MADCAP FLARE 2019 r2

Key Features
Flare has many features in it. Over time, you will find that some of these are more important to you than others. Following are some important characteristics of Flare, including key concepts and features that are especially important for most authors and ones that you are likely to use. See the online Help for more information about each feature.

<table>
<thead>
<tr>
<th>Category</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>4</td>
</tr>
<tr>
<td>Authoring</td>
<td>9</td>
</tr>
<tr>
<td>Design</td>
<td>19</td>
</tr>
<tr>
<td>Importing</td>
<td>23</td>
</tr>
<tr>
<td>Exporting</td>
<td>24</td>
</tr>
<tr>
<td>Multi-Channel Publishing</td>
<td>25</td>
</tr>
<tr>
<td>Search and Micro Content</td>
<td>26</td>
</tr>
<tr>
<td>Project Management</td>
<td>27</td>
</tr>
</tbody>
</table>
General

Following are some general key concepts and features in Flare.

XHTML Authoring

In Flare, content is authored in XHTML (Extensible Hypertext Markup Language), which is a cross between XML and HTML. XML is a standard developed by the World Wide Web Consortium (W3C) and is intended as a replacement standard for HTML to render documents on the World Wide Web. XML is not a fixed set of elements like HTML, but rather a metalanguage (a language for describing languages). It enables authors to define their own tags.

For anyone unfamiliar with XHTML, this might sound intimidating at first. But even if you do not know anything about XHTML, XML, or HTML, you can create your content in Flare’s main content editor (the XML Editor) in much the same way you would use a tool such as Microsoft Word. The XHTML markup is automatically created for you behind the scenes.
If you happen to be experienced with XHTML, you will be glad to know that you can work in the markup, taking advantage of all of the benefits of this structured language.

Single-Sourcing

One of Flare’s biggest strengths lies in single-sourcing, which means to reuse content, and producing multiple outputs from the same set of source files. Flare lets you single-source your projects in many ways, using various features.

This includes (but is not limited to):

- “Building Blocks and Topic-Based Authoring” on the next page
- “Snippets” on page 11
- “Variables” on page 12
- “Conditions” on page 13
Building Blocks and Topic-Based Authoring

In other authoring tools, you’re probably used to everything being a part of a single file—the content, the table of contents, the glossary, the styles, and so on. It’s not like that in Flare. Instead, most of the pieces are separate, sort of like building blocks. This is one of the things that helps to make Flare so powerful and give you so much flexibility in how to create your output.

Your content is stored in topic files and in smaller snippet files. And images exist as separate files and are included in topics and snippets by reference. Your table of contents is a different file, as is the glossary. The styles are stored in a separate cascading stylesheet file. In fact, you might even have a project with multiple TOCs, multiple glossaries, and multiple stylesheets. It all depends on how you want to work.
Indexes are different because they are created in part by inserting index keywords into topics and snippets. There is no separate index file in the Flare project. However, for the most part, you’re dealing with separate files as building blocks in Flare.

Then you use another file called a “target” to take all of these pieces, merge them together, and generate the output you want. It’s kind of like putting a bunch of ingredients into a food processor and ending up with a smoothie.

Topics are where you type your text and other content. If you come from the world of print publishing, you’ll probably be tempted at first to create really long topics—such as one topic for each chapter or even one topic for an entire PDF manual. And while Flare lets you create very long topics and there may be cases where you need to do that, for the most part, you want your topics to be smaller, digestible chunks of content that you can reuse for different outputs. One chapter in a PDF that you create from Flare will usually consist of several separate topics.

But don’t make your topics too short either. We’re not necessarily talking about single sentences or short paragraphs. You want your topics to have enough substance to stand on their own, but short enough to be able to easily reuse them in different places in various outputs.
Open File Architecture

All files in a Flare project are separate XML documents. This means that Flare projects are completely open, transparent, and accessible. One great benefit of this is easier customization of your project and files via scripting.

Schema

A schema is a collection of metadata that describes the elements in an XML document. Flare’s document schema is a hybrid between XSD (XML Schema Definition) and a custom schema created by MadCap Software to account for the unique tags and styles required to support all of Flare’s features. Therefore, the Flare schema conforms to industry standard requirements, recommended by the World Wide Web Consortium (W3C). This means Flare content can be edited in the Flare XML Editor, in external editors, or transferred back and forth between the two at will. Also, because Flare’s code adheres to the W3C specification, it is easier to integrate with other XHTML applications.
Authoring

Following are some key authoring features in Flare.

Cross-References

A cross-reference is a navigation link that lets you connect text in one topic to another topic (or a bookmark within a topic). This is somewhat similar to a text hyperlink. However, cross-references are more powerful in that the links can automatically be updated based on commands (e.g., appears as a text link in online output, but page numbers in print-based output).

The best practice is to use cross-references to create links between files and locations inside a Flare project, and hyperlinks are preferred for links that point outside of the project (e.g., websites, external PDFs). There are exceptions to this, but most of the time this is the recommended approach.
Link Viewer

In Flare there are many ways to link one file to another, such as inserting a cross-reference or text hyperlink, inserting an image, applying a stylesheet to a topic, and more. The Link Viewer window pane (View > Link Viewer) lets you see how your different files are connected and may be one of the most useful tools you use in Flare.
Snippets

A snippet is a chunk of formatted content that is heavily used in single-sourcing. Snippets can include text, tables, images, and whatever else can be included in a normal topic. You can insert snippets into one or more topics throughout your project, thus allowing you to reuse content that is maintained in one place. You can even insert them into other snippets, creating nested snippets. Snippets are not usually intended for single words or very short phrases. In those cases, you probably want to use variables instead.

The major benefit of using snippets is that you only have to create your content once, rather than having to type the same information in each topic where you want to use it. If you need to modify the content of a snippet, you only need to change it in one place and the change is made automatically in every topic where the snippet has been inserted.
Variables

A variable is a brief, non-formatted piece of content that can be edited in one place but used in many places throughout your project. They’re especially good for text that might change frequently, such as version numbers and dates. Variables are stored in variable sets, which can hold multiple variables. Depending on the template you select when creating a project, Flare may provide you with an initial variable set, but you can add as many additional variable sets and variables as you like.

Images

You can insert images into content files (e.g., topics, snippets). Flare supports the following types of raster and vector image files: BMP, EMF, EPS, EXPS, GIF, HDP, JPG, JPEG, PDF, PNG, PS, SVG, SWF, TIF, TIFF, WDP, WMF, XAML, XPS. One of the quickest ways to insert an image is to drag it from the Content Explorer into the open content file. You can also apply styles to images to affect their look (e.g., resizing, thumbnail popup).
Multimedia

You can embed Flash, Windows Media, Quicktime, and HTML5 files, as well as videos from your YouTube or Vimeo account. In addition, you can insert links to movies created in MadCap Mimic. If you want to enhance your output using 3D, Flare supports Universal 3D (U3D) files. These files allow you to show users an interactive 3D model in print or web output.

Also, numerous multimedia files are supported in PDF output. This means that those files play when viewing the PDF in electronic format. If you print the PDF, those files are simply displayed as static images.

Conditions

A condition is a single-sourcing feature that you can apply to files or to different areas of your content, so that some information displays in some outputs but not in others, or it displays only when viewed on a certain device or screen size. For example, maybe you need to produce both online and print-based output. Much of the content you create is going to be the same for both outputs, but some of it is going to be written only for online output, and some only for print-based output. So you can create condition tags for each and use those tags to separate the content when you build your output.

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

Creating and editing a table of contents file in Flare can be very easy to do for both online and print output. You can drag topics from the Content Explorer to the TOC Editor. You can also manually add TOC books and items, and then link them to other files. The links usually point to topics, but for online output they can also point to external files, other Help systems, and movies. You put all of these books and items in a structure that you think would be useful for the individual. In online output, end users browse through a TOC to find information. And in HTML5 Side and Top Navigation output, the TOC items become links in menus.

The TOC files you see in the Project Organizer work differently for online output than they do for print-based output. For online outputs, TOC files are exactly as their name suggests; they are files that create TOCs or menus in the output. But for print-based outputs, that same TOC file functions more like an outline. The element that actually generates a TOC in print-based output is called a “proxy,” which is inserted into a topic. You can manually create that proxy yourself, or you can select an option in the Advanced tab of the Target Editor and let Flare do it all for you. There are pros and cons for both methods.
Context-Sensitive Help

Context-sensitive Help (CSH) is a way to tie your existing topics with specific dialogs or windows in a software application, or with simple web links created somewhere (e.g., on a website). This is done with a map ID. When users open a particular dialog or window in a software application, or click a web link, they can quickly open a topic pertaining to it.

Responsive Web Design

Responsive web design (RWD) is a way to construct your HTML5 output so that the display and content are adjusted automatically depending on the device. Therefore, on tablets and smart phones, users will see a condensed look and perhaps different text that is more appropriate for those devices, compared with larger monitors. You can get the same effect if you shrink your browser to a smaller size.

There are two areas where RWD can be applied: (1) skin and (2) content.

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

Depending on the output type, you can make your content responsive to different devices and screen sizes by using responsive conditions and responsive layouts.
When you break down the authoring process in Flare, you will discover that it can be quite simple. Following are the basic steps that you need to follow for creating and developing a project in Flare.

**BASIC STEPS**

**RECOMMENDED FOR NEW USERS**

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

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

- **Getting Started Video** The Getting Started Video provides a visual demonstration of how to create a new project, introduces basic information about Flare, and points you to additional videos and resources to further develop the project.

- **Getting Started Guide** The Getting Started Guide is a PDF that explains how Flare works and introduces you to some of its key features and concepts. It then outlines the five basic steps for developing a project and producing output.

- **Getting Started Tutorial** The Getting Started Tutorial is a PDF file that you can use alongside Flare to create, develop, and generate output from an actual project.
Mobile layout. Now that the screen is smaller, notice the menus in the skin are now in a flyout, and the hyperlinked images are slacked.
Dynamic Preview

You can use the Preview window pane to see a quick preview for a topic, snippet, or master page. The Preview window pane is dynamic, allowing you to keep the preview open while you work and see changes as you make them in the XML Editor.

Initially, the window pane opens as a floating window. This can be quite useful if you are working with dual monitors, because you can drag the Preview window pane to one monitor while editing the topic in the other monitor. However, you always have the option of docking the Preview window pane to the interface.
Design

Following are some key design features in Flare.

Cascading Stylesheets

Styles are used to control the look and feel of your documentation, and keep the content separate from its presentation. The styling is based on Cascading stylesheets (CSS), which is an international standard for formatting web content, developed by the World Wide Web Consortium (or W3C).

Flare lets you work with stylesheets in a number of ways. You can use a “master” stylesheet, automatically associating it with all files at the target level or project level (recommended). However, if you have some topics that you want to use a different stylesheet, you also have the option of associating those individual files with that other stylesheet. Once you’ve set up your stylesheet, you can apply its styles to the different pieces of content in your topics and snippets.

As much as possible, you should avoid the opposite of styles, which is local formatting. For example, you can highlight some text and make it green and italic right where that content exists. But if you make that same change in many places, it takes a lot longer and it’s a lot more work to control the look of that content if you later change your mind.

In addition to using stylesheets for topics, you can use separate stylesheets in Flare specifically for tables inserted into topics.

For the differences between regular stylesheets, table stylesheets, and local properties—and when you should use one over the other—see the online Help.
Master Pages

A master page lets you automatically apply certain content—such as breadcrumbs, menus, toolbars, search bars, mini-TOCs, or footer text—to multiple topics in the output. A master page is primarily used in online outputs, but it can be used in Word output as well. For Word output, a master page lets you determine page specifications (such as size or orientation) and to apply certain content (e.g., header text or page numbers) to many topics in a manual. For print-based outputs other than Word, page layouts are used instead of master pages.

EXAMPLE

You want every topic in your Help system to include a footer with contact information about your company. Rather than having to type this content or insert a snippet in every topic, you can create a master page and enter the footer in just that one location. Then you associate the master page with any of the targets in your project. The footer is automatically included at the bottom of every topic when you build and view any of those associated targets.

Page Layouts

A page layout is used for page specifications (e.g., size, margins) and to apply certain content (e.g., headers, footers, page numbers) to many (or all) topics in print-based output. It allows for easy configuration through the use of content frames, bleeds, crop marks, registration marks, margins, padding, alignment features, and more.

Page layouts are similar to master pages, but are more flexible and easier to use. The general rule is that page layouts are recommended for print-based output, and master pages continue to be the best method for automatically adding headers, footers, and breadcrumbs in multiple topics for online output. Another difference between page layouts and master pages is that page layouts can be used for either Adobe PDF or Microsoft Word), whereas master pages can be used only for Microsoft Word when creating print-based output.
Use these buttons to create pages and frames, as well as choose other settings for the layout.

Frames are used to hold content when you generate the print-based output. They appear as rectangles in the editor.

Each layout can have multiple pages (e.g., title, first, left, right, empty). From this area, click the page you want to work on. It then opens to the left.
Skins

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

Depending on the type of output, a skin can help to determine the following:

- Pane position
- Slide-out menu style
- Main menu position
- Top menu depth levels
- User interface text
- Size and position of output
- Which navigation elements to include
- Other settings...
## Importing

There are several ways that you can import files into Flare projects.

Following are a few of the most common types of imported files:

- **Word** You can import Microsoft Word documents, including DOC, DOCX, and RTF files. Flare tightly integrates with Word, using modern XML data flow techniques and leveraging the Microsoft XML Schema for Office documents. This allows for superior content fidelity during import.

- **Excel** You can import Microsoft Excel files into Flare projects. They can be imported into existing projects or when creating a new project. The spreadsheet content will be added to tables in Flare when the import is finished.

- **HTML** You can import HTML files, automatically converting them to XHTML.

- **Confluence** If you have an Atlassian Confluence account, you can import pages (HTM and resource files) into Flare projects. You can import these files into a new Flare project or an existing one. Flare supports both the cloud and local versions of Confluence.

- **FrameMaker** You can import Adobe FrameMaker documents, including BOOK, FM, or MIF files. Because you can import the source FrameMaker BOOK and FM files (rather than just MIF files), Flare has full access to FrameMaker variables, conditionals, autonumbering, and so on. This means that those features are converted to Flare seamlessly.
Exporting

You can export an entire Flare project, or parts of one, to another location. One reason you might want to use this feature is to quickly and easily archive projects, especially if you have an extremely large Flare project and need to archive only parts of it. Another use for this feature is translation. If you only need a portion of a master project to be translated, you don't want to send the translator all of the files, but rather a smaller version of the project containing only the files requiring translation.
Multi-Channel Publishing

Flare supports multi-channel publishing, which means you can draw from a common set of source files to generate and publish output in multiple formats. This helps to ensure your content is accessible to end users, wherever they are, and however they prefer to consume the information.

Although you have many choices about the kinds of output you can generate, following are the most popular and recommended formats.

- **HTML5** This output format supports the HTML5 specification developed by the Web Hypertext Application Technology Working Group (WHATWG—http://whatwg.org) and the World Wide Web Consortium (W3C—http://w3.org). Therefore, the HTML5 format results in better markup and offers additional features not found in the WebHelp outputs in Flare. HTML5 also lets you create traditional Tripane output or the newer Side or Top Navigation outputs, which give your output the appearance of a modern website with integrated responsive skins and content. In addition, HTML5 lets you use advanced server-based features (e.g., searching of non-XHTML files) that are also available with WebHelp Plus.

  For more information, see the *HTML5 Guide*. For links to this PDF manual and others, see the online Help.

- **Adobe PDF** Short for "Portable Document Format," PDF is an open standard format for electronic documentation exchange invented by Adobe. PDF files are used to represent a two-dimensional document in an device- and resolution-independent fixed-layout format.

- **Microsoft Word** The output can be exported to Microsoft Word in DOCX, DOC, PDF, or XPS format.

  For more information, see the *Print-Based Output Guide*. For links to this PDF manual and others, see the online Help.
Search and Micro Content

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

How to Apply a Condition Tag

1. For most elements, you can right-click and select Conditions (or Properties > Conditional Text for files).
2. Select the condition tag set and choose the condition(s) you want to apply and click OK.
3. Click OK to save your work.

More Details

Applying Conditions to Content

Learn how to apply condition tags to different content elements. This lets you single-source the content that you create in multiple target outputs.
Flare/Conditions/Applying-Conditions-Content.htm

Applying Conditions for Print-Based Output

Apply condition tags to mark what shows up in print-based manuals and what doesn’t. Learn about other ways to use condition tags in MadCap Flare.
Flare/Output/Printed-Output/Applying-Condition-Tags-Printed-Output.htm

Applying Conditions to Browse Sequences

Learn how to apply condition tags to different content elements. This lets you single-source the content that you create in multiple target outputs.
Flare/Browse-Sequences/More/Applying-Condition-Tags-Browse-Sequence.htm

With some extra effort and scripting outside of Flare, you can also use micro content for other advanced purposes, such as chatbots.
Project Management

Flare provides several features that can be used to manage your project and enhance team authoring.

Analysis and Reports

Tools are available that let you perform analysis and reporting on both source files and output.

ANALYSIS ON SOURCE FILES

No setup is needed for analysis of source files. You simply run scans on your project.

- **Analysis Ribbon** From the Analysis ribbon in Flare, you can scan files and run reports to discover a wide variety of information. This includes broken links or bookmarks, files with changes, topics not in a table of contents (TOC), and more.
**Reports** From the Reports folder in the Project Organizer, you can generate custom reports based on information contained in your project. In addition, you can design the look and feel of reports, save them for future access, and open them in a browser window (where you can print them).

![Report information is organized, making it easy for you to locate and choose data to include in a report.](image1)

**Text Analysis** From the Tools ribbon in Flare, you can run text analysis on content files for readability, average sentence length, and more.

![You can click this check box to select all options in a category.](image2)

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading ease score: 90*</td>
<td>Reading ease score: 81.4*</td>
</tr>
<tr>
<td>Grade level score: 2.5*</td>
<td>Grade level score: 3.8*</td>
</tr>
</tbody>
</table>

* Denotes statistical significance
ANALYSIS ON OUTPUT FILES

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

The process is quite simple. You create an analytics key in a couple of steps, associating it with a target. Then after building and publishing your output, you use the Analytics page on Central to view user activity on the output as it occurs.

Central Analytics works on Flare projects uploaded to Central, or you can host output on your own servers. If you host the output outside of Central, you still need to use Central with a key to view the analytics data. Also, the server where you host the output must not be locked down, but rather able to communicate with Central.
Global Project Linking

By using Global Project Linking, you can import content and project files contained in another Flare project, thus allowing you to maintain the information in one location but reuse it in any other project. With this feature, you can include or exclude particular types of files (e.g., topics, snippets, stylesheets, glossaries, targets), specific individual files, or files that have certain condition tags applied. Simply use the include/exclude methods that work best for you.

This is different than a simple import process, because in this case, the imported files remain linked to the source project. This lets you make future updates to those files in just one place—in the parent project. When you perform ongoing imports using your previous settings, Flare recognizes changes to the source files. Therefore, the new files can be brought over, replacing the outdated ones.
Language Support

Flare supports authoring and output for left-to-right (LTR) as well as right-to-left (RTL) languages. This includes English, French, German, Japanese, Chinese, Arabic, Persian, Hebrew, multi-byte languages, and more. Flare supports bi-directional languages not only at the topic level, but all the way down to the paragraph and sentence level.

There are multiple ways to produce output in different languages. For example, you might translate content directly in a Flare project, perhaps creating a different topic for each language. Alternatively, you can send the Flare files to a translator, who can use a CAT (computer-assisted translation) tool in order to translate the text strings. MadCap Lingo can be used as a CAT tool or for project management, packaging your Flare files to be sent to a translator, who uses a third-party CAT tool for the translation.
MadCap Central Integration

MadCap Central is a cloud-based platform that lets you plan, track, and manage the processes, content, and teams that are at the heart of your organization. MadCap Central's integration with MadCap Flare lets you store copies of your projects in Central, continue to work on them locally in Flare, and keep both sets of copies in sync. You can use Central to quickly build and publish output (and roll back when necessary) without the need to involve an IT department. You can also send topics and snippets for review on Central, as well as use custom checklists to track your progress in Flare projects.

The MadCap Central window pane in Flare lets you upload (bind) and import projects, as well as push project changes to Central. Additionally, you can see project properties, log in and out of your Central account, and launch the Central portal in your browser.

Not only can you build and publish Flare output on MadCap Central, but you also have the option of publishing output directly to Central from your local Flare project. By “publish,” we mean copying your output files to Central, not making that output “Live,” which would make it visible to the general public. You would still need to use Central to make that output live.

- **NOTE** For more complete information about the benefits of Central, see its online Help: https://help.madcapsoftware.com/central/.

- **NOTE** MadCap Central is sold separately from Flare. Please contact MadCap Software Sales for more information.
Reviews and Contributions

There are a couple of processes that you can use for Flare topic reviews and collaboration.

- **Central Reviews (Recommended)** The Central review process lets you send Flare topics and snippets to be reviewed by subject matter experts (SMEs) or other Flare authors on MadCap Central. After making edits and adding annotations (comments) to the files in a lightweight editor on Central, the reviewers submit the finished files, sending them back to your inbox in Flare. You can then accept or reject their changes and accept the file, replacing the original source file. Because this system uses the cloud, SMEs do not need to download any software to review your files. Also, multiple reviewers can edit the same file at the same time.

- **Review Packages** The review package process lets you send Flare files for review, as well as receive file contributions from, SMEs and other Flare authors. Non-Flare users can download a separate application called MadCap Contributor when collaborating with you and your Flare project. You can also use the same features to send files for review to other Flare authors using Flare only.

**NOTE** If you do not choose to use either of these systems, you can always generate output—such as PDF or Word—and send those files to your reviewers. However, although this can be an easy method, it is also more manual and means that you need to copy and paste the changes into your source Flare files.
**SharePoint and External Resources**

Flare supports integration with Microsoft SharePoint. From Flare you can access, edit, and synchronize SharePoint files. You can also and publish output to a SharePoint server.

One of the ways Flare supports team collaboration is that you can create mappings to external resources. The External Resources window pane lets you select and maintain groups of external files that you want to share among Flare projects. The paths of these files are written to the registry so they will be available for all your Flare projects.

**Source Control**

Because all content and project-level files are stored as separate XML files, projects are compatible with all source control systems. All files in a project are independent of one another, which means that there are no file dependencies that hinder multiple authors from accessing project files. Flare also provides built-in support for Microsoft Team Foundation Server, Perforce, Git, and Apache Subversion. Also, if you integrate MadCap Central with your Flare project, you can use Central as a source control solution, with Git working behind the scenes.

**Templates**

A template is an existing project or file that serves as the basis for a new one, providing preset content, settings, or formatting.
APPENDIX

PDFs

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

TUTORIALS

- Getting Started Tutorial
- Product Foldout Tutorial
- Side Navigation Tutorial
- Top Navigation Tutorial

USER GUIDES

- Accessibility Guide
- Analysis and Reports Guide
- Architecture Guide
- Autonumbers Guide
- Condition Tags Guide
- Context-Sensitive Help Guide
- Eclipse Help Guide
- Getting Started Guide
- Global Project Linking Guide
- HTML Help Guide
- HTML5 Guide
- Images Guide
- Importing Guide
- Indexing Guide
- Key Features Guide
- Language Support Guide
- MadCap Central Integration Guide
- Master Pages Guide
Micro Content Guide
Source Control Guide: Perforce
Movies Guide
Source Control Guide: Subversion
Navigation Links Guide
Source Control Guide: Team Foundation Server
Plug-In API Guide
Styles Guide
Print-Based Output Guide
Tables Guide
Project Creation Guide
Tables of Contents Guide
QR Codes Guide
Targets Guide
Reports Guide
Templates Guide
Reviews & Contributions Guide
Topics Guide
Search Guide
Touring the Workspace Guide
SharePoint Guide
Transition From FrameMaker Guide
Skins Guide
Variables Guide
Snippets Guide
What's New Guide
Source Control Guide: Git

CHEAT SHEETS

Folders and Files Cheat Sheet
Print-Based Output Cheat Sheet
Shortcuts Cheat Sheet
Structure Bars Cheat Sheet
Styles Cheat Sheet