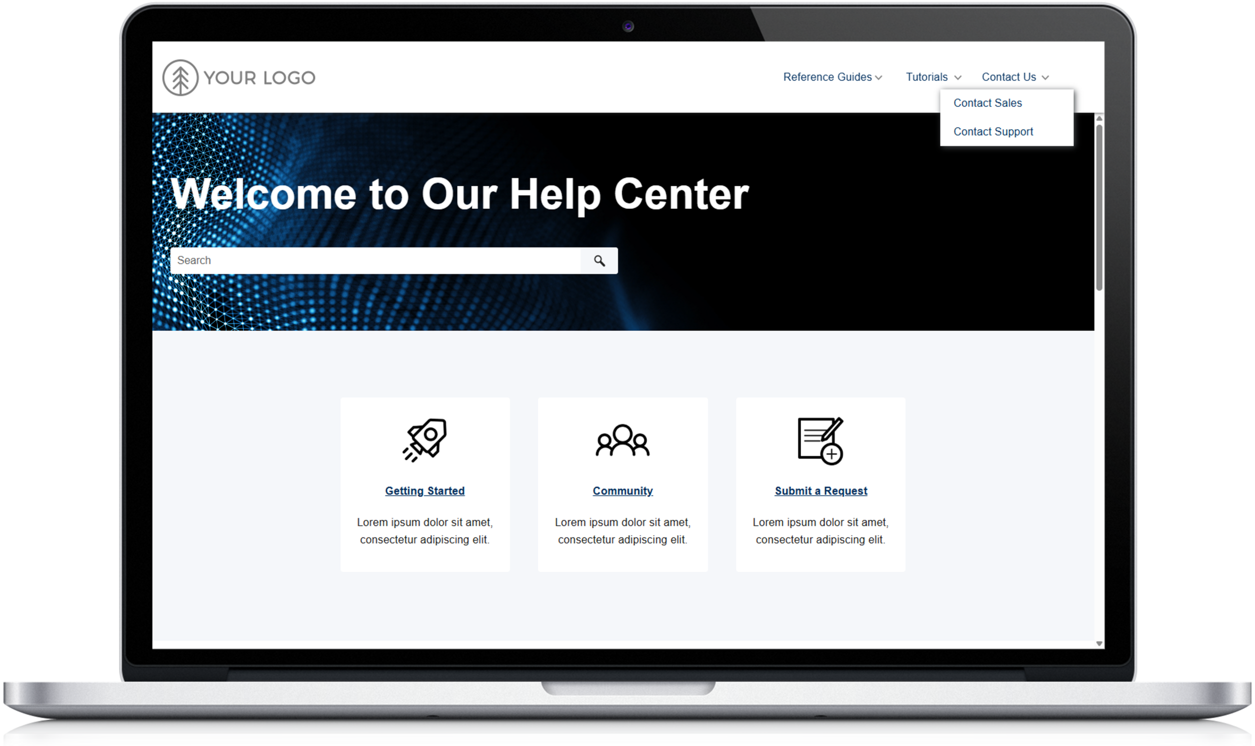
REFERENCE DOCUMENT

Laguna Template



Welcome to Our Help Center

Getting Started

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Community

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Submit a Request

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Table of Contents

Template Overview 3

Prerequisites 3

Saving the Project as a Template for Branding Preview 3

Creating a New Project from the Saved Template 4

Changing Branding Styles 6

Changing Logos 8

Changing the Logo in the Branding.css Stylesheet 8

Changing Hero Images 9

Custom Stylesheet Medium 10

Viewing or Modifying the HTML5 Medium in Flare 10

Viewing the Stylesheet in a Text Editor 11

Associating the HTML5 Medium with a Target 11

Toolbars 12

JavaScript 13

Custom Top Navigation 14

Links in New Tabs 16

Back to top 17

Dynamic Bookmarks 18

Clear Search 20

Collapsible Side Navigation 21

Resizable Side Navigation 22

Search Toggle 23

Changing Variables 24

Using this Template in an Existing Flare Project 24

Importing the Template Project 24

Template Overview

The Laguna Template is an advanced Flare project template that leverages CSS variables, responsive layouts, custom stylesheet mediums, and JavaScript. It also features a custom Top Navigation, along with the standard Side Navigation skin. This guide walks you through its key components and explains how to customize the template to suit your brand.

If you need assistance in implementing the templates in your project, please contact MadCap Technical Support at https://www.madcapsoftware.com/support/.

Prerequisites

Before we begin, there are some steps you should take to ensure that the template is setup properly on your computer.

Saving the Project as a Template for Branding Preview

The template project includes a **custom preview for the Branding Editor** that reflects the template's formatting. To enable this preview, you must first save the project as a template and copy the assets from the included **Preview** folder.

Although this step is optional, it is recommended. With the custom preview enabled, the Branding Editor will display exactly how the final output will look, making it easier to adjust branding styles.

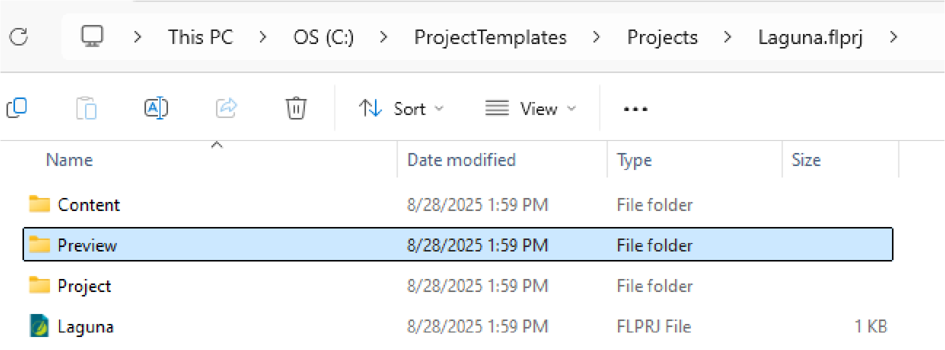
> **Note:** If you do not save the project as a template or copy the Preview assets, the Branding Editor will still function, but the preview will use Flare's default styling instead of your template's formatting.

To save the project as a template:

- On the Project ribbon, select Save Project as Template.
  - If you haven’t set up a Templates folder on your computer, see Creating Template Folders.

After saving the project as a template:

- Open Windows File Explorer and navigate to the location where the Laguna project was originally unzipped.
- Copy the **Preview** folder.
- Navigate to your **Templates > Projects** folder. Open the folder for the project template you created, then paste the **Preview** folder, replacing the one generated when you saved the project as a template.



This PC > OS (C:) > ProjectTemplates > Projects > Laguna.flprj >

Name	Date modified	Type	Size
Content	8/28/2025 1:59 PM	File folder	
Preview	8/28/2025 1:59 PM	File folder	
Project	8/28/2025 1:59 PM	File folder	
Laguna	8/28/2025 1:59 PM	FLPRJ File	1 KB

Creating a New Project from the Saved Template

After saving your project as a template, you’ll need to create a new project based on that template to enable the custom Branding Editor preview. This ensures the preview reflects the template’s actual formatting rather than default Flare styles.

> **Important:** The Branding Editor’s custom preview is tied to template-based projects. Saving the current project as a template does not automatically activate the preview for that same project—you must start a new project using the saved template so the preview is correctly linked.

To create a new project from your saved template:

1. In Flare, go to the File menu and select New Project.
2. In the New Project Wizard, select the From Template option.
3. Browse to your saved template in your Templates folder.
4. Select the template and click Next.
5. Enter a Project Name and choose a Project Location.
6. Complete the wizard to create the new project.

Once your new project is created, open the Branding Editor and confirm that the preview reflects your template’s styling.

Changing Branding Styles

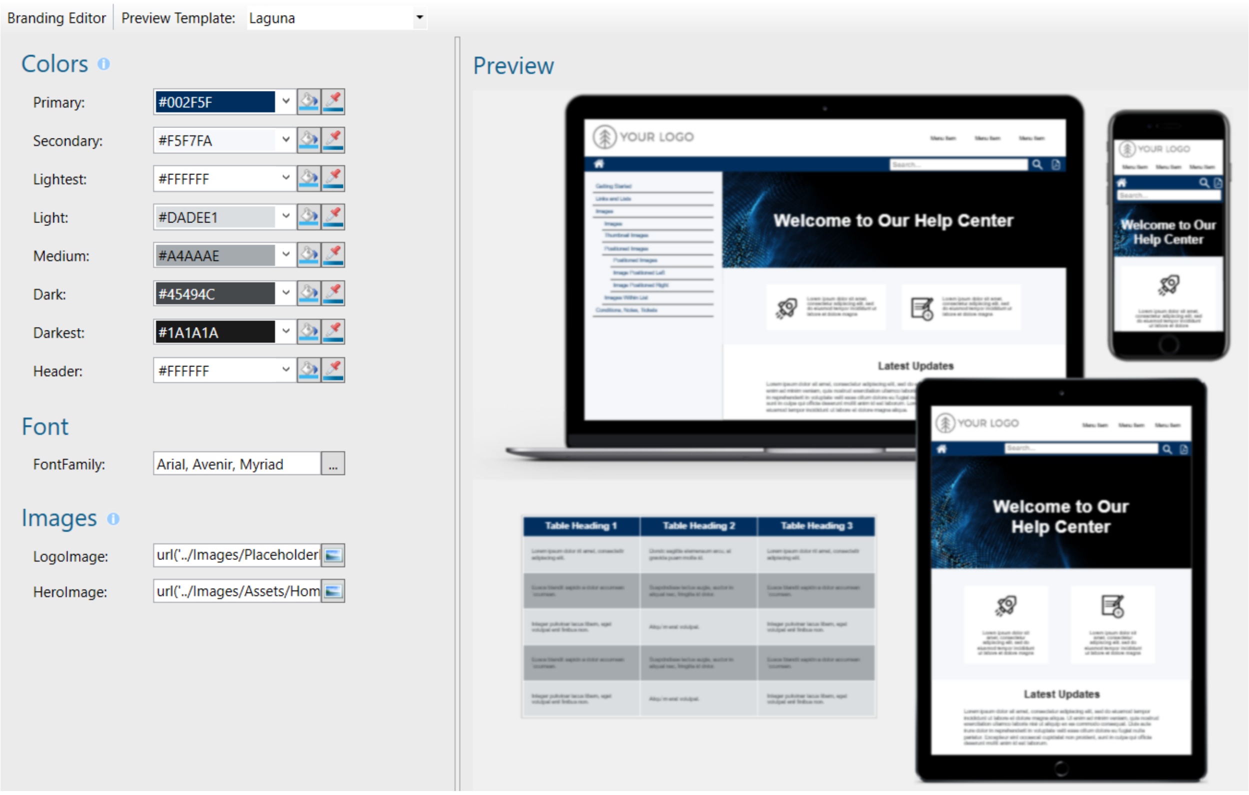
This template uses a branding stylesheet, which provides a centralized way to manage common branding elements such as logos, hero images, fonts, and color palettes. This approach ensures consistent styling across your project and outputs, with all updates controlled in one place. It’s especially helpful during a rebrand—simply update the branding stylesheet, and changes are automatically applied wherever those styles are used.

In Flare, branding stylesheets are stored as separate CSS files and make use of CSS variables.

CSS variables allow you to define reusable values—such as colors, fonts, and spacing—in one place and reference them throughout your project. They’re what make the branding stylesheet so powerful, enabling consistent styling, efficient updates, and reduced duplication. These variables can be edited through Flare’s Branding Editor, which provides an intuitive interface and a live preview so you can see changes in real time.

To access the Branding Editor, go to: **Content Explorer** > **Resources** Folder > **Branding** Folder > **Branding.css**.

Here you can update the branding stylesheet to match your corporate branding.



Branding Editor | Preview Template: Laguna

Colors

Primary: #002F5F

Secondary: #F5F7FA

Lightest: #FFFFFF

Light: #DADEE1

Medium: #A4AAAE

Dark: #45494C

Darkest: #1A1A1A

Header: #FFFFFF

Font

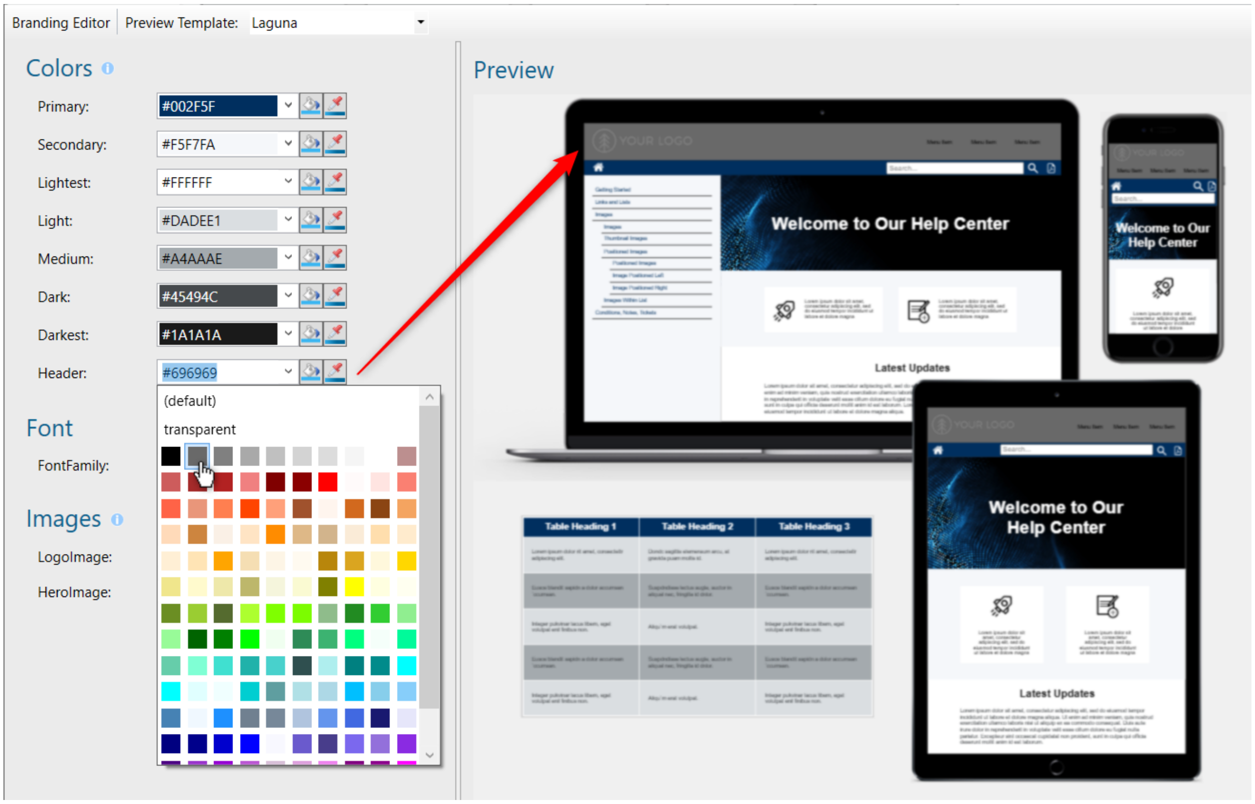
FontFamily: Arial, Avenir, Myriad

Images

LogoImage: url('../Images/Placeholder

HeroImage: url('../Images/Assets/Hom

Preview



Branding Editor | Preview Template: Laguna

Colors

Primary: #002F5F

Secondary: #F5F7FA

Lightest: #FFFFFF

Light: #DADEE1

Medium: #A4AAAE

Dark: #45494C

Darkest: #1A1A1A

Header: #696969

Font

FontFamily:

Images

LogoImage:

HeroImage:

Preview

Changing Logos

The logo is used in two files: **Side-Navigation.flskn** and **HomePage.flmsp**. Since this project uses CSS variables, the logo needs to be replaced in **Branding.css**.

> **Note:** The placeholder logo in the Laguna Template is **229 × 55** pixels. To ensure proper alignment and maintain visual balance within the header, we recommend resizing your custom logo to match the **same height (55px)**. The width can scale proportionally, but keeping the height consistent will help the logo fit seamlessly with the surrounding navigation and layout.

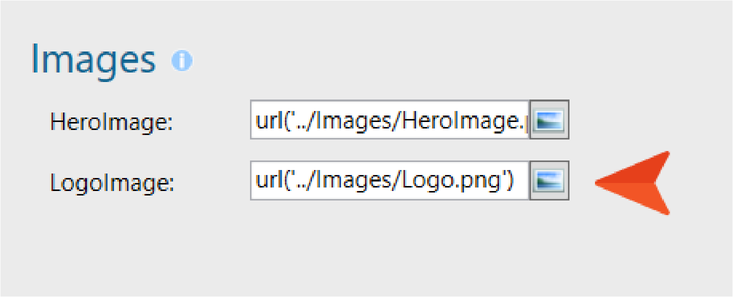
Changing the Logo in the Branding.css Stylesheet

To add a Logo image, do the following:

1. Open **Branding.css**

   Go to: **Content Explorer** > **Resources** Folder > **Branding** Folder > **Branding.css**.

2. In the Branding Editor, replace the image in the **LogoImage** CSS variable field with the desired Logo Image.



Images

HeroImage: url('../Images/HeroImage.

LogoImage: url('../Images/Logo.png')

Changing Hero Images

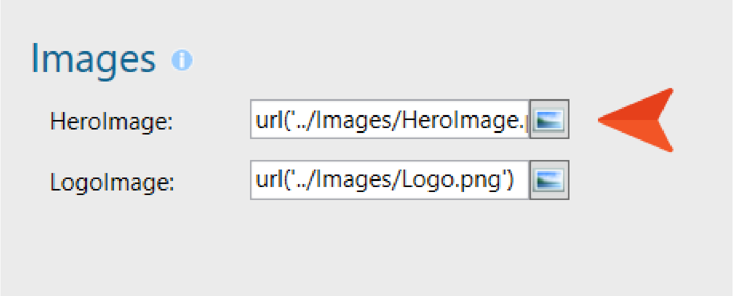
The hero image is the banner image that spans across the screen of the home page . The hero image used in this template is changed in **Branding.css**.

To change a hero image, do the following:

1. Open **Branding.css**

   Go to: **Content Explorer** > **Resources** Folder > **Branding** Folder > **Branding.css**.

2. In the Branding Editor, replace the image in the **HeroImage** CSS variable field with the desired Hero Image.

   Images

   HeroImage: url('../Images/HeroImage.

   LogoImage: url('../Images/Logo.png')

> **Note:** We recommend the hero image to have dimensions of **at least 1200 x 800 pixels**. Since this template includes white font color overlay on the hero images, a darker image will look best.

Custom Stylesheet Medium

The Laguna Template uses a custom stylesheet medium called **HTML5** to enable advanced styling, including **Flexbox** layouts and other CSS features that may not render correctly in Flare’s XML Editor.

This medium is linked to the target file, ensuring that:

- **Default medium** styles are used when editing in the XML Editor so content appears clean and readable during authoring.
- **HTML5 medium** styles are applied to the published output, adding layout enhancements and other advanced styling.

The HTML5 medium inherits from the Default medium, so the overall appearance remains consistent. The difference is that HTML5 styles are layered on top, allowing the additional design features to come through in the final output without affecting the authoring experience. For more information about stylesheet mediums, click here.

Viewing or Modifying the HTML5 Medium in Flare

- In the Content Explorer, open **Resources** > **Stylesheets**.
- Double-click **Styles.css**.
- In the Stylesheet Editor, open the Medium drop-down in the local toolbar.
- Select **HTML5** from the list.
- Make your desired style changes.
- Save the stylesheet.

Viewing the Stylesheet in a Text Editor

In addition to Flare’s built-in Stylesheet Editor, you can open the stylesheet in a text editor to see all CSS code, including the inline comments that explain what certain styles do.

To open the stylesheet in Flare’s Internal Text Editor:

1. In the Content Explorer, right-click the stylesheet (e.g., branding.css).
2. Select **Open With** > **Internal Text Editor**.

To open the stylesheet in another text editor (e.g., Notepad++, VS Code):

1. In the **Content Explorer**, right-click the stylesheet and select **Open with** and select the program from the list.
   - If your preferred program doesn't show in the list, navigate to the stylesheet in **Windows File Explorer** and open it with your preferred text editor.
2. Review the embedded comments for guidance on what each style controls. The **HTML5** medium will be at the bottom of the stylesheet.

Associating the HTML5 Medium with a Target

In the Project Organizer, open the Targets folder.

- Double-click the target you want to modify.
- In the Target Editor, select the **Advanced** tab.
- In the Stylesheet Medium field, choose **HTML5** from the drop-down list.
- **Save** the target.
- Rebuild your output to apply the changes.

Toolbars

This template uses a custom toolbar container with two toolbar skin components located in **Project Organizer** > **Skins**: **Toolbar-Left.flskn** and **Toolbar-Right.flskn**.

Additionally, there are two standalone buttons—the Home icon and Search icon—that are not part of the topic toolbar skin components.

- **Toolbar-Left.flskn** adds the previous and next buttons on the left side of the toolbar container.
- **Toolbar-Right.flskn** adds the print icon on the right side of the toolbar container.

To add more buttons, simply choose the appropriate skin component for the side of the toolbar where you want the button to appear. For more information, see Editing HTML5 Topic Toolbar Skin Components.

To change the Home or Search button icons:

- Open **Content Explorer > Content > Resources > TemplatePages > Other-Topics.flmsp**.
- Replace the existing icons with your desired images:
    - Right-click the image, select **Image Properties**, and choose the replacement image.

        > **Note:** The template icons are **24px** × **24px**. It is recommended that all toolbar icons match this size.

JavaScript

The Laguna Template uses JavaScript (JS) to add dynamic functionality to the output. The JS files are located in Content > Resources > Scripts.

The JavaScript file is structured for clarity and maintainability. Functions are written as named functions, which makes the code modular and reusable across different parts of the output. At the top of the file, an initialization block using **$(document).ready()** is included. This block is responsible for calling the named functions in sequence once the page has finished loading, ensuring that all dynamic features are initialized in the correct order.

This organization has several benefits:

- **Modularity** – Each feature (such as bookmarks, navigation enhancements, or dropdown menus) is encapsulated in its own function.
- **Reusability** – Functions can be reused across different template pages or projects if needed.
- **Initialization control** – By centralizing the function calls in **$(document).ready()**, you can easily manage which features are enabled or disabled, simply by adding or removing calls to specific functions.

This section explains both the purpose of each script and how to adjust the styling of elements controlled by JavaScript. Styling is typically handled via CSS, but many of the JS-driven features add classes or attributes dynamically, so understanding these interactions makes it easier to fine-tune the appearance of your output.

Custom Top Navigation

The Laguna Template features two navigation systems in the HTML5 output:

- **Side Navigation** – Provided by the selected skin and populated automatically by the primary TOC assigned to the target. This serves as the main hierarchical navigation for browsing the full set of topics.
- **Top Navigation** – A custom menu created using a topic menu that points to a separate TOC in the project (**Project/TOCs/Top-Navigation.fltoc**). This allows the top navigation to display its own curated set of links, independent from the side navigation.

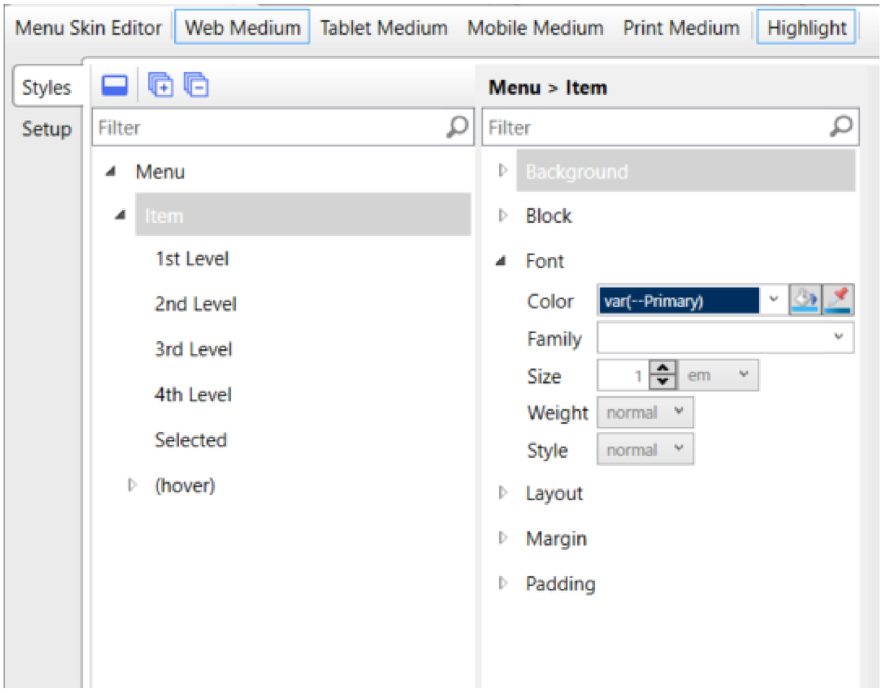
The top navigation menu is saved as a reusable snippet (**Content/Resources/Snippets/Header-link-menu.flsnp**) and included on the template pages, so any updates to the custom TOC automatically flow into all pages using that snippet. CSS and JavaScript are used to position and style the top navigation so it integrates seamlessly with the output's layout.

The top navigation is also fully accessible by keyboard, meaning users can navigate through the menu items using the Tab key. This ensures that the menu is usable not only by mouse but also for those relying on keyboard navigation or assistive technologies.

This dual-navigation approach gives you flexibility—keeping a detailed, structured navigation in the side panel while offering quick-access links or high-level categories in the top menu, all controlled by separate TOCs for easy management.

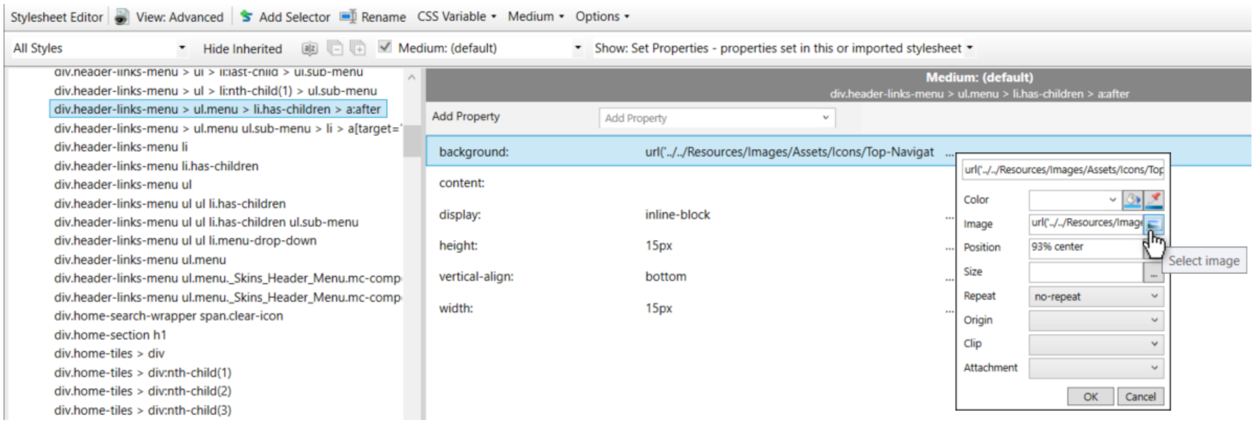
Styling Notes:

To customize the fonts, colors, hover effects, and other styling for the top navigation menu, open **Project/Skins/Header-Menu.flskn**. Then go to the **Styles** tab and select **Menu** > **Item** to adjust the appearance.



Color: var(--Primary)
Family:
Size: 1 em
Weight: normal
Style: normal

To change the icon next to drop-down items in the top navigation, open **Styles.css** and select **(Complex Selectors)** > **div.header-links-menu > ul.menu > li.has-children > a:after**.



☑ Medium: (default)

Show: Set Properties - properties set in this or imported stylesheet

div.header-links-menu > ul > li:last-child > ul.sub-menu
div.header-links-menu > ul > li:nth-child(1) > ul.sub-menu
div.header-links-menu > ul.menu > li.has-children > a:after
div.header-links-menu > ul.menu ul.sub-menu > li > a[target='
div.header-links-menu li
div.header-links-menu li.has-children
div.header-links-menu ul
div.header-links-menu ul ul li.has-children
div.header-links-menu ul ul li.has-children ul.sub-menu
div.header-links-menu ul ul li.menu-drop-down
div.header-links-menu ul.menu
div.header-links-menu ul.menu._Skins_Header_Menu.mc-comp
div.header-links-menu ul.menu._Skins_Header_Menu.mc-comp
div.home-search-wrapper span.clear-icon
div.home-section h1
div.home-tiles > div
div.home-tiles > div:nth-child(1)
div.home-tiles > div:nth-child(2)
div.home-tiles > div:nth-child(3)

Medium: (default)
div.header-links-menu > ul.menu > li.has-children > a:after

background: url('../../Resources/Images/Assets/Icons/Top-Navigat ...
content:
display: inline-block
height: 15px
vertical-align: bottom
width: 15px

url('../../Resources/Images/Assets/Icons/Top
Color:
Image: url('../../Resources/Image
Position: 93% center
Size:
Repeat: no-repeat
Origin:
Clip:
Attachment:

Links in New Tabs

The template includes CSS that adds an indicator icon to top navigation links configured to open in a new tab. For more details, see Specifying Browser Frames for TOC Entries .

To change the icon for links opening in a new tab, open **Styles.css** and select **(Complex Selectors)**, then locate the selector **div.header-links-menu > ul.menu ul.sub-menu > li > a[target="_blank"]:after**, and update the value of the **background-image** property.

Back to top

Purpose:

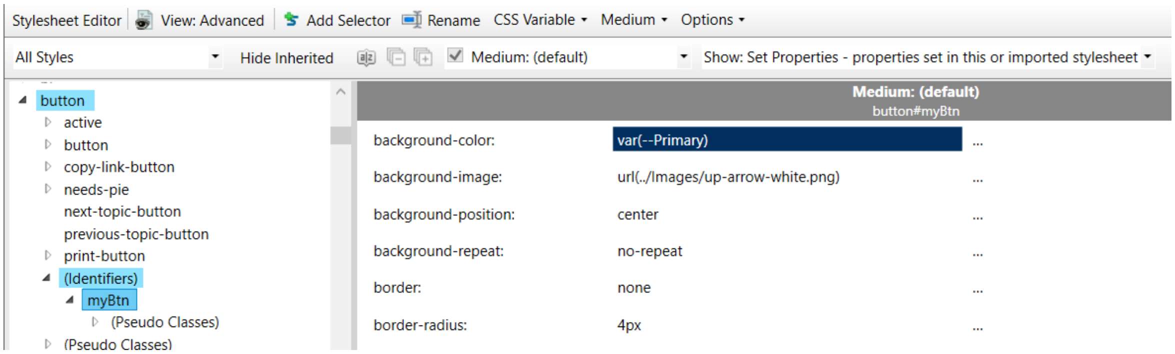
This function adds a "Back to Top" button that appears when users scroll down the page, offering a convenient way to quickly return to the top without manual scrolling. It improves navigation efficiency, especially on long pages or complex content where scrolling back manually can be tedious. Smooth scrolling creates a polished user experience by gently guiding the user’s focus rather than a jarring jump.

Key Features:

- Dynamically adds the button to the output.
- Shows button after scrolling 20px down.
- Smooth scroll animation on click.

Styling Notes:

- Style the button with the ID **myBtn** (button#myBtn) for icon, position, color, size, and hover effects.



Medium: (default)
button#myBtn

background-color:	var(--Primary)
background-image:	url(../Images/up-arrow-white.png)
background-position:	center
background-repeat:	no-repeat
border:	none
border-radius:	4px

Dynamic Bookmarks

Purpose:

This function creates clickable anchor links on headings, or any selector that you choose, enabling users to easily share or bookmark specific sections of content. This improves navigation and collaboration by making it straightforward to link directly to relevant content points. It also supports dynamic anchor generation for headings without IDs, improving overall linkability and SEO. The visual anchor icon helps users discover this functionality intuitively.

Key Features:

- Uses existing anchor IDs or generates them from heading text.
- Prepends a small clickable icon (#anchor-icon) for copying links.
- Updates menu links to match anchor changes.
- Supports smooth scrolling to anchors on page load.

Customizing Dynamic Bookmarks:

By default, the HeadingBookmarks() function in Laguna.js adds dynamic bookmarks to all h1, h2, and h3 elements in the output. These bookmarks create linkable anchors so that users can easily reference specific sections of a topic.

You can customize which elements receive bookmarks by editing the function call in Laguna.js:

1. Open Laguna.js (located in Content > Resources > Scripts).
2. At the top of the file, you’ll see a $(document).ready() block where named functions are called.
3. Look for the call to HeadingBookmarks(), which by default looks like this:

```
HeadingBookmarks("h1,h2,h3");
```

4. The values inside the parentheses determine which elements receive bookmarks. This list accepts any valid CSS selectors.

For example:

- To apply bookmarks only to h2 and h3, remove h1:

```
HeadingBookmarks("h2,h3");
```

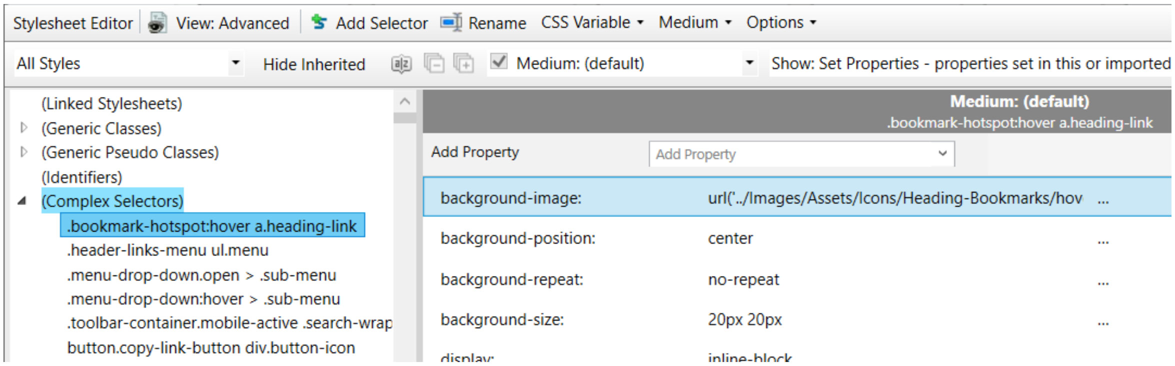
- To apply bookmarks to a different set of elements, such as all h4 and paragraphs with the class .section-title:

```
HeadingBookmarks("h4,p.section-title");
```

This flexibility allows you to control exactly which elements in your output are linkable, ensuring the bookmark functionality aligns with your content structure.

Styling Notes:

- To change the icon, or customize the look of the anchor icon, open your stylesheet and select **(Complex Selectors)** > **.bookmark-hotspot:hover a.heading-link**



☑ Medium: (default)

Show: Set Properties - properties set in this or imported

(Linked Stylesheets)
(Generic Classes)
(Generic Pseudo Classes)
(Identifiers)
(Complex Selectors)
.bookmark-hotspot:hover a.heading-link
.header-links-menu ul.menu
.menu-drop-down.open > .sub-menu
.menu-drop-down:hover > .sub-menu
.toolbar-container.mobile-active .search-wrap
button.copy-link-button div.button-icon

Medium: (default)
.bookmark-hotspot:hover a.heading-link

background-image:	url('../Images/Assets/Icons/Heading-Bookmarks/hov	...
background-position:	center	...
background-repeat:	no-repeat	...
background-size:	20px 20px	...
display:	inline-block	

Clear Search

Purpose:

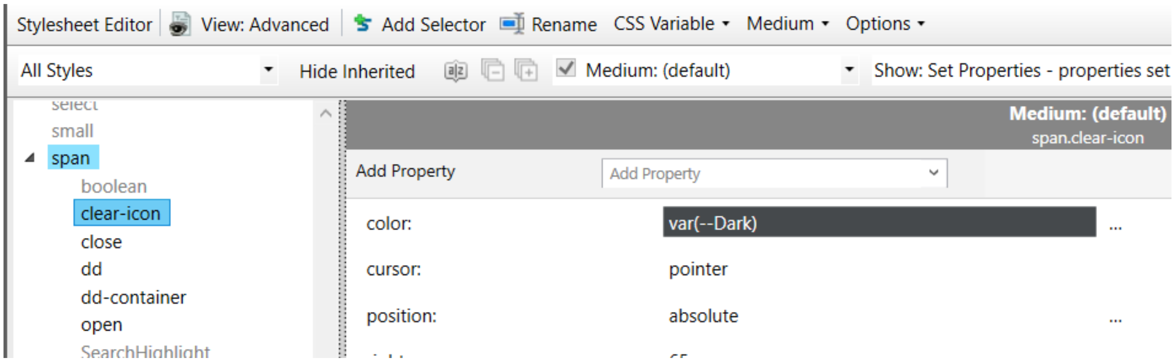
This function adds a clear (x) button to the search field, allowing users to quickly reset their search input with a single click. This reduces friction during searching, especially when users want to start a new query or correct their input.

Key Features:

- Shows/hides clear icon based on input content.
- Clears and refocuses input when clear icon is clicked.

Styling Notes:

- To adjust the styling of the clear search button, select **span** > **clear-icon** (span.clear-search) in the stylesheet.



Medium: (default)
span.clear-icon

color: var(--Dark)

cursor: pointer

position: absolute

Collapsible Side Navigation

Purpose:

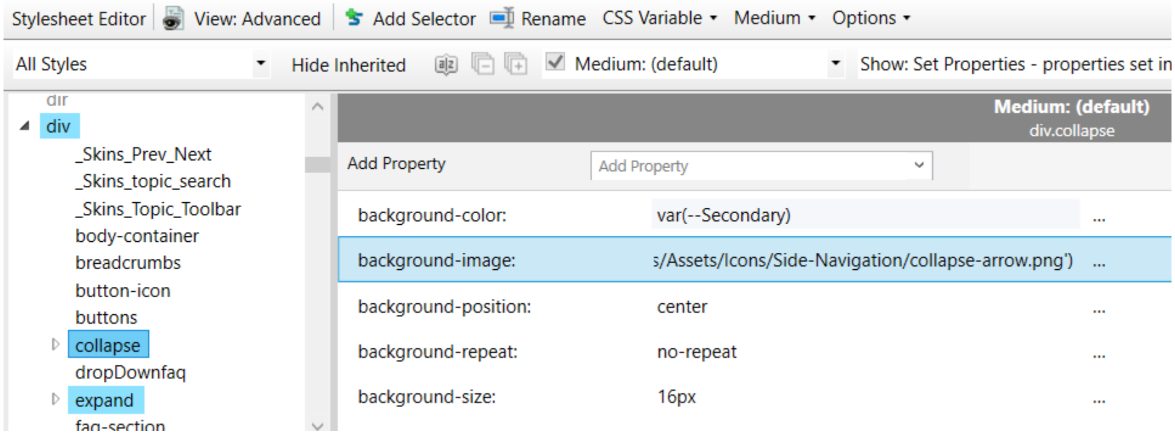
This function allows users to collapse or expand the side navigation panel, giving them control over the interface layout to reduce clutter and focus on main content. Persisting the collapse state with localStorage means the user's preference is remembered across sessions, enhancing usability and personalization. This is especially helpful in large documentation sets where users may want to hide navigation temporarily without losing their place.

Key Features:

- Toggles side nav visibility with slide animation.
- Stores expanded/collapsed state for persistence.

Styling Notes:

- Style **div** > **collapse** and **expand** classes for toggle button appearance, such as background-color or icon changes.



background-color: var(--Secondary) ...
background-image: s/Assets/Icons/Side-Navigation/collapse-arrow.png') ...
background-position: center ...
background-repeat: no-repeat ...
background-size: 16px ...

Resizable Side Navigation

Purpose:

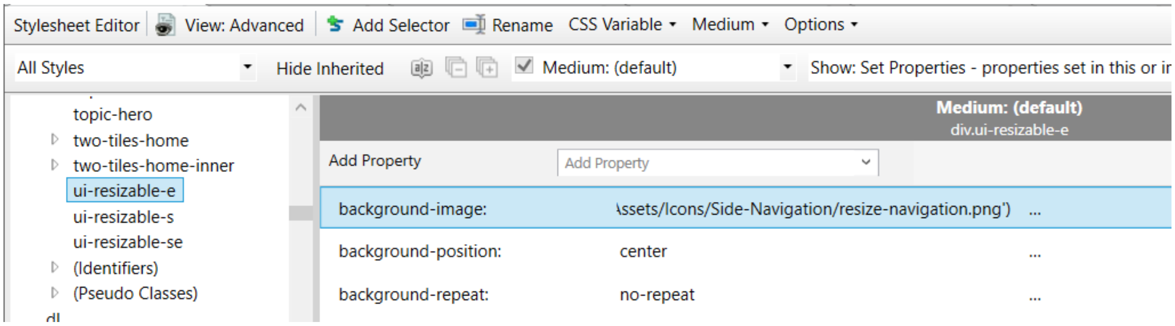
This function allows users to drag-and-drop to adjust the width of the side navigation panel. It uses the jQueryUI Resizable library, which has been linked to the OtherTopics.flmsp template page. There is also a an initializer JS file (Content/Resources/Scripts/resize-initializer.js) which is used to call the resize function on the side navigation wrapper. It also controls the minimum and maximum width that the side navigation panel is allowed to be.

Key Features:

- Drag-and-drop resize of side navigation
- Customizable minimum and maximum width for the side navigation

Styling Notes:

- Style **div** > **ui-resizable-e** (div.ui-resizable-e) to set the icon (background-image), as well as size and positioning of the icon.



Medium: (default)
div.ui-resizable-e

background-image:	\ssets/Icons/Side-Navigation/resize-navigation.png')
background-position:	center
background-repeat:	no-repeat

Search Toggle

Purpose:

This function controls the visibility of the search bar within the toolbar, allowing users to toggle it on or off as needed. Remembering the search bar’s state during a browsing session provides a consistent and personalized experience. Supporting both mouse and keyboard input ensures accessibility and ease of use across different devices and user preferences.

Key Features:

- Animates show/hide except on mobile devices.
- Toggles .mobile-active class on toolbar when visible.
- Supports keyboard toggling (Enter/Space).

Styling Notes:

- Style .search-wrapper for visibility and transitions.
- Style .toolbar-container.mobile-active to indicate active search.
- Customize .search-bar-icon toggle button appearance.

Changing Variables

All company information is set in the **General** variable set.

1. Open the **General** variable set.

   Go to: **Project Organizer** > **Variables** Folder > **General**

2. In the Variable Set edit definitions to replace with your own manual and contact information.
3. Save the variable set.

Using this Template in an Existing Flare Project

In order to use this template, files need to be imported into your existing Flare project.

Importing the Template Project

1. Open the existing Flare project.

   > **Important:** Before continuing these steps make sure to create a backup of your existing Flare project. To easily create a backup go to the **Project** Ribbon > **Zip Project**.

2. Go to **Project** Ribbon > **Import** > **Flare Project**.
3. Select the ... button to browse and open the template project.
4. Select **Next**.

5. In the **Include Files** field, copy and paste one of the two options:

   To include the files that are necessary to display the template layout as is:

   ```
   *.png;*.jpg;*.svg;*.flmsp;*.css;*.fltar;*.flvar;*.flskn;
   *.js
   ```

6. Enable **Auto-include linked files**.

7. Select **Finish**.

8. The **Accept Imported Documents** window will appear. Review the files that will be imported into your Flare project.

   > **Warning:** If there are file names in this template project that are the same in your Flare project, Flare will prompt that files will be overwritten.
   > To continue the import without overwriting project files, rename the files in your Flare project or in the template project.

   > **Important:** Rename files from within Flare in order to maintain links between files.

9. Select **Accept**.

10. Build and View the imported **HTML5** target.

> **Note:** After importing the template project, imported files will have a 🔗 icon overlay. This means that a link is maintained with the imported project. If you do not want to keep the link and want to remove the icon overlay, delete the import file. To do this, use the **Project Organizer** and go to the **Imports** folder, then delete the **Laguna Template** file.