CHEAT SHEET // FLARE DESKTOP 2025

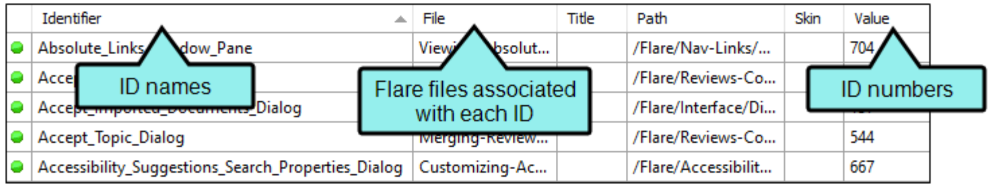
Context-Sensitive Help

Concepts

Alias File An alias file is used to populate a header file with the information necessary for producing context-sensitive Help (CSH). After opening an alias file, you can use the Alias Editor to create and assign identifiers (IDs) for the header file. You can use a single alias file in a project for multiple header files, or you can create a separate alias file to go with each header file.

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

Identifier (ID) A CSH ID is text (i.e., name) or a number (i.e., value) that connects a particular topic or micro content response with the corresponding part of the software interface. You do not need to use both the ID name and value; one is enough. By pointing to the ID (instead of the full path of the topic or micro content), you can be assured that the link will always work in case the documentation file is later renamed or moved.



	Identifier	File	Title	Path	Skin	Value
●	Absolute_Links[illegible]dow_Pane	Viewi[illegible]bsolut...		/Flare/Nav-Links/...		704
●	Acce[illegible]	[illegible]		/Flare/Reviews-Co...		[illegible]
●	Acce[illegible]_Dialog	[illegible]		/Flare/Interface/Di...		[illegible]
●	Accept_Topic_Dialog	Merging-Review...		/Flare/Reviews-Co...		544
●	Accessibility_Suggestions_Search_Properties_Dialog	Customizing-Ac...		/Flare/Accessibilit...		667

Process
1. **Author** Add a header file to the project. 2. **Author** Add an alias file to the project. The alias file lets you populate the header file through a user-friendly interface. 3. **Author** Create and assign identifiers (IDs). (Optional for HTML5) Create a skin to use in CSH, set an ID for it, associate it with the appropriate IDs, and in the target set all skins to be generated. 4. **Author** Associate the alias file with the target. This is necessary only if you have created more than one alias file. 5. **Author** Provide the developer with a copy of the header file (or at least the CSH IDs within it) and the skin ID (if you are using one). 6. **Author** Generate output and provide the developer with a copy of the output files. 7. **Developer** Make CSH calls using information in the header file and the skin ID, if applicable. 8. **Developer** Create a new build of the software application. 9. **Author** Install the new software build and test the CSH.

How To	
Set CSH Options (Automatically applies settings to new IDs)	1. In the local toolbar of the Alias Editor click . 2. Complete any of the options (e.g., starting value, prefix), and click **OK**.
Automatically Create Many CSH IDs	1. In the local toolbar of the Alias Editor click . 2. In the **Header File** area, select whether to create a new header file or choose an existing one. 3. In the **Identifier Options** area, you can override values. 4. In the **Generate Identifiers for** field, select one of the options. 5. Choose to **Keep** or **Delete** existing IDs. 6. Click **Create**.
Manually Create a CSH ID	1. In the Alias Editor toolbar, select the header file (if you have more than one) 2. On the left side of the editor, right-click a topic or micro content phrase, and select **Assign to New Identifier**.

Making CSH Calls (the Developer's Part)

Method 1: JavaScript

Add a reference to the JavaScript file (which is created automatically when the author builds the output). This .js file should be named "csh.js."

```
<script type="text/javascript" src="[path of file]/csh.js"></script>
```

Create a trigger and add the command to let users open the Help system. Following is an example.

```
<input type="button" value="Open Help System" onclick="MadCap.OpenHelp(ID, skin name,
search string, first pick search string value );" />
```

Method 2: URL

Create a trigger and add a link to let users open a specific area of the online output.

Following is the basic structure (although some of the information shown after # is optional).

```
[main entry file].htm#cshid=[ID number, ID name, or topic
path/name]&searchQuery=[search string]&firstPick=true
```

A simple structure for HTML5 output would point to the main Default.htm entry file, followed by the CSH ID name. In the following example, "Soccer" is the name.

```
<a href="http://my.mycompany.com/Default.htm#cshid=Soccer">Click here
to open</a>
```