



madcap
FLARE[™]

Key Features Guide

Version 6.1

Copyrights

Copyright 2010 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

Contents

- Key Features..... 1**
 - General 1
 - Interface 2
 - XML Editor..... 3
 - Single-Source Publishing..... 5
 - Language Support..... 8
 - Page Layouts..... 9
 - Importing..... 10
 - Output..... 12
 - Project Management..... 15
 - Feedback Service Integration..... 16
 - Other Key Features..... 17
- Index..... 19**

Key Features

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

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:

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

EXAMPLE

If you want to work on four topics, two TOCs, and one style sheet 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.

EXAMPLE

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.

- **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** 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, even 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:

- **Print and Web layout modes** The XML Editor lets you work in two different layout modes—Web or Print. The Web mode is the traditional layout for online use. The Print mode displays topics integrated with page layouts that you can create. This allows you to see the margins, headers, footers, etc. that will be used in the print output.
- **Structure bars** The XML Editor in Flare contains four 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.

EXAMPLE

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, simply click on the list item, select **Format>List>List Actions>Make Paragraph Item(s)**, and then press **Enter**.
- **Merge lists** There may be times when you are creating lists and find that you have two or more ordered lists (tags) or unordered lists (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 **Project>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, style sheets, 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.

EXAMPLE

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 style sheet 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 style sheet 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.
- **Snippets** In Flare, there are multiple ways to reuse content in different places. One method is to create a snippet. Snippets are pre-set chunks of content that you can use in your project

over and over. Snippets are used for longer pieces of content that you can format just as you would any other content in a topic. In snippets you can also insert tables, pictures, 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** Another way to reuse content is to create variables. Variables are pre-set terms that you can use in your project over and over. They are stored in "variable sets," which can hold multiple variables. Flare provides you with an initial variable set, but you can add as many additional variable sets as you like. 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) those you create, (2) system variables (e.g., date and time; Chapter, Section, and Volume numbers), (3) Heading variables, and (4) Running Head variables. Some of these are especially useful for page headers and footers in print-based output.
- **Style sheets** You can take advantage of cascading style sheets (CSS files) to control the look of your output in one place. You can apply style sheets to individual topics, or you can use a "master" style sheet, applying it to all files at the target level or project level. In addition to using style sheets for topics, you can use separate style sheets in Flare specifically for tables inserted into topics.
- **Mediums for topic styles** 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 style sheet to create different settings for the same style. When you apply a particular medium to a target, it will be used for that output.
- **Table print styles** A table style sheet allows you to single-source your formatting by setting the properties in one place and reusing them wherever you insert tables in your project. But what if you want the tables in online output to look one way and the tables in your printed output to look another way? One solution is to insert two different tables (one for online output and another for print) throughout your project and then use condition tags on them. A better solution is to insert a single table at each location, using a special version of the table style for print-based 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 Format" 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 all left-right languages** Flare is fully Unicode capable, making it possible to handle the entire world's Unicode language characters. Flare not only supports Western European languages, but also double-byte Asian languages, Eastern European languages, and more.
- **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 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 Plus, WebHelp AIR, and WebHelp Mobile** If you are generating output for WebHelp, WebHelp Plus, WebHelp AIR, or WebHelp Mobile, 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. Page layouts allow for easy configuration through the use of content frames, a snap-to grid, dragging and dropping, 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 PDF, XHTML, Microsoft XPS, Microsoft Word, Adobe FrameMaker), whereas master pages can be used only for Microsoft Word and FrameMaker when creating print-based output.

EXAMPLE

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 and is stored in the Content Explorer under the Resources\PageLayouts folder.

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:
 - **RoboHelp projects** You can import RoboHelp projects (MPG, XPJ files).
 - **HTML Help projects** You can import Microsoft HTML Help projects (HHP 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.
 - **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, auto-numbering, and so on. This means that those features are converted to Flare seamlessly.
 - **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.
 - **HTML files** You can import HTML files and convert them to XHTML by using the **Project>Import HTML Files** option.
 - **XHTML files** You can import XHTML files.
- **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 or Microsoft Team Foundation Server.
- **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.

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.
- **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 HTML Help to create Help for Windows desktop applications.
- **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 XP, Windows Server 2003, Windows Server 2008, or Windows 7. It also uses Microsoft Internet Information Services (IIS) and ASP.NET. To provide faster search, WebHelp Plus uses Microsoft Indexing Service (for Windows XP and Windows Server 2003) or Windows Search (for Windows Server 2008 and Windows 7). 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.

Print-based Output Options:

- **PDF** Short for "Portable Document Format," PDF is an open file format created by Adobe. PDF files represent two-dimensional documents in a device-independent and resolution-independent fixed-layout document format.
- **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 book** XHTML is a browser-based output type that consolidates project content in an XML file. It can be viewed online or printed.
- **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.
 - **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.
 - **PDF** In addition to sending output directly to PDF (as described above), 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.
- **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 (as described above), you can generate a PDF file automatically when building FrameMaker output.

Code Output Option:

- **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.

Batch targets: You can use batch targets to generate and/or publish one or multiple targets in a batch from the user interface, perhaps scheduled to run at a specific time.

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 Scheduled Tasks 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.

Project Management

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

- **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.
- **Source control** Because all content and project-level files are stored as separate XML files, Flare 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 and Microsoft Team Foundation Server. In addition, the Microsoft Source Code Control API (SCC API) allows you to configure your project for integration with other version control tools. One of the highlights of this integration is an instant message/email system, which can be used to quickly send requests to other authors to check in needed files. This same system allows individuals to check in and check out files on the fly, saving valuable time for multi-author teams.

- **Topic reviews and contributions** Flare provides for close collaboration between authors and others through the use of topic reviews and contributions.
 - **Reviews** You can quickly email topics to other individuals for their feedback and/or changes. After inserting annotations (comments) or making changes to a topic with MadCap X-Edit or X-Edit Review, reviewers can send the topic back to you. You can then accept those comments or changes so that they are added to the source file.
 - **Contributions** Authors, developers, or other individuals in your company can use MadCap X-Edit or X-Edit Contribute to create new documents and files, which can be incorporated into your Flare project.

Feedback Service Integration

There is a way to get live user feedback on your documentation by means of MadCap Feedback, which is a live reporting service that can be purchased in addition to Flare. MadCap Feedback provides cutting-edge Web 2.0 community capabilities and helps you to answer the following types of questions: What are users looking for? What are users finding? What are users unable to find? You can then adjust your project accordingly for a better user experience. Best of all, most of the features in MadCap Feedback are available not only for Web-based Help (i.e., WebHelp and WebHelp Plus), but also for desktop Help (i.e., HTML Help and DotNet Help).

You have two options when setting up MadCap Feedback:

- **MadCap Feedback Server** With MadCap Feedback Server, you install the necessary software on your own hardware to host the Feedback data. By using the MadCap Feedback Server Admin, your system administrator can enter the appropriate server information, create the necessary SQL database, and perform maintenance tasks such as scheduling automated backups of the data. Authors can use the Feedback license keys that are issued as needed for their various targets and projects. The Flare output files can be hosted on the same server hosting the Feedback data, but it is not mandatory; the output files can be published anywhere you like.
- **MadCap Feedback Service** With MadCap Feedback Service, there is no need to purchase your own hardware or host the Feedback statistics. MadCap Software does the hosting for you. This does not mean, however, that you publish the output files to MadCap Software's server. Instead, you publish the output files to your own server or anywhere else that you choose. Even if you have attached your online output to a desktop application, you can still take advantage of the Feedback features. The number of Feedback license keys issued depends on the number that you purchase.

Following are some of the features of MadCap Feedback:

- Retrieve and incorporate user comments into your online output.
- View topic ratings.
- See search keywords entered by users, including terms that did not return results for readers.
- Add synonyms to enhance search results.
- View multi-color, multi-shape Feedback charts for your statistics.
- Use Feedback Explorer to see and track all reader activity with a user-friendly interface.

Note: The MadCap Feedback Server and Service are purchased separately from MadCap Flare.

Other Key Features

Following are some additional key features of Flare:

- **Auto-numbering** 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 auto-numbering.
- **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.

EXAMPLE

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 previous page" or "See Figure 2.1 on 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."

- **Enhanced search** In Flare search results are ranked, not listed alphabetically. In addition, you can create synonyms to enhance future search results for end users.
- **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 a picture, applying a style sheet 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** Mimic is MadCap's movie-making software. From within Flare you can launch Mimic, insert movie links, edit movies, and preview movies. Because Mimic uses advanced compression techniques, your movies will be much smaller than they normally would be if you were to use other movie-making software. This, of course, results in smaller overall output and faster loading for end users. All video tutorials and movies in Flare's online Help were created with MadCap Mimic.

- **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.

Index

A

Adobe FrameMaker	
importing	10
output	13
Adobe PDF	13
ASP.NET	12
Auto-numbers	17

B

Batch	
output	14
BOOK file	10
Building	
batch target	14
targets using command line	14
Bulleted lists	See <i>Lists</i>

C

Capture	See <i>MadCap Capture</i>
Cascading style sheets	6
Chapters	
variables	6
CHM	See <i>HTML Help</i>
Classes	See <i>Styles</i>
Color	
screen	4
Command line	
building targets	14
Comments	
lists	3
Compiling	See <i>Building</i>
Condition tags	
content	5

snippets	6
Cross-references	17
CSS file	6, 11
Cursors	
tag-sensitive	4

D

DITA	
DITA file	10
DITAMAP file	10
importing	10
output	14
DOC file	10, 13
Dockable windows	2
DOCX file	10, 13
DotNet Help	12
language	8

E

Easy Sync	10
Extensible Markup Language	See <i>XML</i>

F

Feedback	
MadCap Feedback	16
Files	
tags	15
Floating	
windows	2
FLPGL file	9
FM file	10
Footers	9
Chapter Number variables	6
Heading variables	6
Running Head variables	6

"FrameMaker" through "Microsoft Windows XP"

Section Number variables	6	Unicode	8
Volume Number variables	6	WebHelp	8
FrameMaker	See <i>Adobe FrameMaker</i>	WebHelp AIR	8
		WebHelp Mobile	8
		WebHelp Plus	8
G		Layouts	See <i>Page layouts</i>
Generating	See <i>Building</i>	Links	
Global Project Linking	5	viewing	17
Graphics	See <i>Pictures</i>	Lists	
		adding notes or comments	3
		merging	3
H		Localization	See <i>Language support</i>
Headers	9		
Chapter Number variables	6	M	
Heading variables	6	MadCap Capture	
Running Head variables	6	single-sourcing images	7
Section Number variables	6	MadCap Feedback	16
Volume Number variables	6	MadCap Help Viewer	12
Heading		MadCap Mimic	17
variables	6	MadCap X-Edit	15
HHP file	10	Mapping	
HTML file	4	styles	11
importing	10	Markers	
HTML Help	12	tags	3
importing	10	Master pages	
I		Heading variables	6
Images	See <i>Pictures</i>	Running Head variables	6
Importing	10	snippet conditions	6
DITA files	10	Mediums	6
files from projects	5	Merging	
FrameMaker documents	10	lists	3
HTML files	10	Microsoft HTML Help	See <i>HTML Help</i>
HTML Help projects	10	Microsoft Indexing	12
RoboHelp projects	10	Microsoft Team Foundation Server	10
source control projects	10	Microsoft Visual SourceSafe	10
topics	10	Microsoft Windows 7	12
Word documents	10	Microsoft Windows Server 2003	12
XHTML files	10	Microsoft Windows Server 2008	12
Indexing services	See <i>Microsoft Indexing</i>	Microsoft Windows Vista	13
Internet Information Services	12	Microsoft Windows XP	12
L			
Language support			
interface	8		

Microsoft Word		Heading variables	6
importing	10	Running Head variables	6
output	13	Section Number variables	6
Microsoft XPS	13	Volume Number variables	6
MIF file	10	Page numbering	9
Mimic	<i>See MadCap Mimic</i>	Paragraphs	
Movies		lists	3
MadCap Mimic	17	Pictures	
Moving		single-sourcing	7
table columns	4	Printed output	
table rows	4	DOC	13
MPJ file	10	DOCX	13
Multi-authoring	10, 15	FrameMaker	13
		Microsoft Word	13
		PDF	13
		table styles	6
		XHTML	13
		XML	13
		XPS	13
		Projects	
		files	5
		Global Project Linking	5
		importing	5
		R	
		Redo	2
		RoboHelp	
		importing	10
		RTF file	10
		Running Head variables	6
		S	
		Schemas	
		DITA	14
		W3C	1
		Screen captures	<i>See Pictures</i>
		Search	17
		Sections	
		variables	6
		Selectors	<i>See Styles</i>
		Single-sourcing	5
		images	7
N			
Navigation links			
cross-references	17		
Numbered lists	<i>See Lists</i>		
O			
Outline TOC	<i>See Tables of contents</i>		
Output			
batch	14		
building targets	14		
DITA	14		
DotNet Help	12		
FrameMaker	13		
HTML Help	12		
Microsoft Word	13		
PDF	13		
types	5		
WebHelp	12		
WebHelp AIR	12		
WebHelp Mobile	12		
WebHelp Plus	12		
XHTML	13		
XPS	13		
P			
Page footers	9		
Page headers	9		
Page layouts			
Chapter Number variables	6		

"Snippets" through "XPS"

Snippets	5	Volume Number variables	6
condition tags	6	Version control	See <i>Source control</i>
Source control	15	Videos	See <i>Movies</i>
importing projects	10	Viewing	
Microsoft Team Foundation Server	10	links	17
Microsoft Visual SourceSafe	10	Visual SourceSafe	See <i>Microsoft Visual SourceSafe</i>
SourceSafe	See <i>Microsoft Visual SourceSafe</i>	Volumes	
Structure bars	3	variables	6
Styles	6	W	
mapping	11	W3C	1
mediums	6	WebHelp	12
style sheets	6	language	8
T		WebHelp AIR	12
Tables		language	8
columns	4	WebHelp Mobile	12
rows	4	language	8
style sheets	6	WebHelp Plus	12
Tables of contents	18	language	8
linking	18	Windows	
multiple	18	layouts	2
Tags		Windows Search	12
files	15	Word	See <i>Microsoft Word</i>
Targets		WYSIWYG Editor	See <i>XML Editor</i>
building output	14	X	
Team Foundation Server	See <i>Microsoft Team Foundation Server</i>	X-Edit	See <i>MadCap X-Edit</i>
TOCs	See <i>Tables of contents</i>	XHTML	
Topics		file	1, 4, 10
snippet conditions	6	output	13
U		XML	1
Undo	2	file	13
Unicode	8	XML Editor	1
V		XPJ file	10
Variables	6	XPS	See <i>Microsoft XPS</i>
Chapter Number variables	6		
Heading variables	6		
Running Head variables	6		
Section Number variables	6		