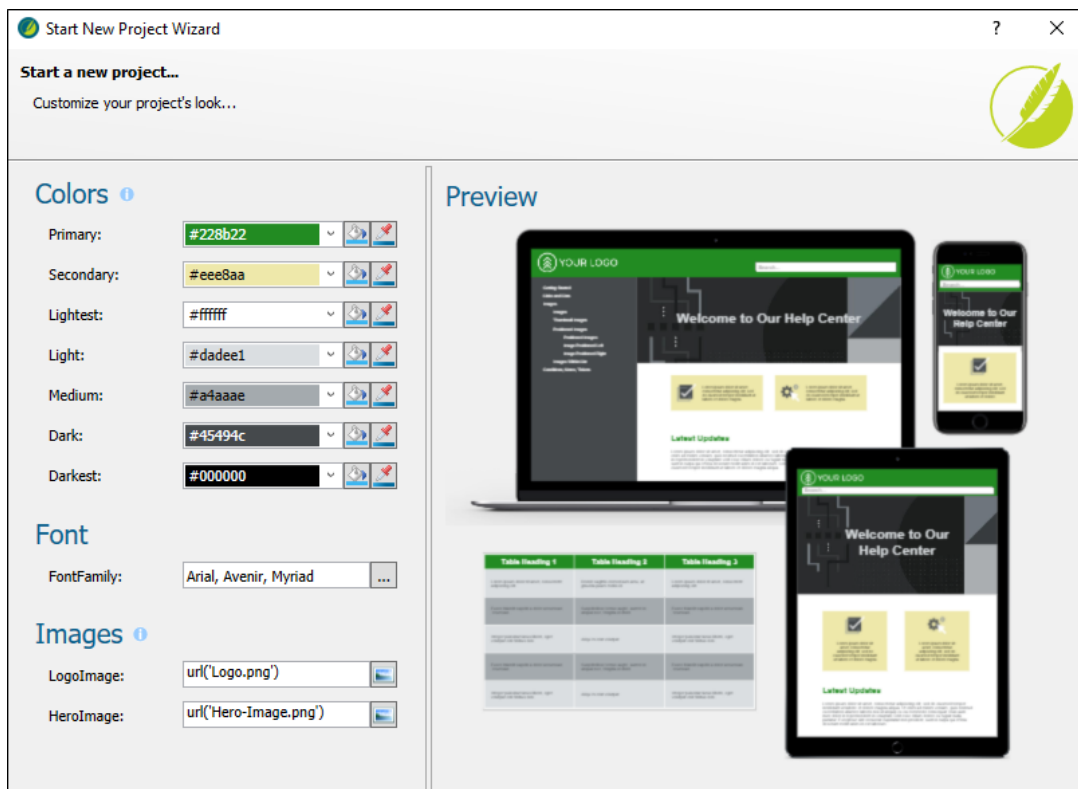


Updating Values in the Regular Stylesheet

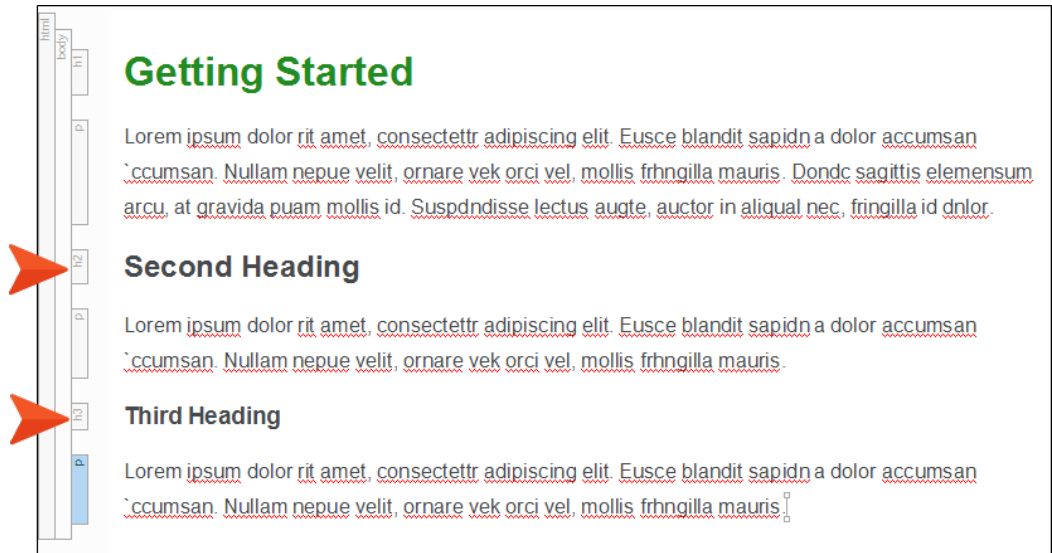
You can update a value for any of the CSS variables in the regular stylesheet, but then related changes made in the future to the Branding.css file will not be shown in the regular stylesheet because you have overridden them. Since the regular stylesheet (e.g., Styles.css) is more specific in the order of cascading stylesheets, the Styles.css would "win" precedence.

If you want to select a CSS variable value for a style with the regular stylesheet, be sure to choose a variable from the Color Picker dialog. That way it points to the branding stylesheet where the variable is already defined for the project.

☆ **EXAMPLE** You have created a new project based on the Side Navigation template with your primary and secondary branding colors. It looks like this:

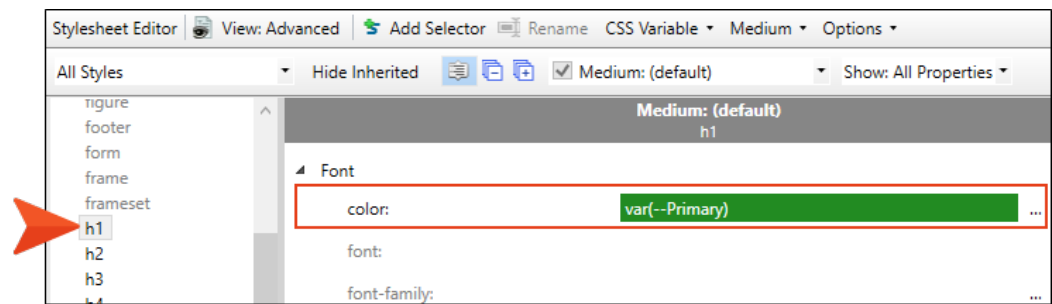


- ☆ You open a topic and notice the primary color shows for the h1 element. That is what you wanted. As you add h2 and h3 elements you notice they are not displaying as your primary branding color. You wanted those headings to be green as well.



How do you change other headings to show the branding color you want? You can update individual styles in the regular stylesheet.

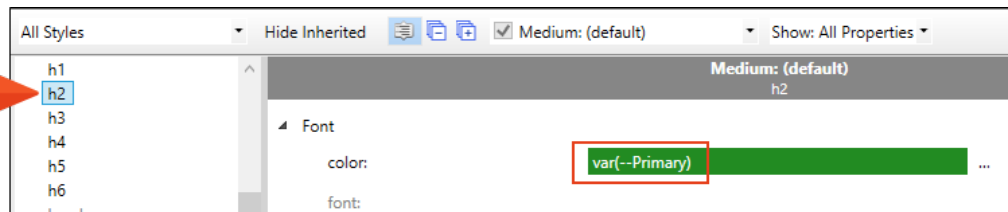
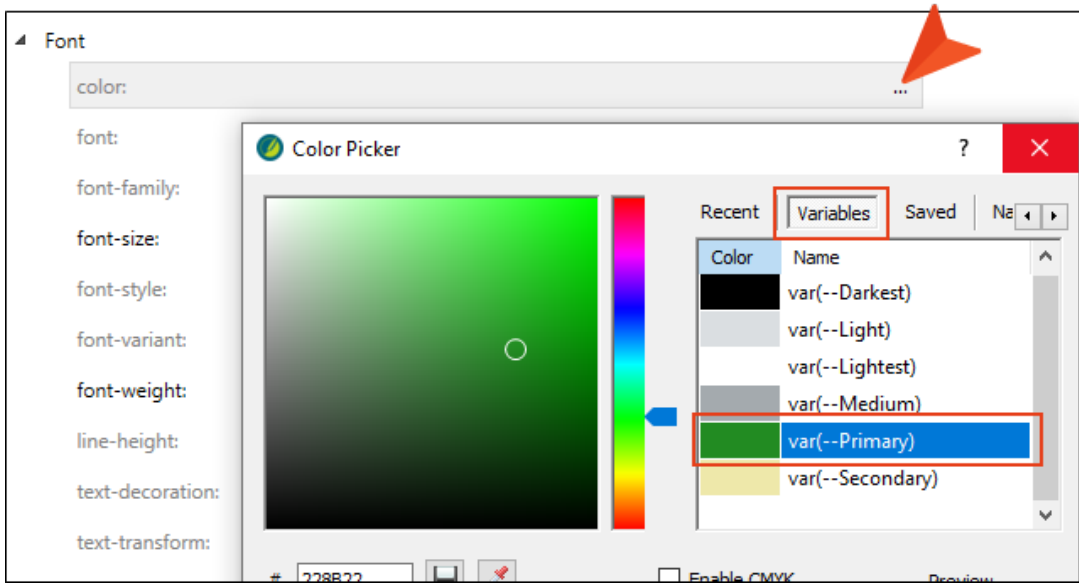
In the Content Explorer, expand **Resources > Stylesheets**, and double-click the **Styles.css** file. In the Advanced view of the stylesheet, locate the h1 element, and view its font properties. You can see that it uses the CSS variable for referencing the primary color.



- ☆ If you locate the h2 element, and view its font properties, notice that the color is not specifically identified.



From the **Color** field, click the horizontal three dot menu and select **More colors** for the Color Picker dialog. Select the **Variables** tab and then the primary branding color.



☆ Do the same for h3, and then **Save** the stylesheet.

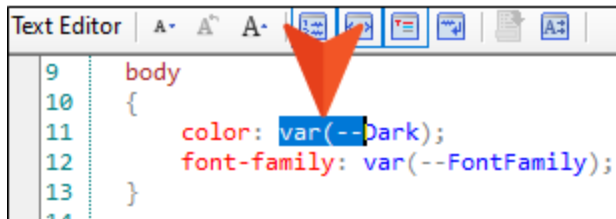
If you go back to the topic, you can see the desired results. (Or, you can see the updated style in the **Style** drop-down.)



What's Noteworthy?

✔ **TIP** When interacting with the Start New Project Wizard or the Branding Editor, you can see how the branding CSS variables are used as they display in the Preview area. But what if you want to find out exactly where in the project CSS variables are used? Do the following.

1. Open the regular stylesheet in the Internal Text Editor.
2. Locate a CSS variable. For example, `var(--Dark)`.
3. Select the `var(--` part of the CSS variable.



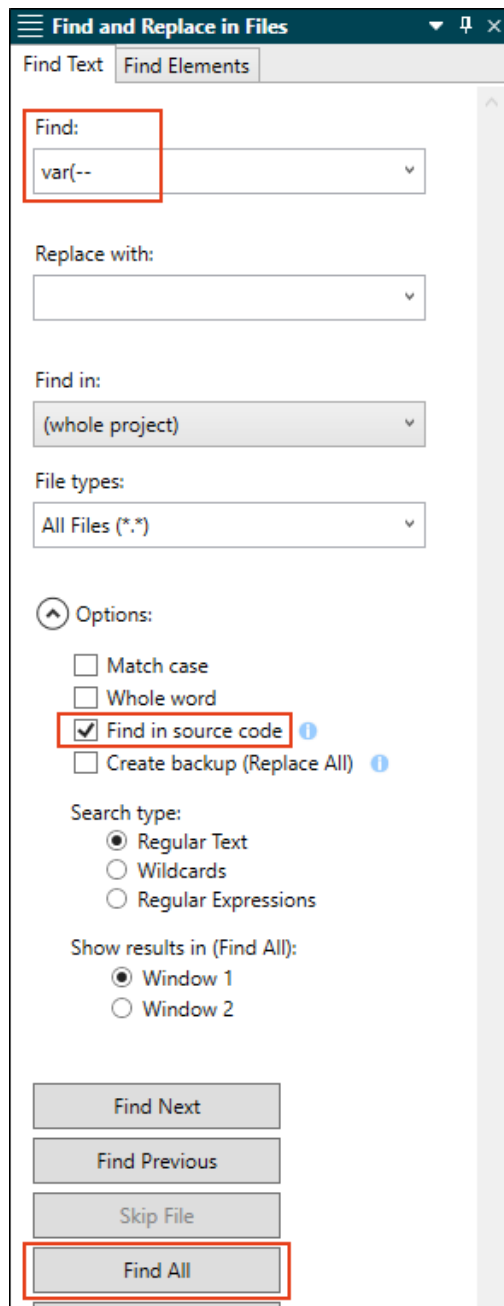
The screenshot shows a text editor window titled "Text Editor" with a toolbar. The code is as follows:

```
9  body
10 {
11     color: var(--Dark);
12     font-family: var(--FontFamily);
13 }
```

The text `var(--` on line 11 is highlighted in blue. A red arrow points from the toolbar down to this selection.

4. Open the Find and Replace window pane. The selected text automatically populates the **Find Text** tab's **Find** field.
5. In the **Options** field, select **Find in source code**.

6. Click Find All.



- ✓ The Find Results window lists all the files (e.g., regular stylesheet, table stylesheet, skin) where CSS variables are used.

File	Context	Folder
Styles.css	...body { color: var(--Dark); font-...	C:_Projects\Brand101\Content\Resources\Stylesheets\
Styles.css	...rk); font-family: var(--FontFamily);...	C:_Projects\Brand101\Content\Resources\Stylesheets\

You can open any of those listed files in the Internal Text Editor. If you use the Find widget to perform a search in the editor for var(--, you can see what styles in the file are using certain CSS variables, and where you have CSS variable references set.

```
238 {  
239   text-align: center;  
240 }  
241  
242 div.topic-hero  
243 {  
244   padding: 5%;  
245   background-image: var(--HeroImage);  
246   background-size: cover;  
247   background-attachment: scroll;  
248 }  
249  
250 div.topic-hero h1  
251 {  
252   color: var(--White);  
253   font-size: 48pt;  
254   text-align: center;  
255 }
```

CHAPTER 3

Main Activities for Branding

Some activities are particularly common and important when it comes to this feature.

This chapter discusses the following:

- Creating Branding for a New Project21
- Downloading Templates 22
- Opening the Branding Editor 23


I Creating Branding for a New Project

You can set up branding when creating a new project using the Start New Project Wizard. The advantage of doing this is that your basic branding design is established from the get-go. Alternatively, you can download a free project template (from MadCap Software's website), and then add your content and branding (via the Branding Editor).

The branding screen in the Start New Project Wizard is essentially the same thing as the Branding Editor that you can open anytime as a standalone editor after a project is created.

How to Add Branding in the Start New Project Wizard

1. When you first open Flare, you will see the Start Page in the middle of the interface. On the Start Page, click **New Project**. The Start New Project Wizard opens.
2. In the **Project name** field, name your project.
3. (Optional) In the **Project folder** field, 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. From the **New from template** area, select a template.
6. Click **Next**.
7. (Optional) Depending on the template selected, choose branding colors, a font family, logo, and hero image to apply to your project. This will give your project a custom look and feel from the start. These options can be changed later in the Branding Editor.

 **NOTE** If you use a factory template or a custom preview project template, the branding page displays. If you select files from a **New from existing** or a **New from import** source, the branding screen will be bypassed.

8. Click **Next**.
9. Select the primary target for your project.
10. Click **Finish**.

I Downloading Templates

You can create a new project by downloading a free ready-made template with a theme from MadCap Software's website.

If you are using the Start New Project Wizard, there is a screen that allows you to select a project template.

To download one from the website, click the **Download additional project templates** link in the wizard. Go to the website: www.madcapsoftware.com/downloads/madcap-flare-project-templates.

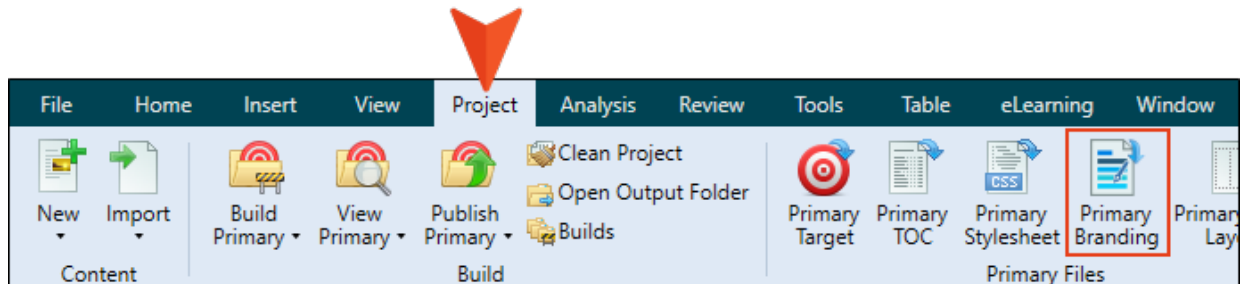
How to Download a Template

1. Download one of the project templates.
2. Extract the file(s) from the zip file to the location where you want the project to live.
3. Double-click the Flare project file (FLPRJ) to open the project.
4. Expand **Resources**, and double-click the **Branding.css** file.
5. In the Branding Editor, you can edit the colors, font, and images to match your own branding motif.

Opening the Branding Editor

Do one of the following, depending on the part of the user interface you are using:

- Ribbon Select Project > Primary Branding.



- Content Explorer
 1. Open the Content Explorer
 2. Expand the **Resources** folder, and then **Branding**.
 3. Double-click **Branding.css**.

CHAPTER 4

Other Activities for Branding

There are some additional tasks you might perform regarding this feature.

This chapter discusses the following:

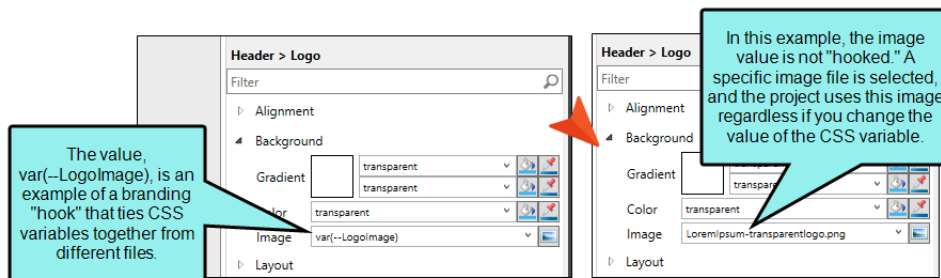
Adding a Branding Stylesheet	25
Associating a Branding Stylesheet at the Project Level	27
Associating a Branding Stylesheet at the Target Level	28
Sharing Branding Files With Others	29

Adding a Branding Stylesheet

You can add a branding stylesheet to a project. It will not include all the necessary connections for instant results. But it will tie the stylesheet to a project if you select it as your primary branding stylesheet. See "Associating a Branding Stylesheet at the Project Level" on page 27 and "Associating a Branding Stylesheet at the Target Level" on page 28.

NOTE For a branding file to have impact on a project there must be styles throughout the project that reference the CSS variables defined in the file. Simply adding a branding file to an existing project might not have an immediate effect because the project needs those styles that use branding variables as "hooks" for connecting it all together.

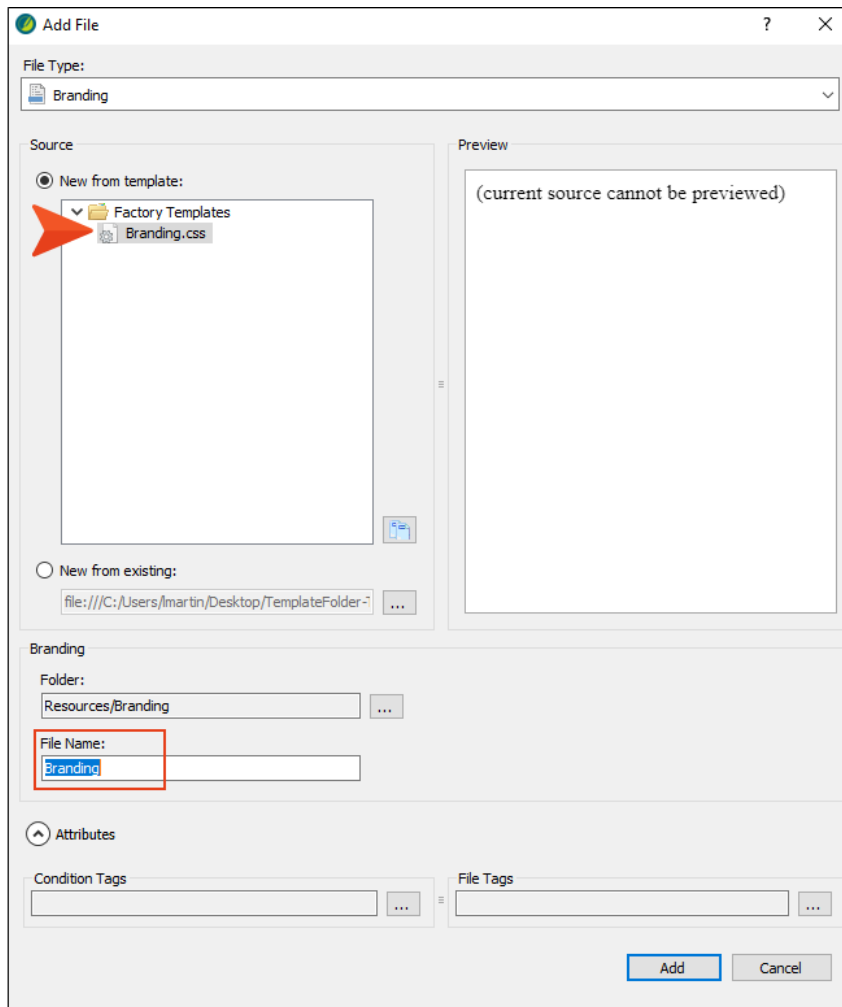
For example, the Header > Logo > Background > Image field in a skin will not use your logo selection from the Branding Editor unless that field is set to use the relevant CSS variable, *var(--LogoImage)*. Fortunately, this particular field is already set with that variable in Flare's factory project templates.



1. From the **Project** ribbon, select the **New** button (down arrow), and then select **Add Branding Stylesheet**.
2. In the Add File dialog, select the factory **Branding.css** file. Adding this file to your project provides it with basic branding connections for updating the main brand color (i.e., Primary), font, logo, and a couple complementary colors.
3. (Optional) In the **Folder** field, select a folder to save the stylesheet to.

TIP If your project does not include a Branding folder under the Content Explorer's Resources folder, it might be a good idea to create a new folder called Branding.

4. In the **File Name** field, type a name for the stylesheet.
5. Click **Add**.



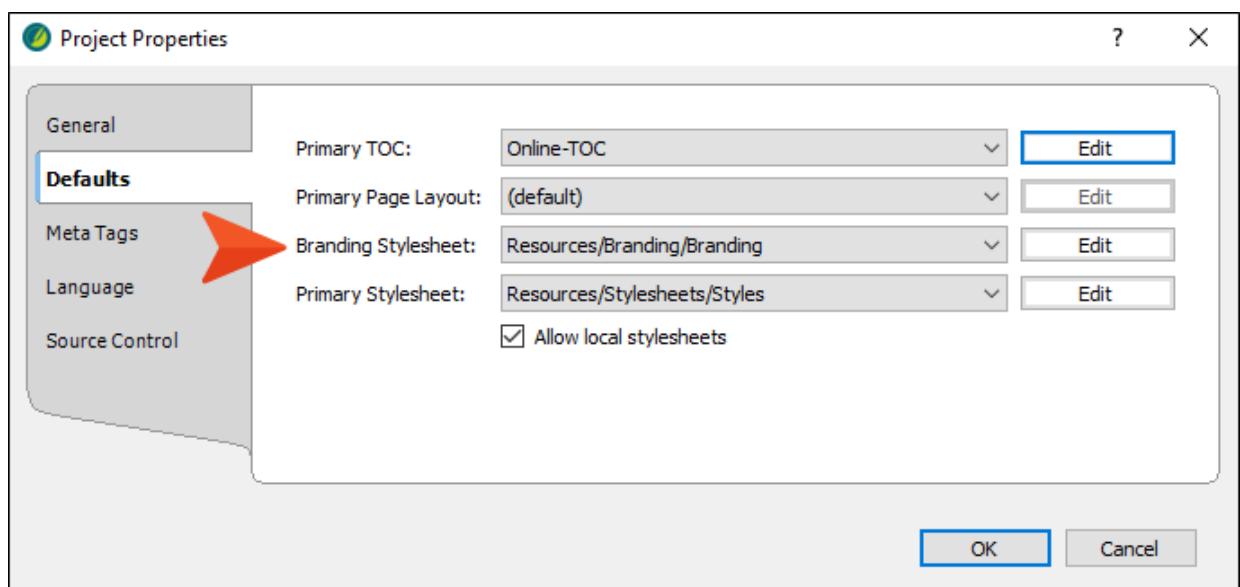
✔ **TIP** You can use the new branding stylesheet as a place to store and view all of your CSS variables. If you open the Branding.css file and the stylesheet containing existing CSS variables using the Internal Text Editor, you can cut and paste CSS variables from the stylesheet to the branding file.


Even though your CSS variables are located in a different stylesheet (i.e., Branding.css), when you open the regular stylesheet (or a skin), you will have access to those variables to insert them as property values for different selectors.

I Associating a Branding Stylesheet at the Project Level

When you add a new branding stylesheet, be sure to associate it as your primary branding stylesheet. Set the stylesheet at the project level for the same branding throughout the project. You can also set it at the target level.

1. From the **Project** ribbon, select **Project Properties**.
2. In Project Properties, select the **Defaults** tab (if not already selected).
3. From the **Branding Stylesheet** drop-down, select the stylesheet as your primary branding stylesheet.



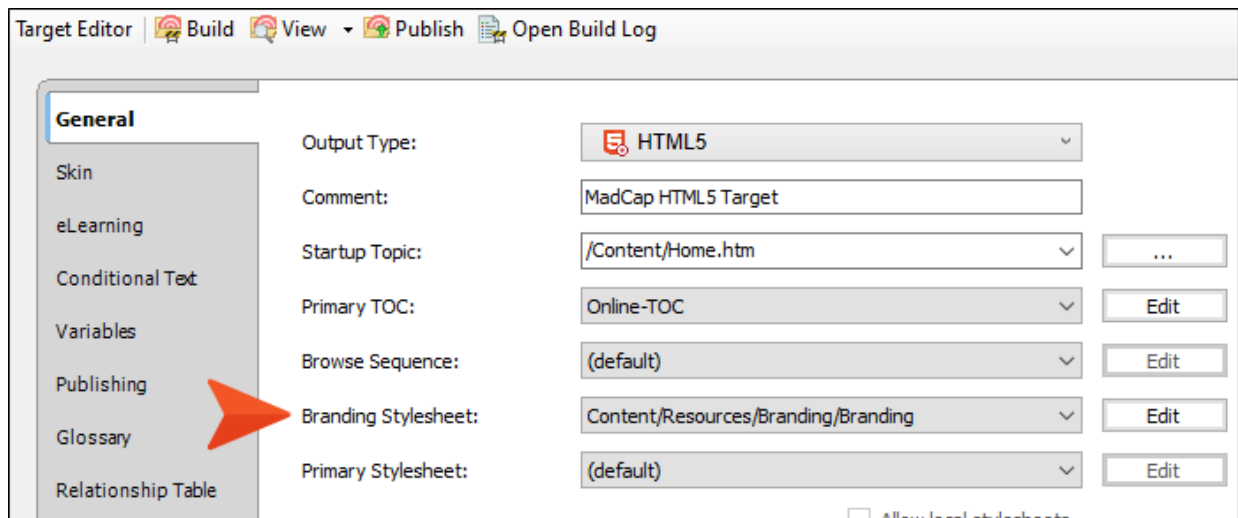
 **NOTE** Associating a branding stylesheet ties it to a project. From the **Project** ribbon, notice the **Primary Branding** option. If enabled, this opens the stylesheet that is set as the branding stylesheet for the project. (The ribbon option is disabled if there is no branding file, or if no branding stylesheet is set in the Project Properties or Target Editor.)

I Associating a Branding Stylesheet at the Target Level

When you add a new branding stylesheet, be sure to associate it as your primary branding stylesheet. If you set the stylesheet at the target level, it will have precedence on whatever is set at the project level.

How to Associate a Branding Stylesheet at the Target Level

1. From the Project Organizer, expand **Targets**, and double-click the target.
2. In the Target Editor, select the **General** tab (if not already selected).
3. From the **Branding Stylesheet** drop-down, select the stylesheet as your primary branding stylesheet.




NOTE Associating a branding stylesheet ties it to a project. From the **Project** ribbon, notice the **Primary Branding** option. If enabled, this opens the stylesheet that is set as the branding stylesheet for the project. (The ribbon option is disabled if there is no branding file, or if no branding stylesheet is set in the Project Properties or Target Editor.)

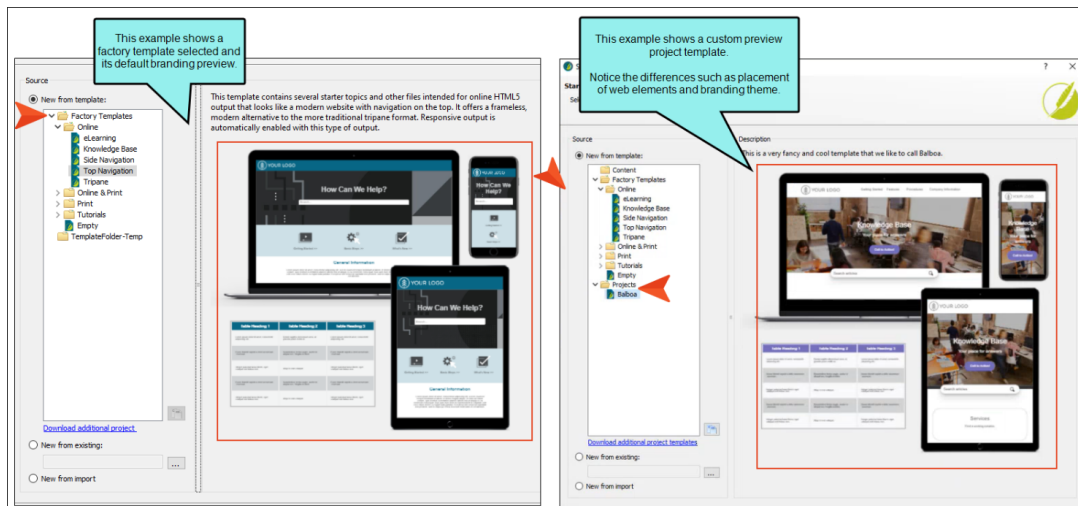
I Sharing Branding Files With Others

Once you have a branding stylesheet that includes custom branding values, you can share it with others. There are different ways to do this depending on your scope of work and what you want to do. Let's explore a few options.

- **No need to share a branding file** With this option, you have no further action than to designate branding for your project and be done with it. This might be the case if you are the only writer in your company, or if you are working with a team but you work in the same project. If the branding is defined in the one project, there is no reason to share it beyond that project.
- **Save as a template and share across multiple projects** You might work on a documentation team where each member maintains a project, but your company wants the team to use the same branding for every project. In this case, a `Branding.css` is created with specific colors, font, and images by one member of the team. It is saved as a template, and shared like any other file either locally or on a network.
- **Use Global Project Linking (GPL) to share the branding file** Your project consists of a parent project, and multiple child projects that link to the main one via GPL. As the lead writer for a large team, you create a branding stylesheet that you control at the parent level. You instruct each member of your team to import the CSS file from the parent project to the child project they are responsible for. In the end, every project is using the same branding file for a consistent look and feel in output.

 **NOTE** It is possible to create custom branding project templates. This means that you can adjust the interactive preview to display as a custom project template. The preview shown in the Start New Project Wizard and the standalone Branding Editor exists as a small HTML web page. You can create all the necessary CSS variable connections so it dynamically updates the look and feel with a custom theme.

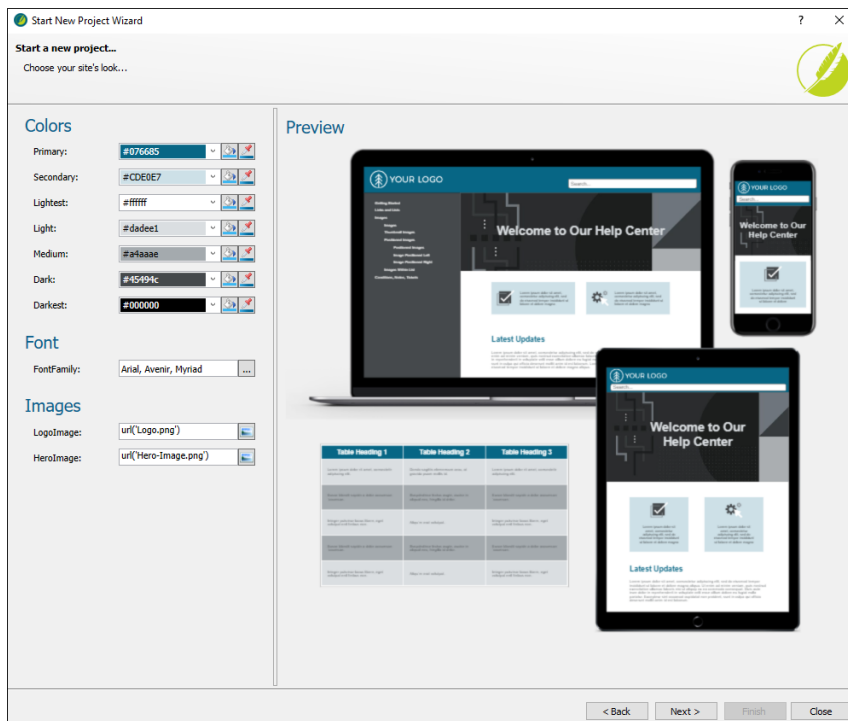
The main question to ask is, do you want the preview showing the templates to do anything? If you want to customize the templates in the preview to your own, then you might want to step through the advanced process of creating a custom preview project template.



CHAPTER 5

Custom Branding Project Templates

When you create a new project, there is a Preview area on the branding page of the Start a New Project wizard (and in the subsequent Branding Editor once the project is loaded). You can quickly change the preview (and therefore also the look of the project you're creating) by adjusting the color, font, and image fields on the page.



The interactive preview that you see is built in to Flare via a small web page associated with the selected template. But if you have created and saved your own custom project template, or if you add a new branding stylesheet to your established project, this preview won't really work unless you do some customization. The following steps explain how to accomplish such advanced customization. These steps should be done by someone who is experienced at both HTML and CSS, especially working in a text editor.

This chapter discusses the following:

Step 1: Create or Open a Project Template	33
Step 2: Identify the Main Branding Elements and Values of the Project	34
Step 3: Rename and Move Image Files	36
Step 4: Copy Branding CSS Variables From a Factory Template	38
Step 5: Replace Values in Stylesheets	44
Step 6: Replace Values in Skins	46
Step 7: Replace Images in Template Pages	48
Step 8: Edit Page Layouts	50
Step 9: Copy and Paste Preview Folder	54
Step 10: Edit the Branding.css Preview File	57
Step 11: Edit the Branding-Template.xml Preview File	59
Step 12: Replace the Image Files in the Preview Folder	61
Step 13: Edit the Preview HTML and CSS Files	63
Step 14: Edit the FLPRJ File	73
Step 15: Remove Unnecessary Folders	74
Step 16: Test It	75

I Step 1: Create or Open a Project Template

If you have already saved a project as a template using the Template Manager, simply open that project template. Otherwise, complete the following steps to create a project template.

1. Open a project that you want to save as a template.
2. If you haven't already done so, create a template folder (using **Tools > Manage Templates**) to hold your project template(s).
3. Select **Project > Save Project As Template**. Then complete the options in the wizard.

I Step 2: Identify the Main Branding Elements and Values of the Project

The following are branding elements that you might use in your custom project template. This list and the names are based on the built-in project templates in Flare. It's possible that you do not need to use all these elements in your custom project template. Also, some of the built-in templates might not have all these elements.

Primary The main brand color. For example, the primary color might impact H1 headings for some templates.

Secondary Another brand color.

Lightest The lightest color (e.g., white, off-white) for ancillary purposes.


Light A light color for ancillary purposes.


Medium A medium color for ancillary purposes.

Dark A dark color for ancillary purposes.


Darkest The darkest color (e.g., black, dark gray) for ancillary purposes.


FontFamily Your main font (or font set).

LogImage The main logo of your company or product. This lets you select an image for the background. Click  and use the dialog to select an image file. The default logo is replaced with the selected logo image file. A raster image (e.g., PNG) is suitable for online output.


LogImagePrint The main logo of your company or product, intended for print-based output. This lets you select an image for the background. Click  and use the dialog to select an image file. Templates that include a PDF allow you to select a vector image (e.g., EPS, SVG), which is made up of geometric elements and provides more clarity for printed output. If you use a raster image it might appear pixelated in the output.

BigLogo The main logo of your company or product. This is used only in the Brochure template to display a large version of the logo in one of panels. By default it is set to use a PNG file, but you can select a different file format, including a vector image.

HeroImage The main hero image shown on the home or cover page of the output. This lets you select an image for the background. Click  and use the dialog to select an image file. The default banner image at the top of online output, or the large image on a PDF title page, is replaced with the selected hero image file.

 **NOTE** The branding colors are designed to progress from lighter to darker shades. The colors are used for the preview and throughout the rest of the project, so use caution when changing a color from one extreme to another. For example, if you change a lighter element to a darker color, this could display dark content on top of other dark content. For the best results, adhere to the gradient progression.

This step is all about identifying the corresponding elements and values in your own project template.

 **EXAMPLE** We are setting up a project template named Balboa so that it can use the branding features in Flare. The most common colors in the stylesheets and skins in this project suggest that the following should be our default branding values:

Primary #7469B9

Secondary We don't have a secondary brand color, so we don't need to use it in our template.

Lightest #FFFFFF

Light #D3D3D3

Medium #7E8890

Dark #404040

Darkest #000000

Looking through our project, we found some colors that are close to one another. For example, #5A5A5A is quite close to #404040. In that case we could decide to use the **Dark** variable (#404040) for those elements as well, or we could just keep #5A5A5A associated directly with some elements. It's a matter of preference.

The main font in the project is Arial, so we can use that for **FontFamily**.

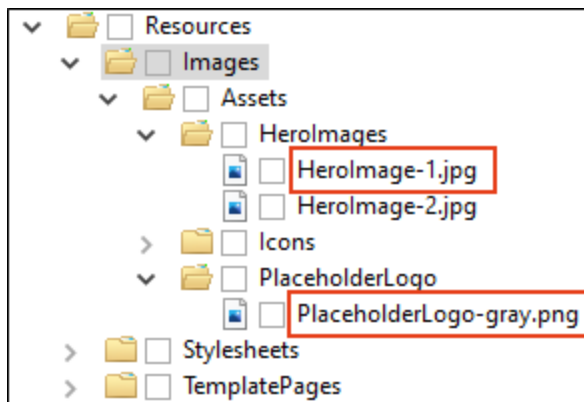
We have a logo image named PlaceholderLogo-gray.png, so we can use that for **LogoImage**. However, we will rename and move it in the next step so that it is mapped to the necessary preview files. Because this project is intended to produce only online output, we'll leave out **LogoImagePrint**.

We have two hero images for our home page in the output. One image simply appears higher than the other on the page. We'll use the first one, named HeroImage-1.jpg, for **HeroImage**. However, we will also rename and move this image file.

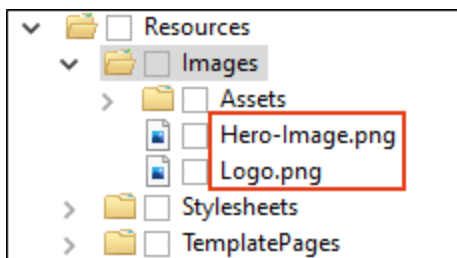
Step 3: Rename and Move Image Files


1. Open the Flare project template from the template folder where you saved it in Step 1: Create or Open a Project Template.
2. If you have a primary logo image, make sure it is named **Logo.png** and located in the **Resources > Images** folder (not in any other subfolder under Resources).
3. If you have a primary logo vector image that is meant for print-based output, make sure it is named **Logo.svg** and located in the **Resources > Images** folder (not in any other subfolder under Resources).
4. If you have a primary hero image, make sure it is named **Hero-Image.png** and located in the **Resources > Images** folder (not in any other subfolder under Resources).

☆ **EXAMPLE** In our Balboa project, this is how our image files originally looked in the Content Explorer:



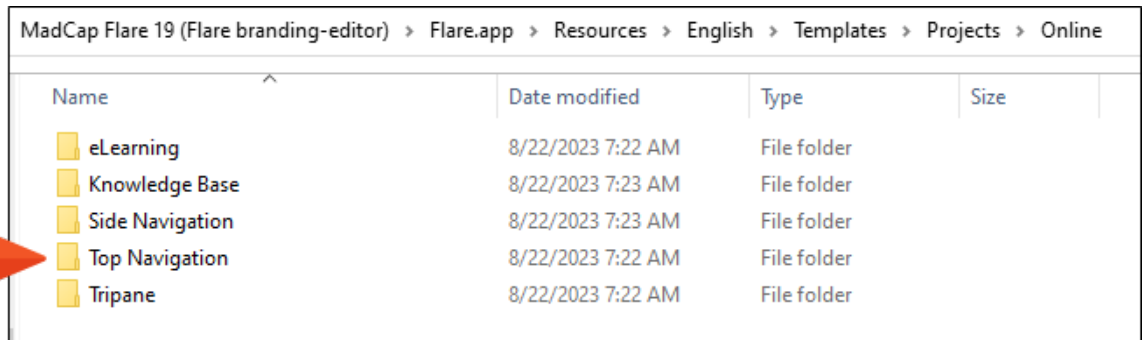
After renaming and moving them, the Content Explorer looked like this:



 **NOTE** If you choose not to rename or move your image files, you might then need to adjust information (e.g., the file path) in some of the other files that you copy from the built-in template later. The instructions in this procedure assume that you have the same names/locations as those in the built-in factory project template you will be copying files from.

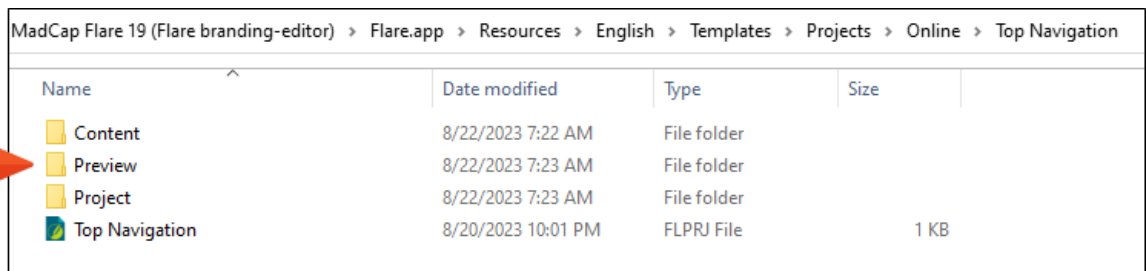
Step 4: Copy Branding CSS Variables From a Factory Template

1. In Windows, navigate to the location where you installed Flare.
2. Within that folder, navigate to **Flare.app > Resources > English > Templates > Projects**.
3. In that location in Windows, navigate to one of Flare's built-in factory template folders that most closely resembles your custom project template. For example, if your project uses the Top Navigation format, you might want to find the Top Navigation project template folder and files.



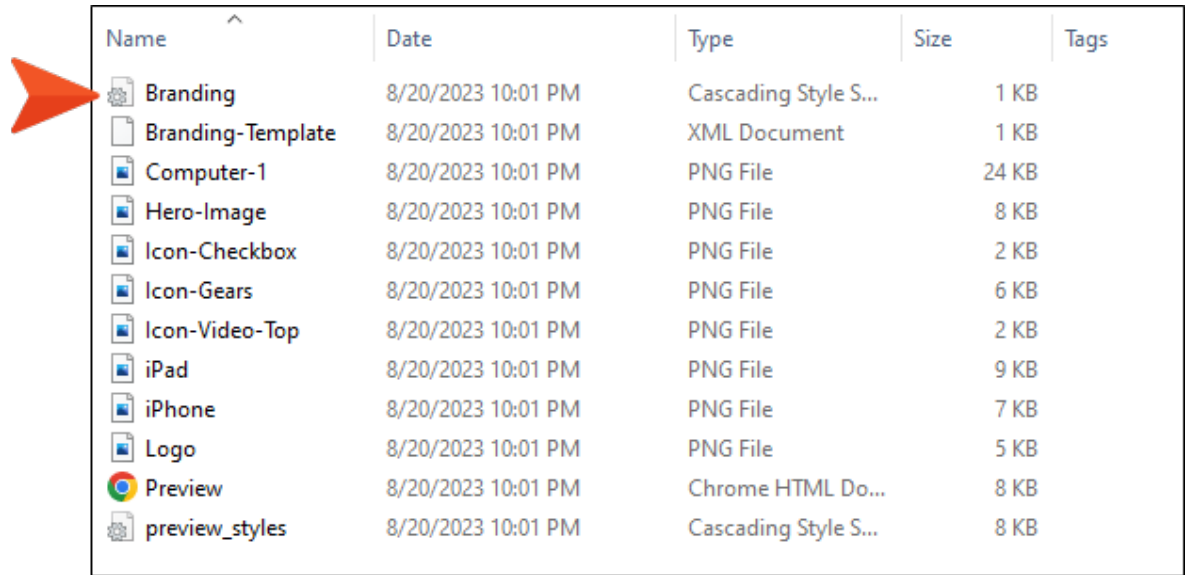
Name	Date modified	Type	Size
eLearning	8/22/2023 7:22 AM	File folder	
Knowledge Base	8/22/2023 7:23 AM	File folder	
Side Navigation	8/22/2023 7:23 AM	File folder	
Top Navigation	8/22/2023 7:22 AM	File folder	
Tripane	8/22/2023 7:22 AM	File folder	













4. Within that project template folder, open the **Preview** folder.



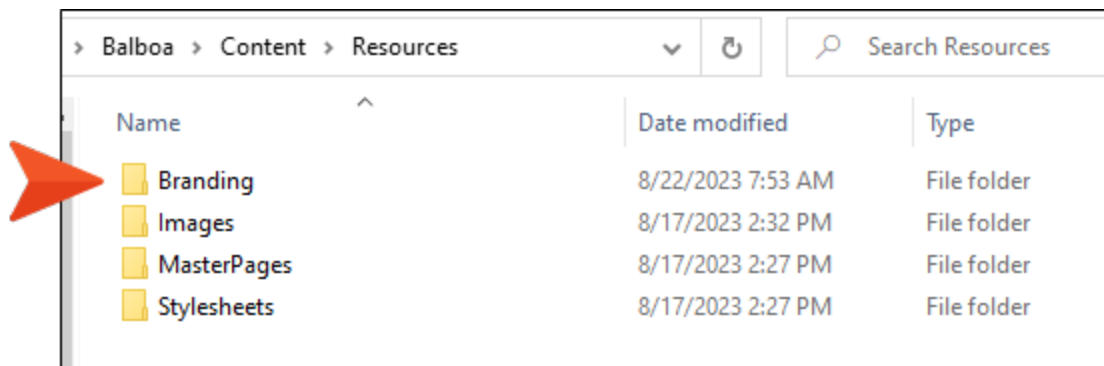
Name	Date modified	Type	Size
Content	8/22/2023 7:22 AM	File folder	
Preview	8/22/2023 7:23 AM	File folder	
Project	8/22/2023 7:23 AM	File folder	
Top Navigation	8/20/2023 10:01 PM	FLPRJ File	1 KB





5. Copy the **Branding.css** file.



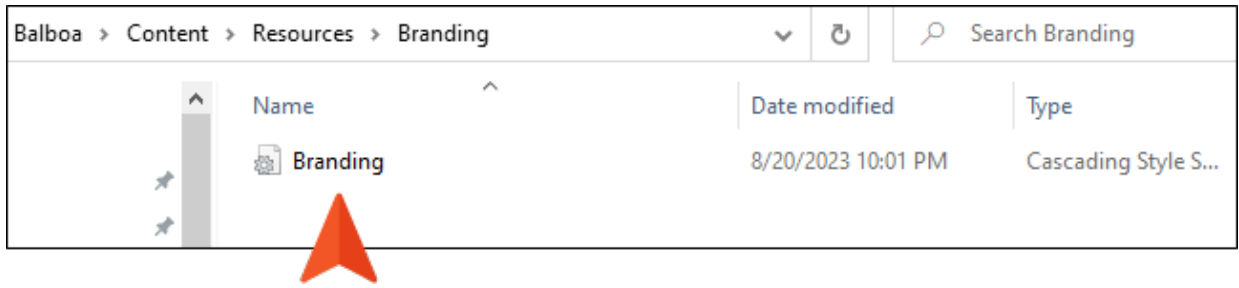
Name	Date	Type	Size	Tags
 Branding	8/20/2023 10:01 PM	Cascading Style S...	1 KB	
 Branding-Template	8/20/2023 10:01 PM	XML Document	1 KB	
 Computer-1	8/20/2023 10:01 PM	PNG File	24 KB	
 Hero-Image	8/20/2023 10:01 PM	PNG File	8 KB	
 Icon-Checkbox	8/20/2023 10:01 PM	PNG File	2 KB	
 Icon-Gears	8/20/2023 10:01 PM	PNG File	6 KB	
 Icon-Video-Top	8/20/2023 10:01 PM	PNG File	2 KB	
 iPad	8/20/2023 10:01 PM	PNG File	9 KB	
 iPhone	8/20/2023 10:01 PM	PNG File	7 KB	
 Logo	8/20/2023 10:01 PM	PNG File	5 KB	
 Preview	8/20/2023 10:01 PM	Chrome HTML Do...	8 KB	
 preview_styles	8/20/2023 10:01 PM	Cascading Style S...	8 KB	

6. In Windows, navigate to your Flare project template. In **Content > Resources**, create a **Branding** subfolder.

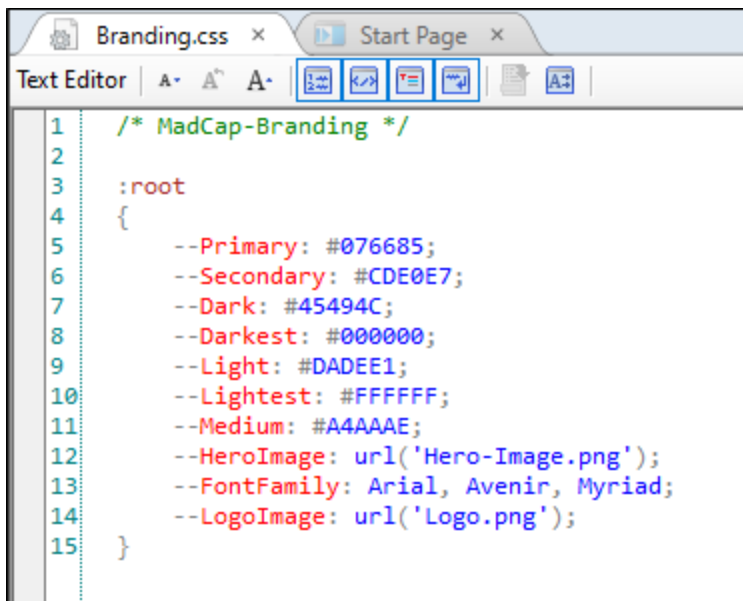


Name	Date modified	Type
 Branding	8/22/2023 7:53 AM	File folder
 Images	8/17/2023 2:32 PM	File folder
 MasterPages	8/17/2023 2:27 PM	File folder
 Stylesheets	8/17/2023 2:27 PM	File folder

7. Paste the **Branding.css** file within it. Make sure you do this in Windows and not in Flare. Otherwise, the files will not match exactly.

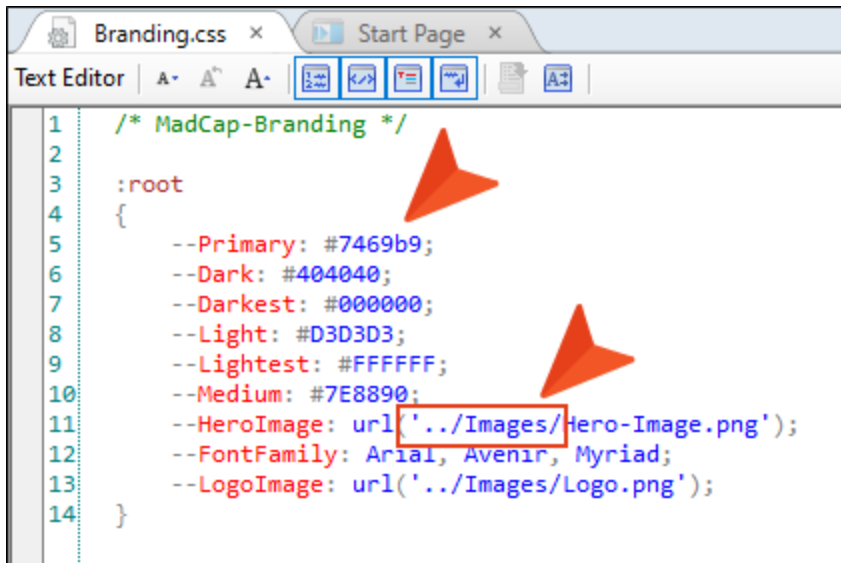


8. In your Flare project template, navigate to the **Branding.css** file in the Content Explorer and select **Open with > Internal Text Editor**. It should look something like this, depending on the factory template you're using.



9. Replace the values:

- Replace the color, image, and font values with those you identified for your project previously.
- Remove any CSS variables that you do not plan to use.
- Add relative paths for the images (i.e., add ../Images/ before the name of the image).

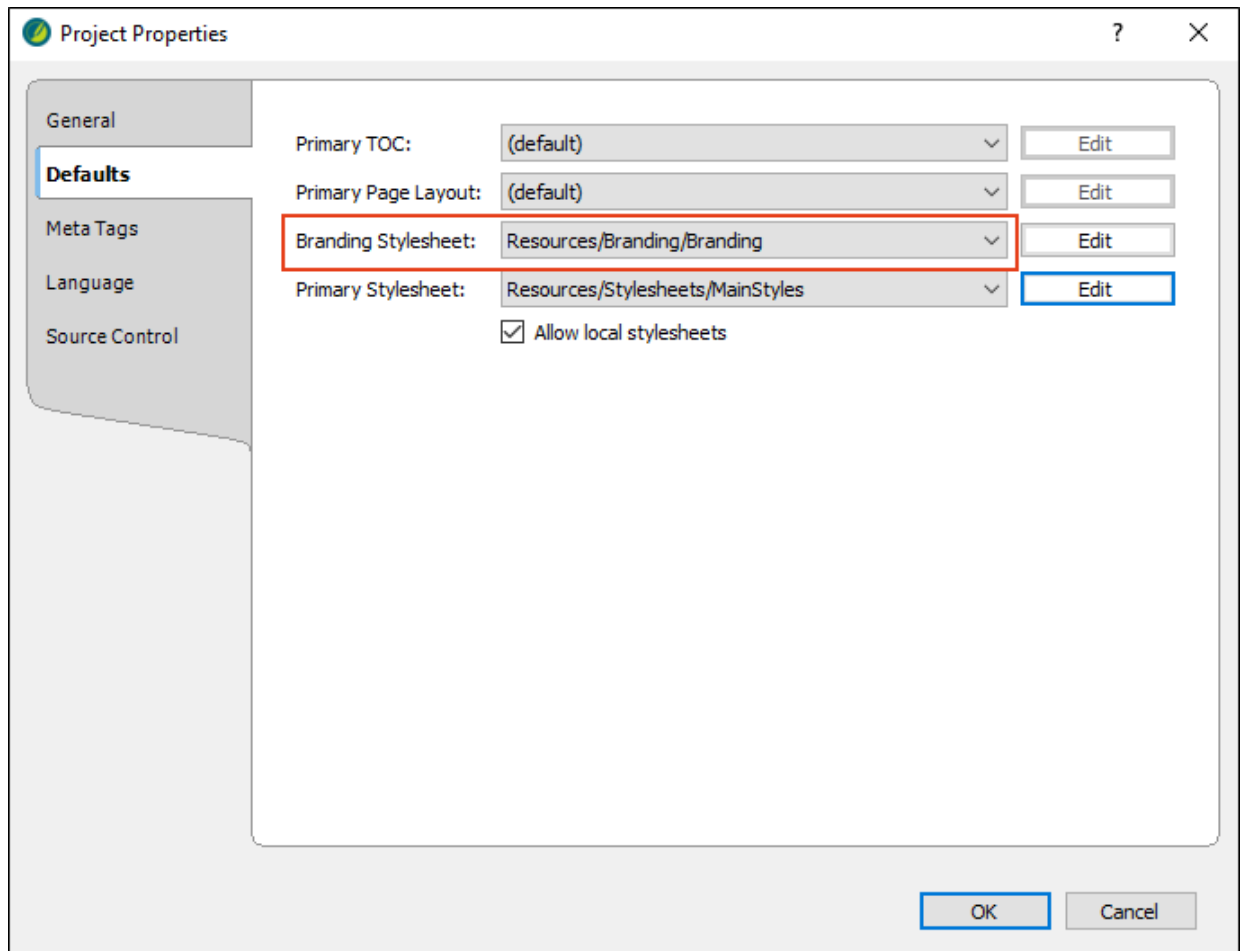


```
1  /* MadCap-Branding */
2
3  :root
4  {
5      --Primary: #7469b9;
6      --Dark: #404040;
7      --Darkest: #000000;
8      --Light: #D3D3D3;
9      --Lightest: #FFFFFF;
10     --Medium: #7E8890;
11     --HeroImage: url('../Images/Hero-Image.png');
12     --FontFamily: Arial, Avenir, Myriad;
13     --LogoImage: url('../Images/Logo.png');
14 }
```

10. Save your changes.

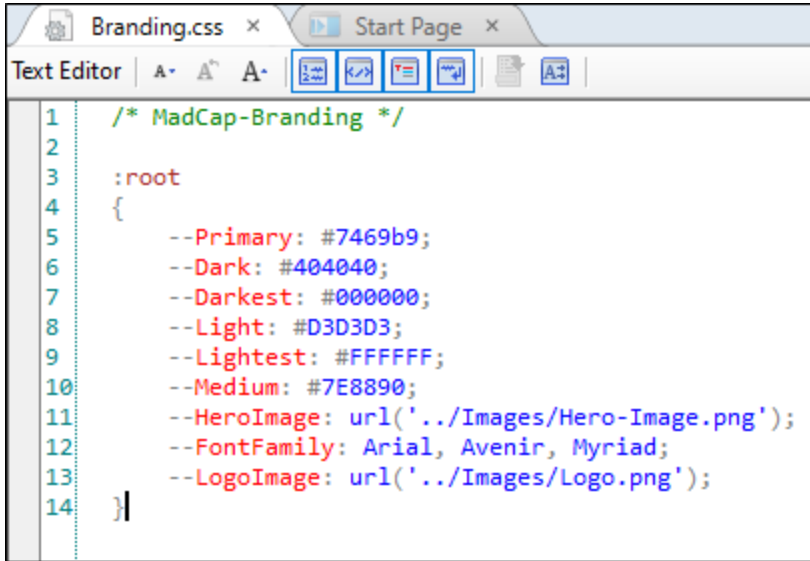
11. In the **Project** ribbon, select **Project Properties**.

- On the **Defaults** tab, make sure that the new Branding.css file is selected in the **Branding Stylesheet** field.



- Click **OK**.

☆ EXAMPLE In our Balboa project template, our edited set of CSS variables looks like this:

A screenshot of a text editor window titled 'Branding.css'. The editor shows CSS code for a project named 'MadCap-Branding'. The code is as follows:

```
1  /* MadCap-Branding */
2
3  :root
4  {
5      --Primary: #7469b9;
6      --Dark: #404040;
7      --Darkest: #000000;
8      --Light: #D3D3D3;
9      --Lightest: #FFFFFF;
10     --Medium: #7E8890;
11     --HeroImage: url('../Images/Hero-Image.png');
12     --FontFamily: Arial, Avenir, Myriad;
13     --LogoImage: url('../Images/Logo.png');
14 }
```

Notice the new color values, and the fact that we removed Secondary, because we don't plan to use it.

Our main font is Arial, but we just left the font set (Arial, Avenir, Myriad) in place because it might come in handy. If Arial is not available on the end user's machine, Avenir will be used. And if that's not available, Myriad will be used.

Also notice the paths of the image files.

Step 5: Replace Values in Stylesheets

1. Open the stylesheet(s) in your custom project template. You can do this in the Internal Text Editor (via find and replace) or in the Stylesheet Editor interface.
2. Look for any instances of color values (or previous CSS variables you might have been using). Replace them with the appropriate branding CSS variables (e.g., `var(--Primary1)`, `var(--Dark)`).

```
div.cta /*Call to action button/Contact Support button*/
{
  -moz-border-radius: 60px;
  -webkit-border-radius: 60px;
  border-radius: 60px;
  padding: 2px 0px;
  border: 1px solid var(--Primary);
  width: 162px;
  color: var(--Primary);
  transition: all 0.3s cubic-bezier(.25,.8,.25,1);
  margin-top: 20px;
}
```

3. Look for any instances of your main font (or font set). Replace them with the `var(--FontFamily)` CSS variable.

```
34 body
35 {
36     font-family: var(--FontFamily);
37     line-height: 1.5em;
38     background-color: var(--White);
39 }
40
```

4. Look for any instances of your main logo and/or hero image. Replace the logo reference with `var(--LogoImage)`, and replace the hero image with `var(--HeroImage)`.

```
120:
121:   div.topic-hero /*This is the first Hero/Banner on the
122:   {
123:     height: 500px;
124:     padding-bottom: 150px;
125:     padding-top: 150px;
126:     background-color: var(--Light);
127:     background-image: var(--HeroImage);
128:     background-size: cover;
129:     background-repeat: no-repeat;
130:     /* Create the parallax scrolling effect */
131:     background-attachment: fixed;
132:     background-position: center 0;
133:   }
134:
```

5. Save your changes.

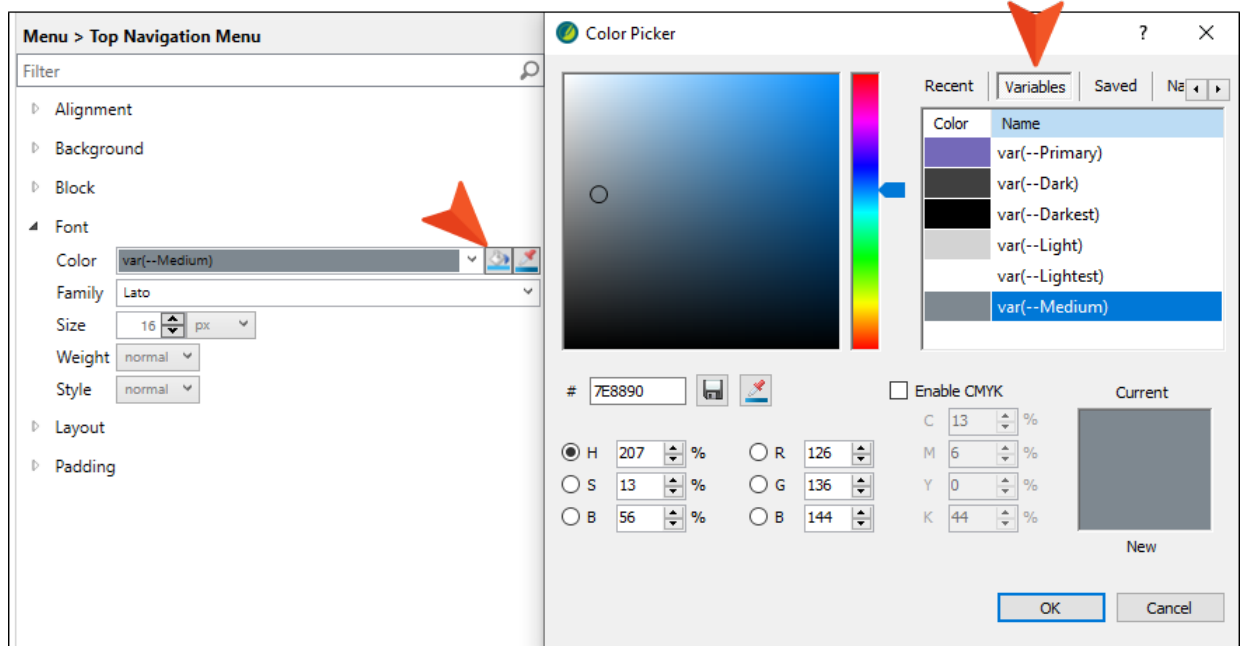
Step 6: Replace Values in Skins

In your skins, replace the color, font, and logo/hero image references with the appropriate CSS variables, just as you did in the previous step for the stylesheet(s).

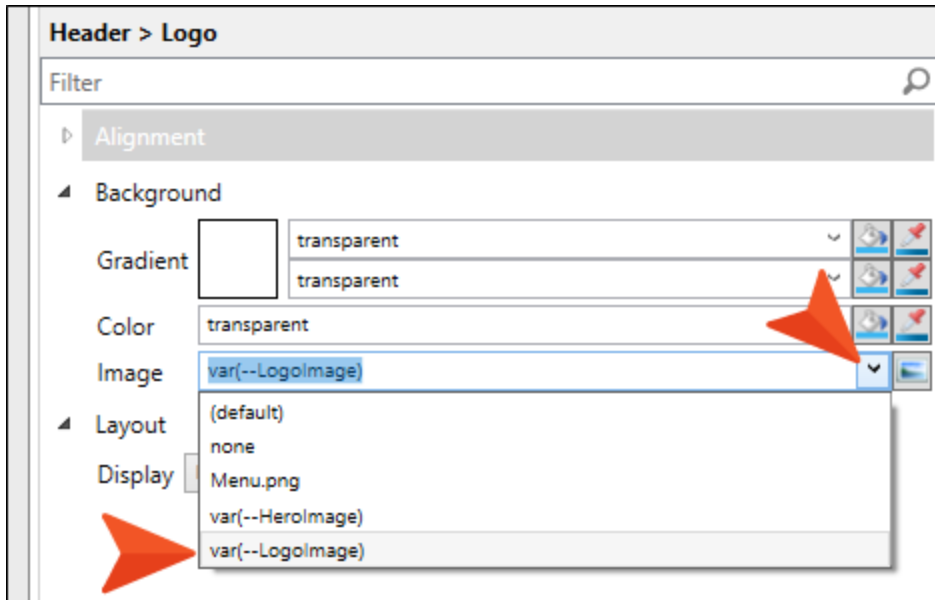
Make sure you do this for the Tablet and Mobile mediums in the skin as well, because there might be some fields that do not inherit values from the Web medium.


If you use the Skin Editor interface to change values (as opposed to the Internal Text Editor), you can use the following to select variables for the different elements:

- **Color** Click the **More colors** button.



- **Image** Use the drop-down field.

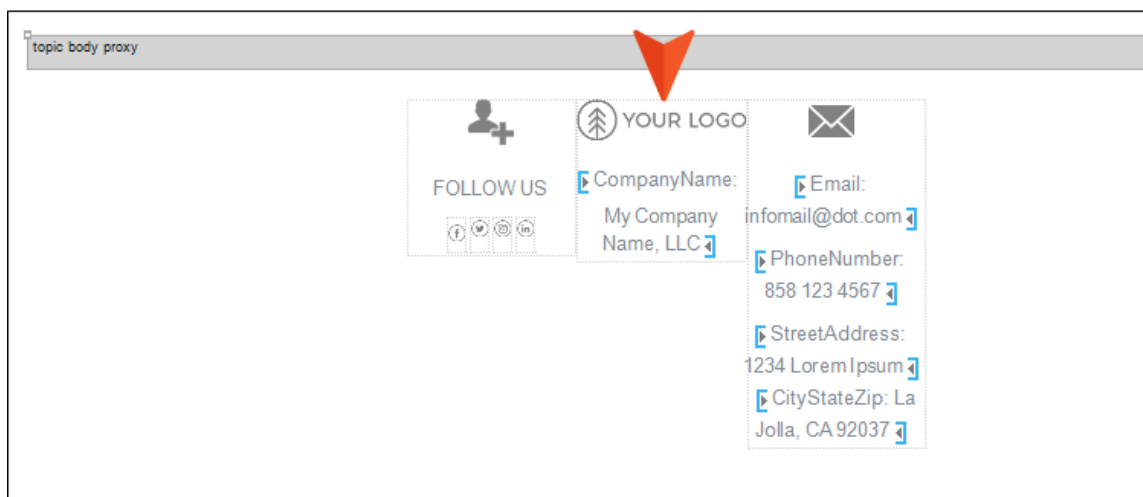


 **NOTE** There is currently no way to use the interface to select a FontFamily variable. Instead, you must enter it manually in the field or use the text editor.

Step 7: Replace Images in Template Pages

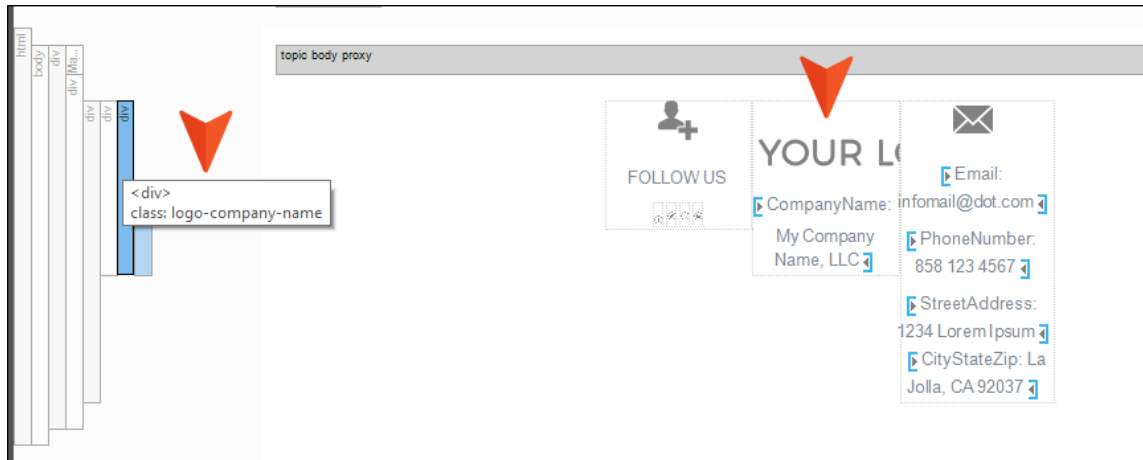
If you have inserted an image (most likely a logo) in any template pages, you might want to change this to use the appropriate CSS variable too. However, you are not able to select a CSS variable for an image inserted directly into a content file in Flare. Instead, you must use a background image. Therefore, you might need to adjust your content in the template page a bit to make use of the CSS variable for the background image.

☆ **EXAMPLE** In our Balboa project, the template page used for our Home page in the online output looks like this, with the logo inserted directly into it:



☆ To solve this, we created a special div in our stylesheet called `div.logo-company-name`, since that is the content that will be shown within it.

We removed the inserted image and wrapped the content in that new div.



Here are the properties and values for that div, which sets the `LogoImage` variable on the background image:

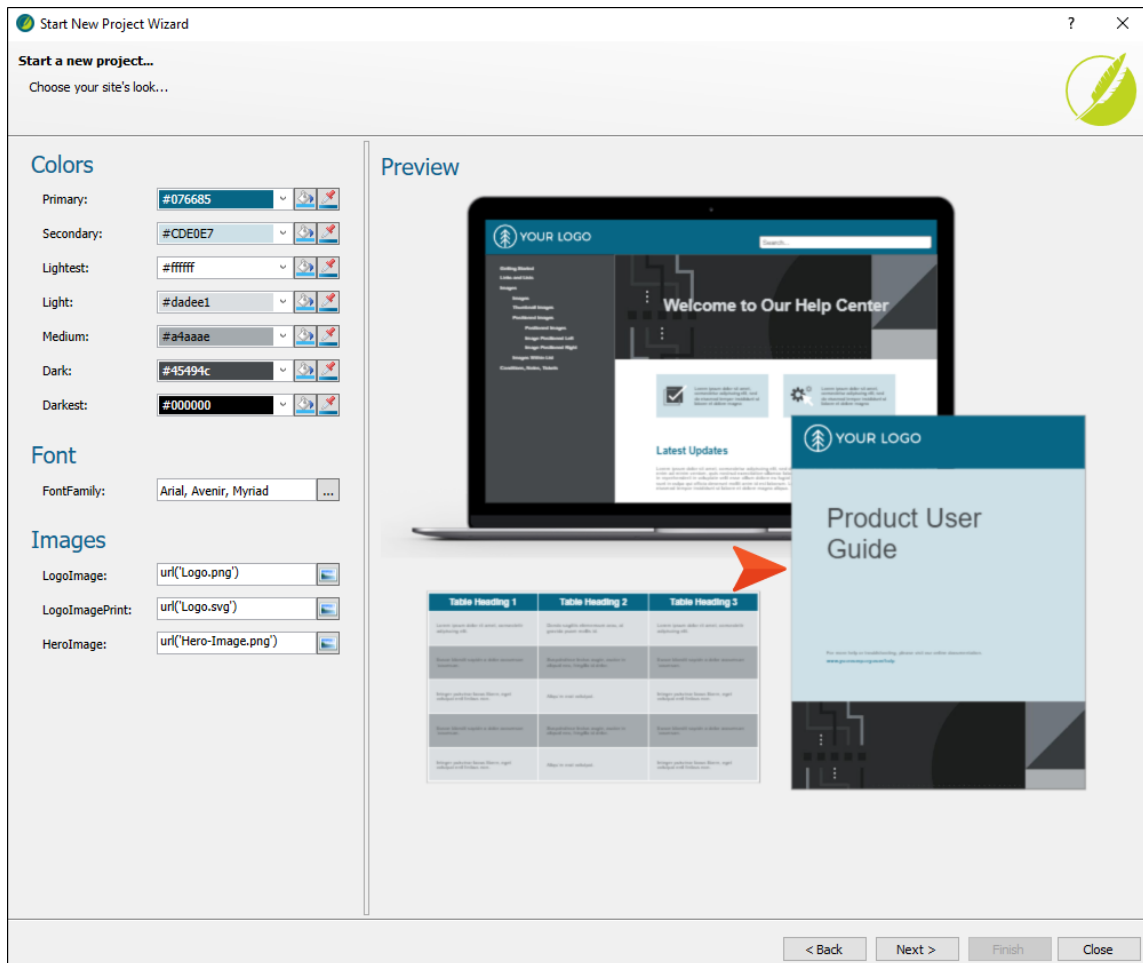
```
div.logo-company-name
{
  background-repeat: no-repeat;
  background-image: var(--LogoImage);
  padding-top: 70px;
  background-position: 50% 15%;
}
```

Step 8: Edit Page Layouts

If you have a project template that includes PDF output, the steps for creating a preview are a little more involved. The reason for this is that the Page Layout Editor does not allow you to select CSS variables in the fields.

If you have a template with PDF output, you'll need to make some changes to your page layout(s) using the Internal Text Editor. These changes will include references to styles that you create in your regular stylesheet. By observing the page layouts and stylesheets in Flare's factory project templates that include PDF output (e.g., Side Navigation and PDF, Top Navigation and PDF), you can leverage the information for your own custom template.

☆ **EXAMPLE** If you create a test project based on Flare's Side Navigation and PDF template, you will notice that the preview for the PDF title page looks like this:



- ★ Various sections of the related page layout (in this case, Frontmatter.flgpl) can be customized by selecting color, font, and image CSS variables.

When you open the Frontmatter page layout file in the Internal Text Editor, you will see the following markup. To integrate CSS variables with the page layout, the sections with the <Content> tags (highlighted in yellow) were added inside the frames, with references to style classes. A normal page layout would not have these sections.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <PageStyles>
3 <Page Width="7in" Height="9in" Padding="1in" PageSize="Letter" PaddingLeft="1in" PaddingRight="1in" PaddingTop="1in" PaddingBot
4 <Frame Flow="header" X="-99px" Y="-101px" Width="825px" Height="155px" Right="-102px" Bottom="810px" Anchors="">
5 <Content>
6 <xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
7 <xhtml:body>
8 <xhtml:div xhtml:class="page-layout-front-header">
9 <xhtml:table />
10 </xhtml:div>
11 </xhtml:body>
12 </xhtml:html>
13 </Content>
14 </Frame>
15 <Frame Flow="decoration0" X="-49px" Y="-70px" Width="724px" Height="90px" Right="-102px" Bottom="810px" Anchors="">
16 <Content>
17 <xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
18 <xhtml:body>
19 <xhtml:div xhtml:class="page-layout-front-logo">
20 <xhtml:table />
21 </xhtml:div>
22 </xhtml:body>
23 </xhtml:html>
24 </Content>
25 </Frame>
26 <Frame Flow="decoration1" X="-99px" Y="54px" Width="825px" Height="635px" Right="-102px" Bottom="810px" Anchors="">
27 <Content>
28 <xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
29 <xhtml:body>
30 <xhtml:div xhtml:class="page-layout-front-main">
31 <xhtml:table />
32 </xhtml:div>
33 </xhtml:body>
34 </xhtml:html>
35 </Content>
36 </Frame>
37 <Frame Flow="body" FlowIndex="0" ColumnCount="1" ColumnGap="0.25in" ColumnDivider="Solid 1pt Black" Width="615px" Height="3
Bottom="398px">
38 </Frame>
39 <Frame Flow="body" FlowIndex="1" ColumnCount="1" ColumnGap="0.25in" ColumnDivider="Solid 1pt Black" Width="615px" Height="1
Bottom="207px" BorderTopStyle="solid" BorderTopWidth="1px" BorderTopColor="#ffffff">
40 </Frame>
41 <Frame Flow="footer" X="-99px" Y="686px" Width="825px" Height="275px" Right="-102px" Bottom="-97px" Anchors="">
42 <Content>
43 <xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
44 <xhtml:body>
45 <xhtml:div xhtml:class="page-layout-front-footer">
```

- ☆ If you then open the Styles.css file in the Internal Text Editor, you will notice these generic style classes, which contain references to different branding CSS variables.


```
.page-layout-front-header
{
  background-color: var(--Primary);
  height: 155px;
  width: 825px;
}


.page-layout-front-logo
{
  height: 90px;
  width: 724px;
  background-color: transparent;
  background-image: var(--LogoImagePrint);
  background-repeat: no-repeat;
  background-size: auto 100%;
}


.page-layout-front-main
{
  background-color: var(--Secondary);
  height: 650px;
  margin: 0px;
}

.page-layout-front-footer
{
  height: 275px;
  margin: 0px;
  background-color: var(--Dark);
  background-image: var(--HeroImage);
  background-repeat: no-repeat;
  background-size: cover;
}
```

Using this method, you can ensure that your page layout elements can be updated easily via CSS variables.

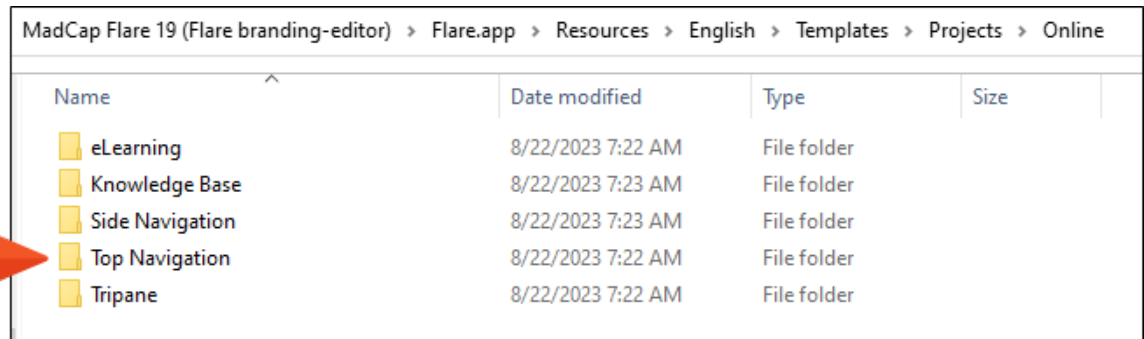
 **NOTE** If you want to use different style classes those provided by the factory templates and mentioned in the example above, you would add them to the markup of the page layout file. Also, make sure those classes are added to your stylesheet, and provide properties and values to specify how you want that part of the page layout to look.

 **NOTE** When customizing portions of a page layout in this way, make sure you are working within a section where the "Flow" is a header, footer, or decoration. Do not make changes in a section where the Flow is body.

 **NOTE** If you want to set a background color or image on a div, you should set the height for the frame in both the Page Layout Editor, as well as the height for the div in your stylesheet. Otherwise, the div will "collapse" and you won't get the results you want.

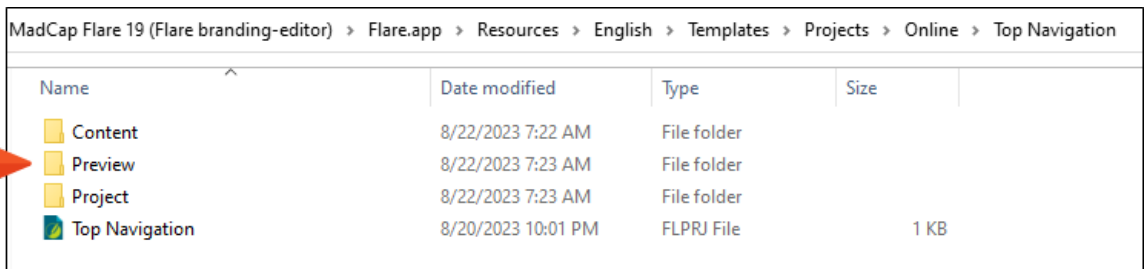
Step 9: Copy and Paste Preview Folder

1. In Windows, navigate to the location where you installed Flare.
2. Within that folder, navigate to **Flare.app > Resources > English > Templates > Projects**.
3. In that location in Windows, navigate to one of Flare's built-in factory template folders that most closely resembles your custom project template. For example, if your project uses the Top Navigation format, you might want to find the Top Navigation project template folder and files.



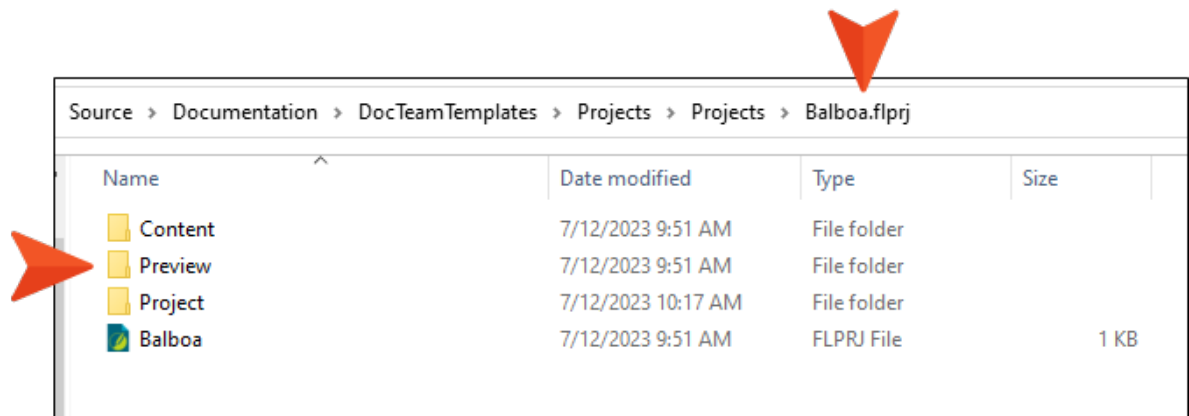
Name	Date modified	Type	Size
eLearning	8/22/2023 7:22 AM	File folder	
Knowledge Base	8/22/2023 7:23 AM	File folder	
Side Navigation	8/22/2023 7:23 AM	File folder	
Top Navigation	8/22/2023 7:22 AM	File folder	
Tripane	8/22/2023 7:22 AM	File folder	

4. Within that project template folder, copy the **Preview** folder.



Name	Date modified	Type	Size
Content	8/22/2023 7:22 AM	File folder	
Preview	8/22/2023 7:23 AM	File folder	
Project	8/22/2023 7:23 AM	File folder	
Top Navigation	8/20/2023 10:01 PM	FLPRJ File	1 KB













- Paste it into the Windows folder holding your custom project template files. In the following example, we pasted it into our Balboa project template, which is located on one of our company's network servers. (This is the same location where we used the Template Manager to save our project template in Step 1: Create or Open a Project Template. The location for your project template will of course be different.)



The Preview folder should contain these files:

- Branding.css
- Branding-Template.xml
- Preview.html
- Preview_styles.css (might not be included in some of Flare's built-in project templates)
- Any image files that are part of the template that support the branding preview (e.g., Logo.png, Hero-Image.png)

Here are the contents of the Preview folder that we copied from the Top Navigation project template for this example:

Name	Date modified	Type	Size
 Branding	7/10/2023 11:10 AM	Cascading Style S...	1 KB
 Branding-Template	7/5/2023 7:08 AM	XML Document	1 KB
 Computer-1	7/5/2023 7:08 AM	PNG File	24 KB
 Hero-Image	7/5/2023 7:08 AM	PNG File	8 KB
 Icon-Checkbox	7/5/2023 7:08 AM	PNG File	2 KB
 Icon-Gears	7/5/2023 7:08 AM	PNG File	6 KB
 Icon-Video-Top	7/5/2023 7:08 AM	PNG File	2 KB
 iPad	7/5/2023 7:08 AM	PNG File	9 KB
 iPhone	7/5/2023 7:08 AM	PNG File	7 KB
 Logo	7/5/2023 7:08 AM	PNG File	5 KB
 Preview	7/10/2023 11:10 AM	Chrome HTML Do...	7 KB
 preview_styles	7/5/2023 7:08 AM	Cascading Style S...	7 KB

As you enhance your own project template, you'll edit some of these, and perhaps replace or remove others.

Step 10: Edit the Branding.css Preview File

The Branding.css file holds the main CSS variables that will be used on the branding page of the Start a New Project Wizard or the Branding Editor in Flare.


1. From the Preview folder you copied, open the **Branding.css** file in a text editor, such as Notepad++.
2. Replace the values with those you will be using in your project template.


☆ **EXAMPLE** In our Balboa project template, our edited Branding.css file looks like this:

```
1  /* MadCap-Branding */
2
3  :root
4  {
5      --Primary: #7469b9;
6      --Dark: #404040;
7      --Darkest: #000000;
8      --Light: #D3D3D3;
9      --Lightest: #FFFFFF;
10     --Medium: #7E8890;
11     --HeroImage: url('Hero-Image.png');
12     --FontFamily: Arial, Avenir, Myriad;
13     --LogoImage: url('Logo.png');
14 }
```

As previously done, we removed Secondary, because we do not intend to use it.

3. Save your changes.

 **NOTE** Just make sure that whatever variables you have for your preview files matches whatever names you have in the rest of the project (e.g., Branding.css file under Resources > Branding, other stylesheets, skins). Otherwise, you might find that the preview feature in Flare seems to work fine, but the correct values aren't applied to files in the rest of the project, and therefore also your output.

 **NOTE** You can add other CSS variables in this file if you like. For example, you might want a couple of CSS variables for link colors. If you decide to do this, make sure you also add those custom variables to the other Resources > Branding > Branding.css file you edited before, as well as to the Branding-Template.xml file that you will be editing in the next step.

Step 11: Edit the Branding-Template.xml Preview File

The Branding-Template.xml file contains the branding schema for the project template.

1. From the Preview folder you copied, open the **Branding-Template.xml** file in a text editor, such as Notepad++.
2. Replace the values with those you will be using in your project template. In other words, they should be the same as those in the Branding.css file you just edited.

☆ **EXAMPLE** In our Balboa project template, our edited Branding-Template.xml file looks like this:

```
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <BrandingTemplate Name="Top Navigation" >
4   <Property Name="Primary" Type="Color" Group="Colors" DefaultValue="#7469B9" />
5   <Property Name="Lightest" Type="Color" Group="Colors" DefaultValue="#FFFFFF" />
6   <Property Name="Light" Type="Color" Group="Colors" DefaultValue="#D3D3D3" />
7   <Property Name="Medium" Type="Color" Group="Colors" DefaultValue="#7E8890" />
8   <Property Name="Dark" Type="Color" Group="Colors" DefaultValue="#404040" />
9   <Property Name="Darkest" Type="Color" Group="Colors" DefaultValue="#000000" />
10  <Property Name="FontFamily" Type="Font" Group="Font" DefaultValue="Arial, Avenir, Myriad" />
11  <Property Name="LogoImage" Type="Image" Group="Images" DefaultValue="url('Logo.png')"/>
12  <Property Name="HeroImage" Type="Image" Group="Images" DefaultValue="url('Hero-Image.png')"/>
13 </BrandingTemplate>
```

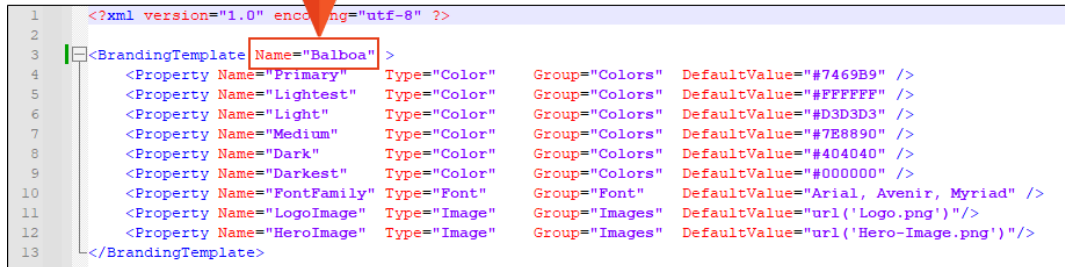
Just as we did with the variables in our stylesheet, we removed Secondary, because we do not intend to use it.

3. (Optional) The order in which the template properties appear in this file determines the order they will appear in the branding page of the Start a New Project Wizard or the Branding Editor.

If you want to change the order of the fields, you can move the properties around in this file. However, this only affects the order of the fields in the same group (i.e., Colors, Font, Images). For example, you can move the color properties around so that they display in alphabetical order, but you cannot move the image fields above the color fields.

4. Edit the name of the template to reflect your project name.

☆ **EXAMPLE** Our project template is name Balboa, so that's the name we gave it in this file:

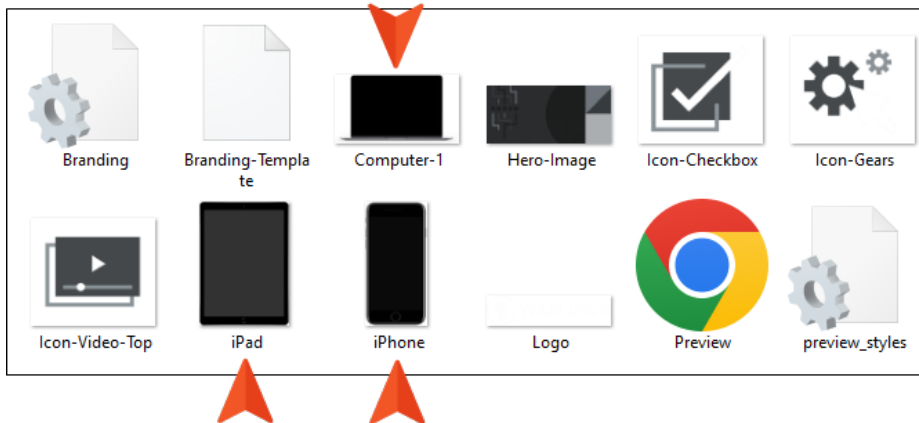


```
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <BrandingTemplate Name="Balboa" >
4   <Property Name="Primary" Type="Color" Group="Colors" DefaultValue="#7469B9" />
5   <Property Name="Lightest" Type="Color" Group="Colors" DefaultValue="FFFFFF" />
6   <Property Name="Light" Type="Color" Group="Colors" DefaultValue="#D3D3D3" />
7   <Property Name="Medium" Type="Color" Group="Colors" DefaultValue="#7B8890" />
8   <Property Name="Dark" Type="Color" Group="Colors" DefaultValue="#404040" />
9   <Property Name="Darkest" Type="Color" Group="Colors" DefaultValue="000000" />
10  <Property Name="FontFamily" Type="Font" Group="Font" DefaultValue="Arial, Avenir, Myriad" />
11  <Property Name="LogoImage" Type="Image" Group="Images" DefaultValue="url('Logo.png')"/>
12  <Property Name="HeroImage" Type="Image" Group="Images" DefaultValue="url('Hero-Image.png')"/>
13 </BrandingTemplate>
```

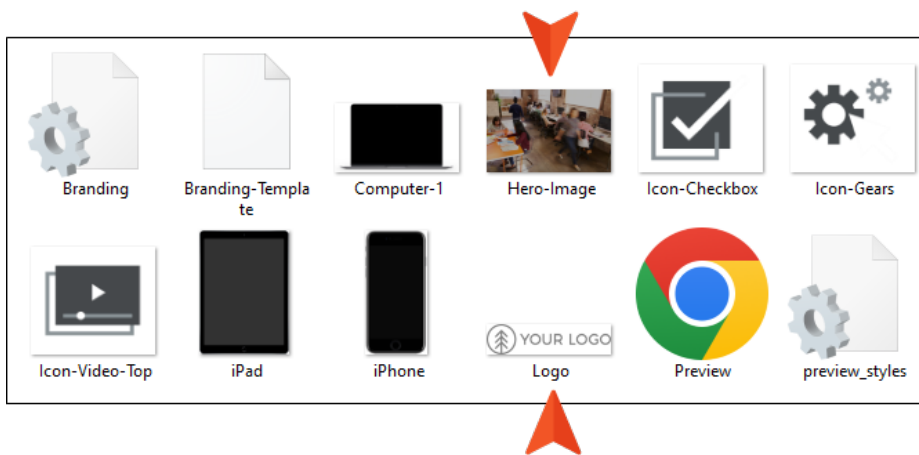
5. Save your changes.

Step 12: Replace the Image Files in the Preview Folder

1. From the Preview folder you copied, notice the image files. Some of these (Computer-1, iPad, and iPhone) will be used to display your output pages in the branding preview in Flare. You can just leave these files in the folder, unless you don't plan on using one, in which case you can remove it.

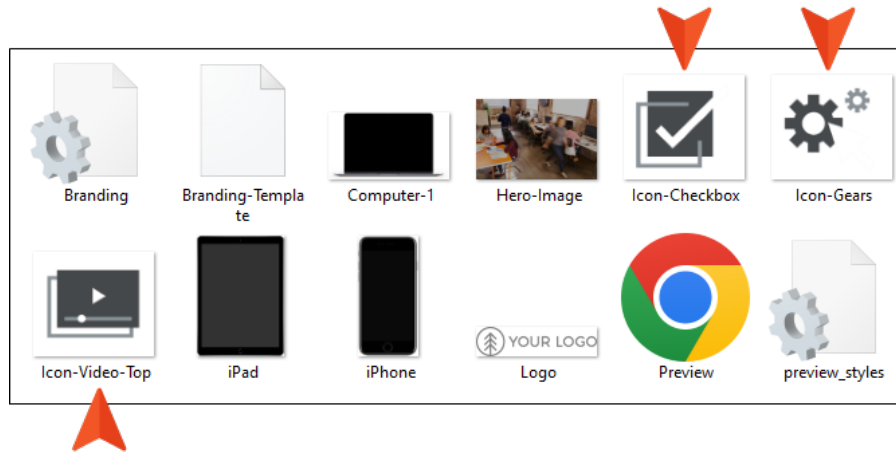


2. Replace the **Hero-Image** and **Logo** images with those from your project.



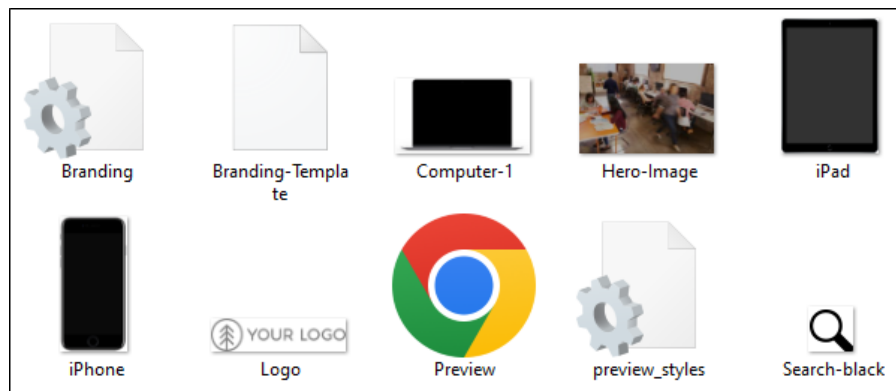
3. Add any other images you might need to use in your preview, and remove others that you don't need.

☆ **EXAMPLE** In our Balboa project template, the Preview folder has three images that were used in the Top Navigation template, which we used as the basis for our customization.



We don't need any of these images, because they're not used in our project, so we removed them.

We added one image to be used with the search field in our preview. So after all our changes, our Preview folder looks like this:



I Step 13: Edit the Preview HTML and CSS Files

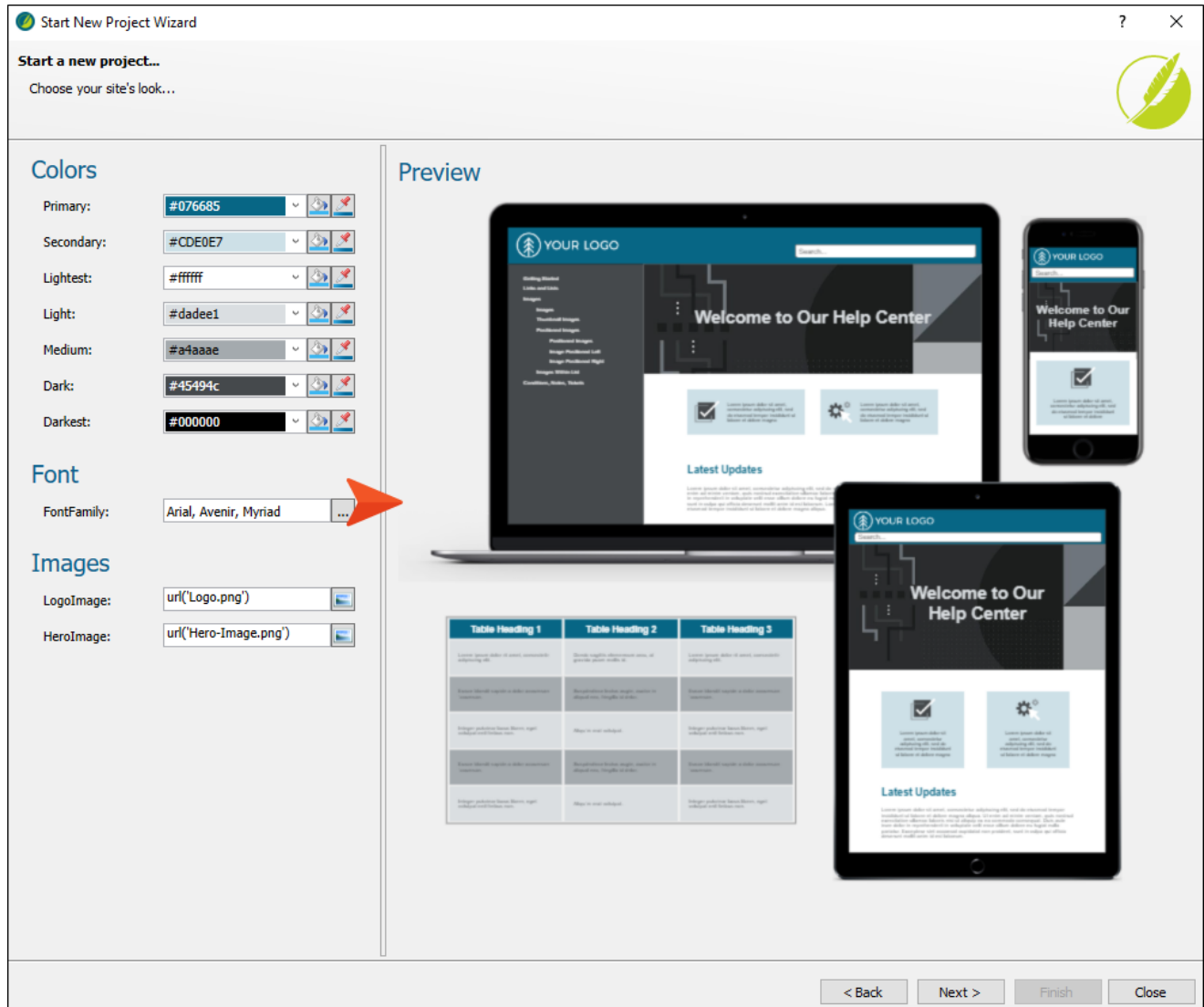
This is where much of the work occurs, and it requires someone who is knowledgeable at editing both HTML and CSS text files.

How to Edit the Preview HTML and CSS Files

1. Build output from your project. In this case, we are generating HTML5 output.
2. Open the output to see how it looks on a large screen, a tablet, and a smart phone.
3. In Windows, navigate to your Preview folder and open the **Preview.html** and **preview_styles.css** files in a text editor, such as Notepad++.
4. Based on how your output looks, and on the following information, edit both of those files until you match the desired look. It can also help to open the Preview.html file in a browser and refresh it as you make edits to the source files.
5. Save your changes.

Preview.html File

This is the small web page viewed on the branding page of the Start a New Project Wizard or the Branding Editor.



You will make changes to this file to re-create the look of your project template output for the branding preview.

In the <head>, make sure the following line is included:

```
<script language="javascript"></script>
```


This allows the resizing JavaScript code to run.

```
<!doctype html>
<html lang="en">

<head>
  <title>TopNav</title>
  <script language="javascript"></script>
  <link rel="stylesheet" href="preview_styles.css">
  <link rel="stylesheet" href="Branding.css">
</head>

<body>
  <div id="main_area" class="FontFamily_font">
    <div id="computer">
      <div id="inside_computer">
        <div class="Lightest_bg" id="computer_top_banner">
          <div class="logo LogoImage_img">&nbsp;</div>
          <div class="nav_wrapper Medium_col">
            <div class="nav1">
              <p class="nav">Getting Started</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

The bounding box of this web page should be an element ID `main_area`. This element is referenced by name in the resizer script, so it needs to be there, and it should have a size of 880px wide by 800px tall.

```
<!doctype html>
<html lang="en">

<head>
  <title>TopNav</title>
  <script language="javascript"></script>
  <link rel="stylesheet" href="preview_styles.css">
  <link rel="stylesheet" href="Branding.css">
</head>

<body>
  <div id="main_area" class="FontFamily_font">
    <div id="computer">
      <div id="inside_computer">
        <div class="Lightest_bg" id="computer_top_banner">
          <div class="logo LogoImage_img">&nbsp;</div>
          <div class="nav_wrapper Medium_col">
            <div class="nav1">
              <p class="nav">Getting Started</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Here are the naming conventions for areas of elements that can be changed with branding properties:

Name_col Font color property of element (e.g., Primary_col).

Name_bord_col Border color property of element (e.g., Primary_bord_col).

Name_bg Background color property of element (e.g., Primary_bg).

Name_img Background image property of element (e.g., LogoImage_img).

Name_font Font family property of element (e.g., FontFamily_font).

```

<!doctype html>
<html lang="en">

<head>
  <title>TopNav</title>
  <script language="javascript"></script>
  <link rel="stylesheet" href="preview_styles.css">
  <link rel="stylesheet" href="Branding.css">
</head>

<body>
  <div id="main_area" class="FontFamily_font">
    <div id="computer">
      <div id="inside_computer">
        <div class="Lightest_bg" id="computer_top_banner">
          <div class="logo LogoImage_img">&nbsp;</div>
          <div class="nav_wrapper Medium_col">
            <div class="nav1">
              <p class="nav">Getting Started</p>
            </div>
            <div class="nav2">
              <p class="nav">Features</p>
            </div>
            <div class="nav3">
              <p class="nav">Procedures</p>
            </div>
            <div class="nav4">
              <p class="nav">Company Information</p>
            </div>
          </div>
        </div>
        <div class="HeroImage_img" id="computer_hero_area">
          <h1 class="Lightest_col">Knowledge Base</h1>
          <h2 class="Lightest_col">Your place for answers</h2>
          <div class="centered call to action Primary_bg">
            <p class="cta Lightest_col">Call to Action!</p>
          </div>
        </div>
        <div class="general_area Lightest_bg" id="computer_general_area">
          <div class="centered search_bar Dark_col Lightest_bg">
            <p class="search_text">Search articles</p>
            </img>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>

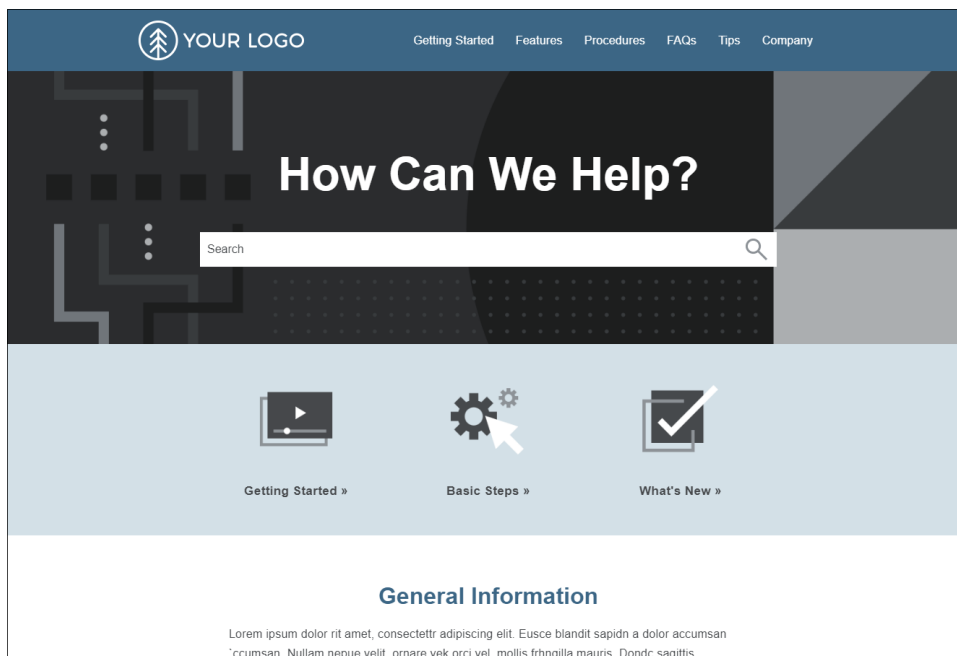
```

These relate to the properties in the Branding-Template.xml file, and to the branding CSS variables that will become part of a new Flare project created from the custom project template. For example, perhaps there is an area of your preview web page that is currently using Primary_bg. It will therefore display the color associated with the Primary CSS variable in the project. But you could edit this to something else, such as Secondary_bg. And in that case, this section of the preview web page will use the color from the Secondary CSS variable instead.

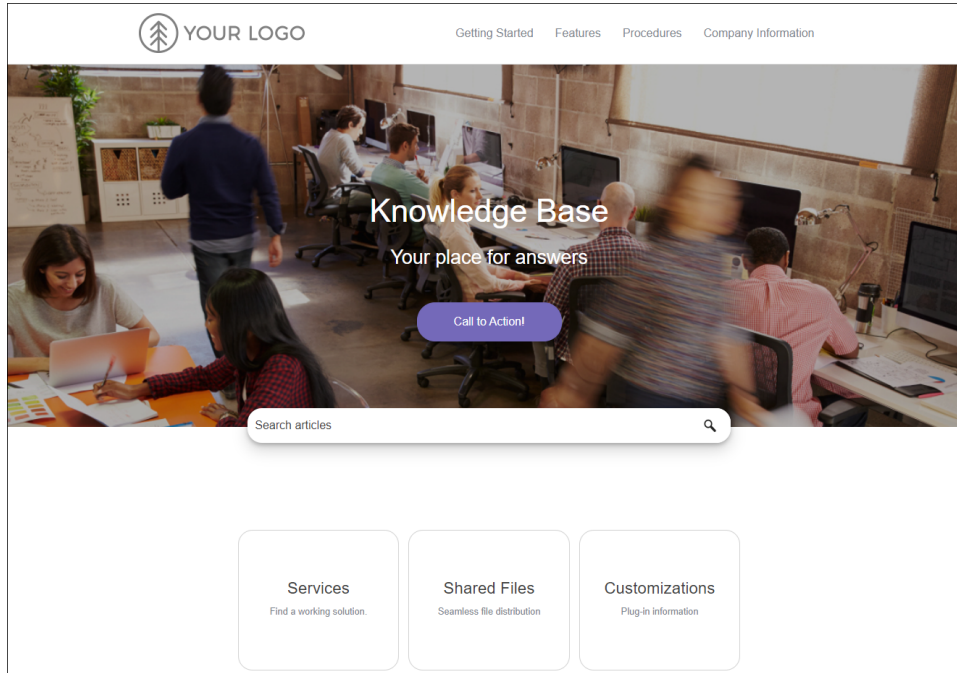
Preview_styles.css File

This file is used to control the look of elements in the small preview web page (Preview.html File). In addition to other styles, you will see selectors corresponding to the various CSS variables (e.g., .Primary_bg is the selector relating to the background color for the Primary CSS variable). In addition, you will see IDs for the different elements in the preview (#computer for the desktop, #iphone for a smart phone, #ipad for a tablet).

☆ **EXAMPLE** The HTML5 output for the Top Navigation project (the basis we selected for our new project template) looks like this on a big monitor:



☆ But our Balboa project looks like this on a big monitor:



So we needed to make many changes.

This is just the Home page, and it doesn't even show all of it. As you will see below, this is even more true with the tablet and smart phone views. Therefore, you might decide to display an interior topic for one or more of the other views so that users can see exactly what will change as they select colors, a font, and images from the branding fields. But in this example, we will just use the Home page for all of our views.

- ☆ For the large desktop view, the Preview.html file started out looking like this (since it was based on the Top Navigation template):

```
<!doctype html>
<html lang="en">

<head>
  <title>TopNav</title>
  <script language="javascript"></script>
  <link rel="stylesheet" href="preview_styles.css">
  <link rel="stylesheet" href="Branding.css">
</head>

<body>
  <div id="main_area" class="FontFamily_font">
    <div id="computer">
      <div id="inside_computer">
        <div class="Primary_bg" id="computer_top_banner">
          <div class="LogoImage_img">&nbsp;</div>
          <div class="nav_wrapper Lightest_col">
            <div class="nav1">
              <p class="nav">Getting Started</p>
            </div>
            <div class="nav2">
              <p class="nav">Features</p>
            </div>
            <div class="nav3">
              <p class="nav">Procedures</p>
            </div>
            <div class="nav4">
              <p class="nav">FAQs</p>
            </div>
            <div class="nav5">
              <p class="nav">Tips</p>
            </div>
            <div class="nav6">
              <p class="nav">Company</p>
            </div>
          </div>
        </div>
      </div>
      <div class="HeroImage_img" id="computer_hero_area">
        <h1 class="Light_col">How Can We Help?</h1>
        <input class="centered" type="text" placeholder="Search">
      </div>
    </div>
  </div>
</body>
</html>
```

- ☆ We made many changes, such as the following (highlighted in yellow), to change the look of our preview:

```
<!doctype html>
<html lang="en">

<head>
  <title>TopNav</title>
  <script language="javascript"></script>
  <link rel="stylesheet" href="preview_styles.css">
  <link rel="stylesheet" href="Branding.css">
</head>

<body>
  <div id="main_area" class="FontFamily_font">
    <div id="computer">
      <div id="inside_computer">
        <div class="Lightest_bg" id="computer_top_banner">
          <div class="logo LogoImage_img">&nbsp;</div>
          <div class="nav_wrapper Medium_col">
            <div class="nav1">
              <p class="nav">Getting Started</p>
            </div>
            <div class="nav2">
              <p class="nav">Features</p>
            </div>
            <div class="nav3">
              <p class="nav">Procedures</p>
            </div>
            <div class="nav4">
              <p class="nav">Company Information</p>
            </div>
          </div>
        </div>
        <div class="HeroImage_img" id="computer_hero_area">
          <h1 class="Lightest_col">Knowledge Base</h1>
          <h2 class="Lightest_col">Your place for answers</h2>
          <div class="centered call_to_action Primary_bg">
            <p class="cta Lightest_col">Call to Action!</p>
          </div>
        </div>
        <div class="general_area Lightest_bg" id="computer_genera
          <div class="centered search_bar Dark_col Lightest_bg"
            <p class="search_text">Search articles</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

☆ And we edited the `preview_styles.css` file to account for our changes in the HTML file:

```
1  body
2  {
3      background-color: #f0f0f0;
4      margin: 0px;
5      font-family: arial;
6      overflow: hidden;
7      filter: blur(.035em);
8  }
9
10 h1
11 {
12     font-size: 18px;
13     font-weight: normal;
14     padding-top: 25px;
15 }
16
17 h2
18 {
19     font-size: 12px;
20     font-weight: normal;
21     margin-top: -5px;
22     margin-bottom: 20px;
23 }
24
25 p
26 {
27     font-size: 6px;
28     margin: 0px 15% 0px 15%;
29 }
30
31 p.cta
32 {
33     font-size: 9px;
34     margin-top: 6px;
35 }
36
37 .nav_wrapper
38 {
39     position: relative;
40     left: 125px;
41     top: -13px;
42     font-size: 9px;
43 }
44
45 .nav
46 {
47     font-size: 9px;
48 }
```

Of course, we made many more changes than these images show.

Step 14: Edit the FLPRJ File

1. Close your project template in Flare.
2. In Windows, open the main FLPRJ file in a text editor, such as Notepad++.
3. Edit the **Comment** area so that you have a custom description of your template.

```
<?xml version="1.0" encoding="utf-8"?>
<CatapultProject Version="1" xml:lang="en-us" SourceControlBound="false" Comment="This is a very fancy and
cool template that we like to call Balboa." MasterStylesheet="/Content/Resources/Stylesheets/MainStyles.css"
BrandingStylesheet="/Content/Resources/Branding/Branding.css" MasterStylesheetOverride="True">
</CatapultProject>
```





4. If necessary, copy the **MasterStylesheet** section, paste it, and edit it for a new **BrandingStylesheet** section as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<CatapultProject Version="1" xml:lang="en-us" SourceControlBound="false" Comment="This is a very fancy and
cool template that we like to call Balboa." MasterStylesheet="/Content/Resources/Stylesheets/MainStyles.css"
BrandingStylesheet="/Content/Resources/Branding/Branding.css" MasterStylesheetOverride="True">
</CatapultProject>
```

5. Save your changes.

I Step 15: Remove Unnecessary Folders

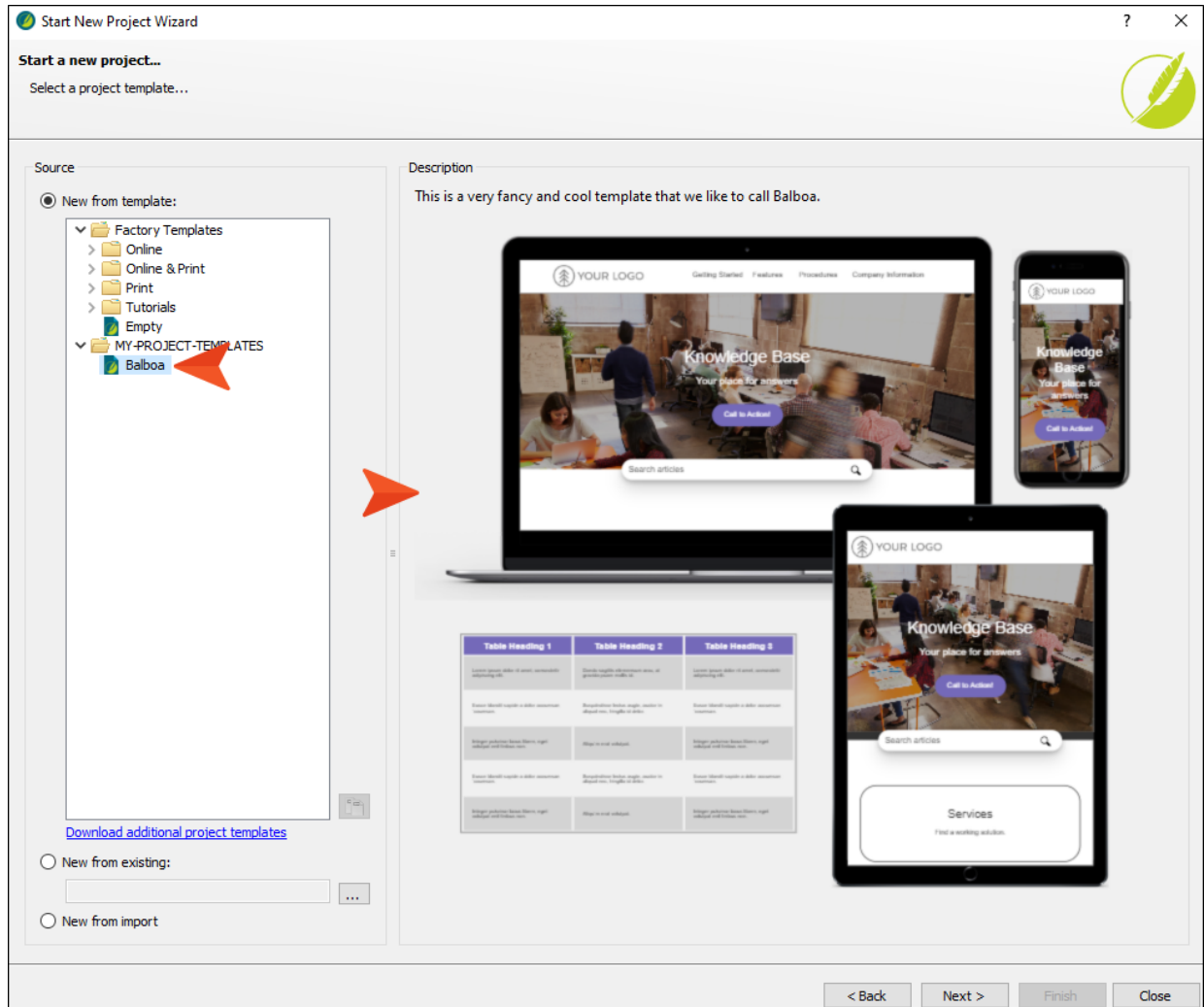
1. In Windows, delete the **Analyzer** and **Output** folders for your project template, so that you just have the Content, Preview, and Project folders at that level, along with the FLPRJ file.

Name	Date modified	Type	Size
 Content	8/22/2023 9:51 AM	File folder	
 Preview	8/22/2023 12:55 PM	File folder	
 Project	8/22/2023 10:02 AM	File folder	
 Balboa	8/22/2023 1:34 PM	FLPRJ File	1 KB

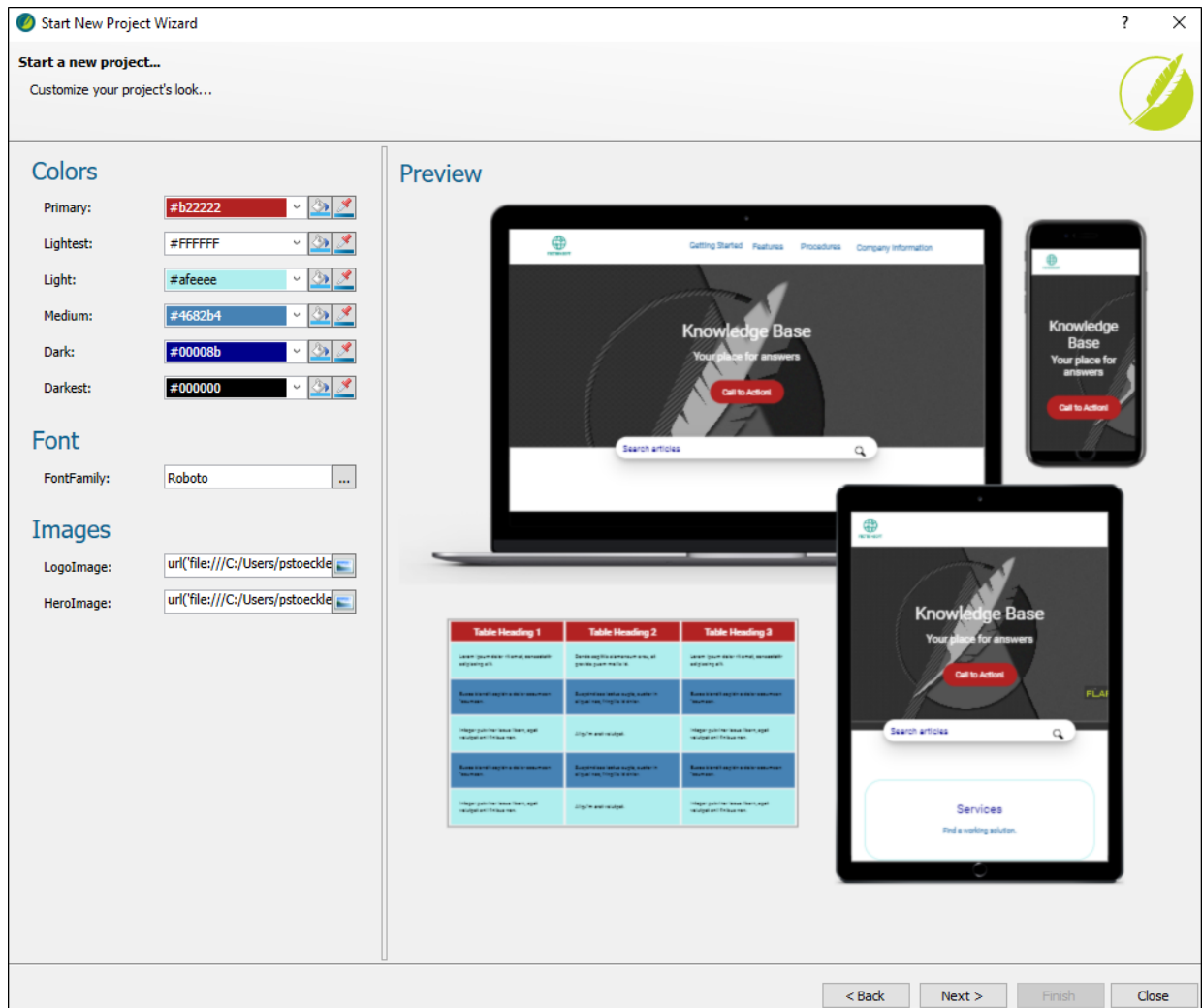
2. Open the Project folder and within it delete the **Users** folder.

Step 16: Test It

1. Open a new session of Flare.
2. Begin creating a new project and select your custom project template. Ours looks like this:



- On the next page of the wizard, change some of the values. The preview updates accordingly.



- After finishing the wizard, build the output and notice how it looks.

APPENDIX

PDFs

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

I Tutorials

Getting Started Tutorial

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

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

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