MadCap Software

Key Features Guide

Flare 10

Copyright 2014 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
7777 Fay Avenue
La Jolla, California 92037
858-320-0387
www.madcapsoftware.com

THIS GUIDE WAS CREATED USING MADCAP FLARE.

Key Features

Following are some of the key features that make Flare unique. See the online Help for more information about each feature.

General 4
Interface 5
XML Editor 8
Single-Source Publishing 11
Language Support 14
Page Layouts 15
Importing 16
Exporting 18
Output 19
Project Management and Team Authoring 23
Pulse Integration 25
Other Key Features 26

General

Following are some general key features in Flare.

XML AUTHORING

Even if you do not know anything about Extensible Markup Language (XML), you can use Flare to create XML content (in Flare's main content editor, the XML Editor). If you are experienced with XML, you will be able to take advantage of all of the benefits of this structured language in Flare.

XML EVERYWHERE = OPEN FILE ARCHITECTURE

Topics are not the only Flare documents that use XML. All Flare files are separate XML documents—topics, TOCs, browse sequences, targets, skins, snippets, glossaries, destinations, condition tag sets, variable sets, and more. This means that Flare projects are completely open, transparent, and accessible.

W3C SCHEMA COMPLIANCE

Flare content conforms to industry standard schema requirements from the World Wide Web Consortium (W3C). Therefore, Flare content can be edited in the Flare XML Editor, in external editors, or transferred back and forth between the two at will. Flare’s code adheres to the W3C XHTML Schema specification, making it easy to integrate with other XML or XHTML applications.

Interface

Following are some key features of the Flare user interface.

RIBBONS OR TOOL STRIP VIEW

As an alternative to the classic tool strip view, you can use a more modern ribbon layout that follows the example of newer Microsoft Office products.

The ribbon layout consists of a File drop-down with common tasks such as creating new projects and files, as well as opening and saving documents. This is followed by the Home ribbon, which is the default, and several others related to various features in Flare.

One of the advantages of the ribbon view is that you can create a Quick Access toolbar at the top of the workspace to display options that you use most often. When options are added to the Quick Access toolbar, they are easily accessible, regardless of the ribbon that is open.

You can use the Options dialog to switch between ribbons and the classic tool strip layout. For more information see the online Help.

Keep in mind that the smaller the application window becomes, the more the options in a ribbon shrink. Therefore, you might only see a small icon instead of text, or you might see only a section name displayed with a down arrow to access the options in it. You can hover over small icons to see tooltips that describe them. You can also enlarge the application window or click one of the section drop-downs in the ribbon to locate a hidden feature.

MULTIPLE DOCUMENTS OPEN SIMULTANEOUSLY

Flare is unique in that you can have multiple documents open in the interface simultaneously.

E X A M P L E

If you want to work on four topics, two TOCs, and one glossary all at the same time, you do not have to close one document before working on another. You can have all seven documents open at the same time. You can even "float" and move them so that they are placed side-by-side. If you have access to dual monitors, you can have one document displayed on one monitor and another document displayed on the second monitor.

FLOATING AND DOCKABLE WINDOW PANES

You can "float" window panes and editors, which lets you then click and drag them wherever you want in the interface. A window pane can be attached (or "docked") to the sides, top, or bottom of the program interface.

RESIZABLE DIALOGS AND DROP-DOWN MENUS

All dialogs and even drop-down menus can be resized by clicking and dragging their edges.

WINDOW LAYOUTS

You can save a custom configuration (layout) of your workspace.

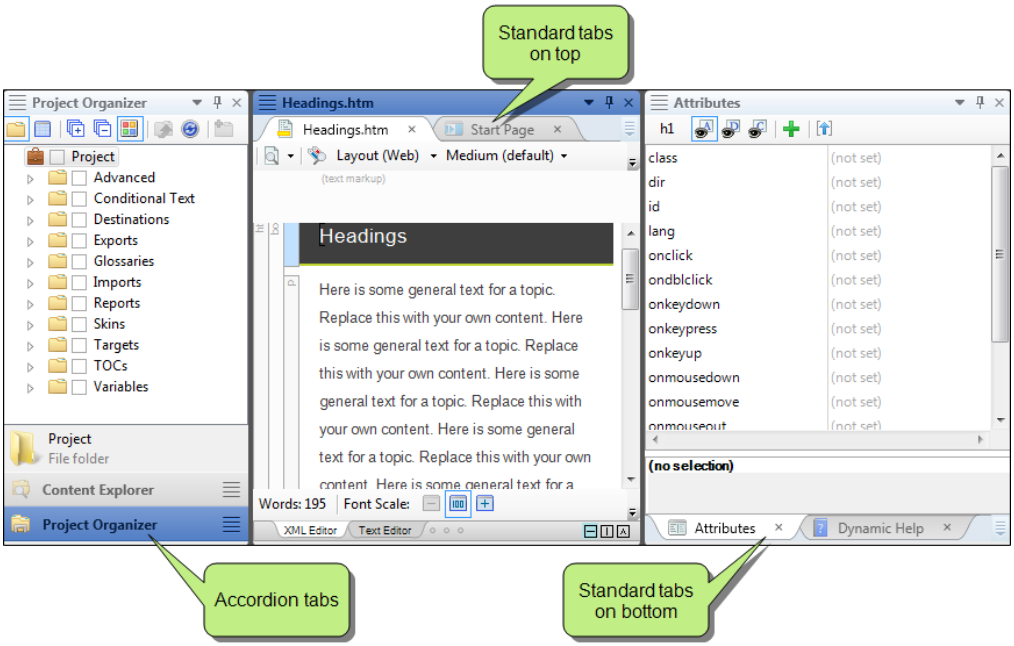
E X A M P L E

You might want to use Flare's default layout most of the time. However, when you are working on creating an index, you want to open the Index window pane. Furthermore, you may want to move that window pane to a certain place in the workspace. After you have this element just where you want it, you can then save that configuration of the workspace as a layout so that you do not have to move things around each time you want to work on indexing.

TAB OPTIONS

You can choose how to view window panes in the workspace.

- **Accordion Tabs** Stacks the document tabs below the window pane or editor.
- **Standard Tabs on Top** Places the document tabs on the top of the window pane or editor.
- **Standard Tabs on Bottom** Places the document tabs on the bottom of the window pane or editor.



ENHANCED UNDO AND REDO

Flare's undo and redo functions are enhanced as follows.

- **Unlimited** You can undo or redo an unlimited number of times, back to the last time you opened the document, even after you have saved it.
- **Multiple Buffers** Each document in the Flare interface has its own Undo/Redo buffer. This means that you can undo actions in one document and then switch to another document and undo actions specific to that file.
- **User-Friendly Interface** If you ware working in the tool strip view, the name of the action is displayed on the Undo or Redo button. In addition, you can use the drop-down arrow next to the button to select multiple steps to undo or redo at once, resizing the drop-down box as necessary.

XML Editor

The XML Editor is the primary editor that is used in Flare. It provides access to the underlying XML structure of documents in a comfortable visual authoring environment. This editor is used to enter, modify, and format the content for topics that users see in the output. Not only is this editor used for topics, but it is also used for working with master pages and snippets. Although this editor lets you produce XML files, you do not need to know anything about XML to use it. Following are some of the key features of this editor.

SPLIT VIEW WITH TEXT EDITOR

The XML Editor and the Text Editor are closely synchronized with each other. They are in essence shown in a single window with options for viewing both at the same time and configuring their appearance.

You can view the XML Editor and Text Editor in a docked state, where only one editor can be seen at a time, but you can easily access the other editor by clicking on a tab. Alternatively, you can undock the window and view both the XML Editor and Text Editor in a split view. Buttons allow you to switch between a horizontal split view and a vertical split view.

PRINT AND WEB LAYOUT MODES

There are two modes that you can use when editing—Web Layout mode and Print Layout mode. These are necessary because Flare lets you create many different kinds of outputs from the same content, and online output is usually meant to look much different than print-based output. So how does the XML Editor know which way to display the content as you are editing it? Should it look like the web output that you intend to generate, or should it look like the PDF manual that you have in mind?

This is where the Web and Print Layout modes come into play. They let you view your work in any number of ways for both online and print-based outputs. Coupled with a style medium and page layout settings, what you see in the XML Editor can vary greatly. If your content doesn't look the way you expect it to in the XML Editor, it likely has something to do with these settings.

STRUCTURE BARS

The XML Editor in Flare contains two kinds of "structure bars," which (as the name suggests) are bars around the topic content that show its structure. Not only do structure bars let you see the tags and spans in a topic, but you can also perform numerous tasks by clicking on the individual bars and making a selection from the context menu. These bars might seem unusual when you first start working in Flare (especially if you are used to working in another tool), but you will soon find that they can be extremely useful and an integral part of creating and designing content.

MARKERS

Flare uses industry-driven markup in content. A marker is a highlighted "flag" that shows the insertion of certain features in the XML Editor—such as variables, bookmarks, index keywords, and concepts ("See Also" links). A major benefit of using markers is that they are deleted when you remove the content containing them. This means that orphan keywords are not left in the project without a point of reference.

E X A M P L E

Let's say you insert an index keyword into a topic. Later, if you decide to delete that topic, the index keyword is also deleted. If markup was not used in the application, but rather a separate database was used to hold all index keywords, you might forget to remove the index keyword after you delete the topic. Then, when a user clicks the index keyword in the output, the application will be unable to find the topic.

ADD NOTES AND COMMENTS IN LISTS

The "Make Paragraph Item" feature lets you add a paragraph tag at the end of a numbered or bulleted list item. This allows you to easily add a comment after the item without interrupting the flow of the list. To use this feature, click on the list item, right-click the **<li>** structure bar on the left side of the editor and from the context menu select **Make Paragraph Item (s)**. You can also press **CTRL+;** (hold down the **CTRL** key and press the semicolon key) if you prefer to use a keyboard shortcut.

MERGE LISTS

There may be times when you are creating lists and find that you have two or more ordered lists (<ol> tags) or unordered lists (<ul> tags) next to each other. If necessary, you can quickly merge the lists together into one list.

DRAG AND DROP TABLE ROWS AND COLUMNS

The XML Editor allows you to drag and drop content as necessary. One unique aspect to this is the ability to click individual rows or columns in a table and drag them to new locations in the table.

COLOR SELECTION TOOL

When you need to apply color to content, you can easily select and customize colors by using Flare's Color Picker dialog. You can even use the "Screen" tool to hover over any area of your computer screen and click to select the precise color beneath it.

HTML TO XHTML CONVERSION

When you attempt to open an HTML file in the XML Editor, a dialog opens, which guides you through the process of converting that file to XHTML so that it can be modified in the XML Editor. You can also select **File>Import HTML Files** to import and convert multiple HTML files to XHTML at once.

TAG-SENSITIVE CURSOR

In Flare's XML Editor, the cursor knows when it is inside or outside of a tag. By default, the cursor is a bracket when it is located between tags (e.g., inline formatting). This bracket cursor faces right ([) or left (]), depending on whether it is inside a particular tag.

E X A M P L E

Let's say you have written a sentence, and at the end of the sentence you have typed the word "Close" in bold font, followed by a period in regular font. Now you want to add the word "button" immediately after the word "Close," but you do not want the word "button" to take on the bold font. You hover the cursor at the end of the word "Close." The bracket faces left (]), which tells you that if you were to begin typing at that location, the next text would also be bold. So you move the cursor slightly toward the right and the bracket cursor faces right ([), which tells you that the next text at this location would take on the regular font of the period. You click the mouse button and type the new text in regular font.

Single-Source Publishing

"Single-sourcing" is a fancy term that means something very simple—to produce multiple results from one source. In Flare, you can make use of single-sourcing in many different ways.

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. When you use this feature to import files, 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 allows you to make future updates to those files in just one place—in the source project file. 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 files.

E X A M P L E

Let's say you are working on three different Flare projects. Within those projects, you might have 35 topics and 50 images that are identical in the three projects. In addition, you might use the same stylesheet in each project. Rather than maintaining three different sets of identical files, you can store one set of those files and import them into the individual projects when needed. Here are a couple of options: (1) One option is that you could consider one of your three Flare projects as the "global parent" for those shared files. (2) Another option is that you could create a new Flare project (perhaps naming it "global"); this project could have no other purpose than to serve as a repository for the shared files across your projects. In other words, you would not necessarily generate any output from this parent project, but simply use it as a place to hold your shared information.

When you want to use any of the shared topic, image, or stylesheet files from the global project, you would import them into the child project. This creates a link between the imported files and those in the global project. Therefore, when you edit those files in the future, you would do so from the global project and then re-import the changes (either manually or automatically) to the other child projects.

MULTIPLE OUTPUTS FROM ONE PROJECT

Flare allows you to generate output in a variety of online and print-based formats, creating as many different targets as you want from the same Flare project.

CONDITION TAGS AT ALL LEVELS

You can apply condition tags at all levels in Flare—character, paragraph, file, and more. In addition, you can create condition expressions using a basic, easy-to-use interface, or you can write your own advanced condition expressions.

SNIPPETS

Snippets are pre-set chunks of content that you can use in your project over and over. They are sort of like miniature topics and can be inserted within topics, or even within other topics. In snippets you can insert text, tables, images, and whatever else can be included in a normal topic.

SNIPPET CONDITIONS

Snippet conditions are condition tags that you can apply to content within snippets. With snippet conditions, you can separate certain snippet content so that it displays in some topics or master pages but not in others. This allows you to use one snippet for many purposes, rather than having to create multiple snippets. Whereas regular conditions are included or excluded at the target level, snippet conditions are included or excluded at the topic or master page level.

VARIABLES

Variables are used for brief, non-formatted pieces of content (such as the name of your company's product or your company's phone number). There are different kinds of variables: (1) basic ones that you create, (2) custom date/time variables that you create, (3) system variables (e.g., date and time; titles; page count; Chapter, Section, and Volume numbers), (4) Heading variables, and (5) Running Head variables. Some of these are especially useful for page headers and footers in print-based output.

STYLESHEETS

You can take advantage of cascading stylesheets (CSS files) to control the look of your output in one place. You can apply stylesheets to individual topics, or you can use a "master" stylesheet, applying it to all files at the target level or project level.

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.

STYLE MEDIUMS

Let's say you want one style setting (e.g., underline font) to be used for online output and another setting (e.g., do not underline font) to be used for printed output. You can use a medium in your stylesheet to create different settings for the same style. When you apply a particular medium to a target, it will be used for that output.

SINGLE-SOURCE IMAGES

Capture is MadCap's screen capture and graphics editing application. From within Flare, you can launch Capture, add or insert a new screen capture, and edit images. Capture contains many unique features that are especially useful for online documentation authors, including the ability to single-source images in Flare projects. For a single image, you can provide one group of settings for online output, and another group of settings for printed output.

E X A M P L E

You might want to use a resolution of 72 DPI for the image in online output, and a resolution of 300 DPI for the image in printed output. Rather than creating two separate images and using condition tags, you can use this feature.

You can also single-source images when resizing them in Flare. This can be done through the use of styles (applying the settings to many images at once) or local formatting (applying the settings to one image). When you generate online output, the image will be displayed in one size, and when you generate print-based output, the image will be displayed in another size.

Note: If you use the "Flare Print tab" feature in MadCap Capture to single-source your images, you may need to take into account the DPI (dots per inch) setting that you set for the image in Capture. If you specify a large DPI that is recommended for print output (e.g., 300 DPI), you may notice that the image shrinks when you generate print output. If necessary, you can resize the width and/or height of the image ahead of time in Capture or later in the Flare project to enlarge the image as necessary for the generated output. The DPI that you specify will be retained, and you will end up with the image dimensions that you need.

Language Support

Following are some of the ways that you can take advantage of language support in Flare.

UNICODE SUPPORT FOR LEFT-TO-RIGHT AND RIGHT-TO-LEFT LANGUAGES

Flare is fully Unicode capable, making it possible to handle the entire world's Unicode language characters. This includes English, French, German, Japanese, Chinese, Arabic, Persian, Hebrew, double-byte languages, and more.

SELECTING A LANGUAGE

You can select a language for a project, target, topic, or even specific content. Selecting a language affects spell checking as well as the language skin used.

MADCAP LINGO INTEGRATION

One of the easiest ways to translate a Flare project is for a translator to open that project within MadCap Lingo, which is tightly integrated with Flare. Because of this integration, there is no need to transfer localized files outside of the actual project, which helps prevent content and formatting corruption. In addition, translators can leverage all previous translations created in other tools by importing Translation Memory eXchange (TMX) files.

After opening your project in Lingo, a translator can immediately see a list of all of the files (e.g., topics, snippets, variables), index markers, and concepts that need to be localized. Then, after translating the content in the Lingo interface, the translator can export the results to a new Flare project in that language. For more information, please refer to the documentation provided with MadCap Lingo.

MULTIPLE INTERFACE LANGUAGES IN ONE PRODUCT

In addition to English, you have the option of viewing the Flare interface in French, German, or Japanese. All of these languages are available from the same version of Flare (you do not need to purchase different versions for different languages).

LOCALIZED INTERFACE FOR DOTNET HELP

The MadCap Help Viewer also supports multiple languages in the interface. Therefore, if you build and distribute DotNet Help, users will be able to select from the available languages to view the interface for your online Help.

LOCALIZED INTERFACE FOR WEBHELP, WEBHELP 2.0 (HTML5), WEBHELP AIR, WEBHELP MOBILE, AND WEBHELP PLUS

If you are generating output for one of the WebHelp output types, you can customize skins in other languages so that your users can see your interface in their own language.

Page Layouts

A page layout is an element that you can create in your project in order to determine 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 of thumb is that page layouts are recommended for *print-based output* (when possible), 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 any of the print-based outputs (Adobe FrameMaker, Adobe PDF, Microsoft Word, Microsoft XPS, XHTML), whereas master pages can be used only for Microsoft Word and FrameMaker when creating print-based output.

EXAMPLE—FRONT MATTER/CHAPTERS/INDEX

Let's say you are creating a manual that consists of front matter (e.g., title page, copyright page, and table of contents), 10 chapters, and an index. Perhaps you want all of the pages in the manual to measure 8 inches in height and 6 inches in width. Furthermore, you might want some pages (e.g., title and copyright pages) to contain no headers or footers, while you want the other parts of the manual to contain header text and page numbers at the bottom. In a situation such as this, you might create one page layout for your title and copyright pages, a second page layout for your TOC, a third page layout to be used by all of the chapters, and a fourth page layout to be used by the index. Each page layout might contain the same page size settings, but different page headers and footers.

Like all other files in Flare, a page layout is an XML file. It has an .flpgl extension. The traditional location to store a page layout in the Content Explorer is in the Resources\PageLayouts folder. However, you can store it anywhere in the Content Explorer that you like.

Importing

Following are some of the ways that you can import files into Flare projects.

IMPORT LEGACY DOCUMENTATION

Flare allows you to import existing documentation in the following formats.

- **DITA File Content** In Flare you can import content from files that have either a .dita or .ditamap extension. You can start a new project by importing DITA file content, or you can import DITA file content into an existing project. When DITA file content is imported, the DITA elements are converted to their equivalents in the Flare XHTML environment. For example, formatting elements are converted to style classes, and <resourceid> elements inside <prolog> elements are converted to context-sensitive Help IDs.
- **FrameMaker Documents** 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.
- **HTML Files** You can import HTML files and convert them to XHTML.
- **HTML Help Files** You can import Microsoft HTML Help files (CHM files).
- **HTML Help Projects** You can import Microsoft HTML Help projects (HHP files).
- **RoboHelp Projects** You can import RoboHelp projects (MPG, XPJ files).
- **Word Documents** You can import Microsoft Word documents, including DOC, DOCX, and RTF files. Flare tightly integrates with Word (including Word 2007), using modern XML data flow techniques and leveraging the Microsoft XML Schema for Office documents. This allows for superior content fidelity during import.
- **XHTML Files** You can import XHTML files.
- **Other Individual Files** You can import a wide variety of other individual files into an existing Flare project.

IMPORT PROJECTS FROM SOURCE CONTROL

If you are using a source control application that is integrated with Flare, you can import an existing Flare project from the source control tool. You might use this method, for example, if you are working on a multi-author project and another member of the team has placed the Flare project in a source control tool such as Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, or Apache Subversion.

EASY SYNC

If you import files from another project, copies of those files are placed in the current project and a link exists between the imported files and the source files in the parent project. Also, if you import Word, FrameMaker, or DITA file content into Flare, you can specify whether a similar link ties those imported files to the source Word, FrameMaker, or DITA files.

These links mean that future changes to the imported files can be made at the source (i.e., in your parent project, or in the source Word, FrameMaker, or DITA files). When you make future changes to the source documents, those files can be re-imported into the project so that they are included in the current project's output. You have the option of re-importing these files manually. However, you can also tell Flare to do this for you automatically when you attempt to generate the output. This is known as "Easy Sync."

The Easy Sync option is labeled **Auto reimport before 'Generate Output'** and is located in the Project Import Editor (for Global Project Linking); Import Microsoft Word Wizard and Word Import Editor (for Word imports); Import FrameMaker Wizard and Frame Import Editor (for FrameMaker imports); and Import DITA Wizard and DITA Import Editor (for DITA imports).

FULL STYLE MAPPING

Flare has full style mapping capabilities. For example, if you import six Word documents, you don't have to worry about six different CSS files being created.

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.

Output

Following are the output options available in Flare.

ONLINE OUTPUT OPTIONS

- **DotNet Help** This is a Help output format developed by MadCap Software. It was designed to include the best attributes of Microsoft HTML Help and WebHelp, while filling the holes left behind by those formats. DotNet Help is designed specifically to support Visual Studio 2005 developers. It includes a freely redistributable viewer (MadCap Help Viewer), as well as components for the Visual Studio 2005 developer. These components can be dropped into your Flare project to facilitate context-sensitive Help, embedded Help, and features such as automated search string communication between the application and the DotNet Help documentation.

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

- **Eclipse Help** This is a Java-based Help system that requires the latest Java Runtime Environment (JRE) and Java version. Use Eclipse Help to create an Eclipse plug-in for the Eclipse Help Viewer.

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

- **HTML5** This is an online format that is the next generation of WebHelp. It is similar to the regular WebHelp format; however, whereas the WebHelp format works with HTML4, this output type supports the new 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. HTML5 also lets you use advanced server-based features (e.g., searching of non-XHTML files, faster search, automatic runtime merging) that are also available with WebHelp Plus.

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

- **Microsoft HTML Help** This is an HTML-based Help format that runs on Windows 32-bit platforms and requires Internet Explorer on the end users' systems. Use Microsoft HTML Help to create Help for Windows desktop applications.

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

- **WebHelp Outputs** There are multiple types of WebHelp output available in Flare.

  - **WebHelp** This is a Web-based Help format that can run on almost any browser or platform. Use WebHelp to create Help for the Internet or an intranet, as well as for desktop applications.
  - **WebHelp AIR** This is a Web-based Help format that is identical to the regular WebHelp output in most ways. However, WebHelp AIR uses direct integration with Adobe AIR, which is designed to bring Web-related content to a desktop environment by taking Web files and incorporating them into a single file to be opened locally, rather than from a server.
  - **WebHelp Mobile** This is an output type that lets you deploy Web-based, XHTML output to mobile devices. WebHelp Mobile maintains an easy and intuitive interface that fits on a very small screen. The Home page in WebHelp Mobile output contains navigation links to access the various panes that you can include: TOC, Index, Glossary, Search, Favorites, Browse Sequences.

» **WebHelp Plus** This is a Web-based Help format that is identical to the regular WebHelp output in most ways. However, WebHelp Plus is designed to work on a Web server running Windows 8, Windows 7, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Windows XP, or Windows Vista. It also uses Microsoft Internet Information Services (IIS) and ASP.NET. To provide faster search, WebHelp Plus uses Microsoft Indexing Service or Windows Search. The benefit of publishing WebHelp Plus output is that you and your users can take advantage of some advanced features, including searching of non-XHTML content, faster server-side search, and automatic runtime merging.

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

PRINT-BASED OUTPUT OPTIONS

- **Adobe FrameMaker** The output is exported to Adobe FrameMaker in one of the following formats.
  - **BOOK** This is FrameMaker's native book (or "document collection") file.
  - **FM** This is FrameMaker's native single document file.
  - **PDF** In addition to sending output directly to PDF, you can generate a PDF file automatically when building FrameMaker output.
- **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 is exported to Microsoft Word in one of the following file formats.
  - **XML** This is the default document format created (if you do not select one of the other formats in this list).
  - **DOC** You can export Word to the standard DOC format.
  - **DOCX** This is Microsoft Word's platform-independent, open XML format.
  - **PDF** In addition to sending output directly to PDF, you can generate a PDF file automatically when building Word output. Because Flare supports Microsoft Vista and Word 2007, you can send Word output to PDF format, even if you do not have the Adobe Distiller installed.
  - **XPS** In addition to sending output directly to XPS (as described above), you can generate an XPS file automatically when building Word output. You can do this by installing a free add-in download from Microsoft.
- **Microsoft XPS** Microsoft's XML Paper Specification (XPS) is a document format with a markup language that is a subset of XAML for Windows Presentation Foundation. XPS is an alternative to Adobe's Portable Document Format (PDF).
- **XHTML** XHTML is a browser-based output type that consolidates project content into an XML file. It can be viewed online or printed.

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

OTHER OUTPUT TYPES

- **EPUB** This is a format for creating reflowable digital books (ebooks). It is developed and maintained by the International Digital Publishing Forum (IDPF). EPUB is sort of a hybrid format. The output is designed to be viewed electronically; in that way, it is like an online format. On the other hand, its structure is like a book or manual; in that way, it is like a print-based format. In fact, you can follow many of the same steps for creating EPUB output that you would follow for producing a PDF. For additional information about this group and the EPUB format, see **http://idpf.org/**.
- **DITA** Darwin Information Typing Architecture (DITA) file content is supported in Flare. DITA is an XML-based markup language with its own schema for authoring, producing, and delivering technical information. It is a standard of the Organization for the Advancement of Structured Information Standards (OASIS), and it consists of a set of design principles for creating "information-typed" modules at a topic level and for using that content in various delivery modes.In Flare you can generate output that produces DITA files. When you build this type of output, a DITA map file is generated, with multiple DITA files in it. The XHTML tags are converted to DITA elements. In other words, although it is considered an "output" from the standpoint of the Flare process, the end result is actually a collection of "source" files, which you can later use in another tool (or import back into Flare) to produce the final output.

COMMAND LINE COMPILE/BATCH PROCESS

You can generate targets from your operating system's command line. Using this method, you do not have to open Flare at all. In addition, this method allows you to build a single target or all targets in your Flare project in one batch. The best way to use the command line feature is to create a batch file with the necessary commands in it. Then you can use a scheduling tool (such as the Task Scheduler utility in Windows) to run the batch file automatically whenever you want.

The batch target in the Flare interface is similar to the "command line" feature. However, the command line feature works outside of the Flare interface, is a bit more manual, and does not support as many processes.

RESPONSIVE OUTPUT

Responsive output is essentially "smart" output. When you generate HTML5 targets using a skin enabled with responsive output, the output display changes depending on the size of your browser or device you are using. So instead of seeing the entire HTML5 output on a tablet or smart phone, you will see a condensed look that is more appropriate for that device. You get the same effect if you shrink your browser to a smaller size.

Project Management and Team Authoring

Flare provides the following features, which can be used to manage your project and enhance team authoring.

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 integrated support for version control applications. Built-in support is available for Microsoft SourceSafe, Microsoft Team Foundation Server, Perforce, and Apache Subversion. In addition, the Microsoft Source Code Control API (SCC API) allows you to configure your project for integration with other version control tools.

SHAREPOINT INTEGRATION

Flare supports integration with Microsoft SharePoint. If your company uses Microsoft SharePoint (software that allows organizations to collaborate, share files, and publish information to the Web), you can connect to a SharePoint server. Doing this makes it easy to access and edit the SharePoint files from any of yourFlare projects. You can even copy SharePoint files to your project with mappings that let you keep them synchronized with the source files. In addition, you can publish Flare output to a SharePoint server. Finally, you can use the SharePoint server to store any kind of template files supported in Flare so that they can be used by any Flare users in your company.

EXTERNAL RESOURCES

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.

External resources can be virtually any local or network files that you have access to (e.g., images, PDF files, Flare project files). From the External Resources window pane, you can easily bring external files into a project (i.e., a copy of the file is added to your Flare project) and keep them synchronized with the source files through mappings.

The external resources feature is ideal for shared files that you expect to change over time (e.g., logo images, PDFs, stylesheets), as opposed to, say, a template file that is simply copied into your project and changed only in that project.

REVIEWS AND CONTRIBUTIONS

Flare provides for close collaboration between authors and others through the use of file reviews and contributions.

- **Reviews** You can email review package files containing topics and/or snippets to other individuals for their feedback and/or changes. Alternatively, you can use SharePoint integration or the external resources feature to store the review package files. The reviewer's changes are always automatically tracked so you can see what edits were made. After inserting annotations (comments) or making changes to a file with Flare or with Contributor (for non-Flare users), reviewers can send the package back to you. You can then view those comments and accept or reject the edits so that they are added to the source file(s).
- **Contributions** Authors, developers, or other individuals in your company can use Contributor to create new documents and files, which can be incorporated into your Flare project.

FILE TAGGING

You can assign tags to topics and any other files in Flare, even folders. You can use file tags for many different purposes, such as assigning authors or milestones to topics. Flare lets you generate reports based on the tags that are assigned. This makes project development easier to track, manage, and schedule.

Pulse Integration

By integrating your output with MadCap Pulse, you can involve an audience in your output, communicating with them and observing how they are using your documentation. Pulse is a documentation-centric social collaboration platform that enables you to connect, collaborate, and share knowledge with fellow authors, employees, and customers. With advanced socially-enabled features, Pulse helps maximize the value of social collaboration and enhance the quality of your documentation.

Note: MadCap Pulse is sold separately from MadCap Flare.

Other Key Features

Following are some additional key features of Flare.

ACCESSIBILITY

Accessibility has to do with making sure that your documentation can be accessed by people with disabilities (i.e., individuals who have visual, hearing, and mobility impairments). Documentation is considered accessible when it is easier for those with disabilities to use and easier to read using assistive technologies such as Microsoft Windows Narrator, Connect Outloud, and JAWS (Job Access With Speech). Flare has several features to help you ensure that your output is accessible.

AUTONUMBERING

This is just what it sounds like—a feature where content is numbered automatically. Of course, if you want to create simple numbered lists in topics, you can always use Flare's quick list drop-down options. But if you want an alternative that is more advanced and powerful, you can use autonumbering.

AUTO SUGGESTION

When you first start typing content, a popup may display below your text as you type. This is the Auto Suggestion feature, which analyzes text as you type and compares it to other content (e.g., phrases, variables, snippets) that already exists in the project. This means you can quickly select existing content from the popup so that you do not have to finish typing it.

CMYK

Flare supports the CMYK color model, in addition to the RGB model. CYMK is an acronym for the four primary colors used in printing (Cyan, Magenta, Yellow, and Key—or black). Whereas CMYK is often the preferred model for print-based output, the RGB (Red, Green, Blue) model is primarily intended for electronic content. CMYK changes can be seen in two locations in the Flare interface: Color Picker dialog and Target Editor (for PDF).

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. You can also convert cross-references to elements such as page numbers for printed output.

You can also take advantage of context-sensitive cross-references, which are especially designed for generating print-based output. When you use a context-sensitive cross-reference, the text automatically changes based on the relationship of the link and the target location if they are on the same page or only one page away.

E X A M P L E

Let's say you have a cross-reference designed to display the text "See Figure 2.1." If the link and the target fall on the same page, the cross-reference is updated to display the text, "See Figure 2.1 above" or "See Figure 2.1 below." If the link and the target are on adjacent pages, the cross-reference is updated to display the text, "See Figure 2.1 on the previous page" or "See Figure 2.1 on the next page." If the document is double-sided with the

link on the left page and the target on the right page, the cross-reference displays the text, "See Figure 2.1 on the facing page."

LINK VIEWER

In Flare there are many ways that you can link one file with another, thus creating a dependency. Link dependencies are created when you perform tasks such as: inserting a text hyperlink, inserting an image, applying a stylesheet to a topic, and more. The Link Viewer window pane lets you see what other files a particular file is linked to and from.

MOVIE SUPPORT

Not only can you explain concepts and tasks to users, but you can also show them through the use of movies. From the interface, you can insert links to MadCap Mimic movies, and you can also embed Flash, Windows Media, QuickTime, and HTML5 movies.

MULTIPLE TOCS

You can have more than one TOC in your project. This is especially useful if multiple authors are working on the same project. One author can work on one part of the TOC while another author works on a different part of the TOC. You can then link the TOCs together. Another use for multiple TOCs is multiple languages (e.g., an English TOC and a French TOC). Yet another use for multiple TOCs is different versions of output (e.g., a TOC for a "light" version, and a different TOC for a "professional" version).

Note: Please be aware that, for online output, a TOC does not by itself determine which topics are generated in the output. However, for print-based output, a TOC works more like an outline and *does* dictate the content included in the output. A TOC for online output is simply a means by which end users can navigate through your information. If you want to use multiple TOCs to generate different outputs that include different topics, you can accomplish this as long as you also make use of Flare's condition tags and targets.

QR CODES

You can insert a QR code into a content file (e.g., topic, snippet) using the XML Editor. A QR code is a type of barcode that can be read by devices such as smart phones. The data encoded in the QR code can be text, a website URL, an email address, contact information, or SMS (Short Message Service, which is used for sending text messages). Basically, QR codes are a way to bridge the gap between a static print document and searchable, more detailed online information at your fingertips.

SLIDESHOWS

You can add a slideshow as an alternative way to display content, using navigation controls to move from slide to slide to view different content. You can place most kinds of content into a slideshow, including snippets, text, tables, and more.

APPENDIX

PDF Guides

The following PDF guides are available for download from the online Help.

Accessibility Guide
Analyzer Guide
Autonumbers Guide
Condition Tags Guide
Context-sensitive Help Guide
DotNet Help Guide
Eclipse Help Guide
Getting Started Guide
Global Project Linking Guide
HTML Help Guide
HTML5 Guide
Images Guide
Importing Guide
Index Guide
Key Features Guide
Language Support Guide
Movies Guide
Navigation Links Guide
Print-based Output Guide
Project Creation Guide
Pulse Guide
QR Codes Guide
Reports Guide
Reviews & Contributions Guide
Search Guide
SharePoint Guide
Shortcuts Guide
Skins Guide

Snippets Guide

Source Control Guide

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Variables Guide

WebHelp Outputs Guide

What's New Guide