



madcap  
**FLARE**<sup>™</sup>

## **Quick Guide**

**Version 5.0**

## **Copyrights**

Copyright 2009 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](http://www.madcapsoftware.com)

# Contents

---

<b>CHAPTER 1 Welcome</b> .....	<b>1</b>
How It Works.....	2
Major Benefits.....	4
<b>CHAPTER 2 Starting Projects</b> .....	<b>7</b>
Ways to Start Projects.....	7
Sharing Files Between Projects.....	8
<b>CHAPTER 3 Adding Stuff to Projects</b> .....	<b>9</b>
<b>CHAPTER 4 Making It Look Good</b> .....	<b>13</b>
<b>CHAPTER 5 Developing Outputs</b> .....	<b>15</b>
Important Concepts.....	15
Basic Tasks.....	18
<b>CHAPTER 6 Building Output</b> .....	<b>21</b>
<b>CHAPTER 7 Distributing Output</b> .....	<b>23</b>
<b>Index</b> .....	<b>27</b>



## CHAPTER 1 Welcome

---

Welcome to MadCap Flare—the first native XML content authoring application designed for single-source, multi-channel publishing. Flare is the flagship product in MadCap Software’s integrated suite of next generation authoring software. Like the other products in this suite, Flare is a flexible, open-architecture application that produces XML files with Unicode support for all left-to-right languages.

With Flare, you can deliver content for user Help systems, sales and marketing, training and eLearning, technical support, policies and procedures, and knowledge bases—in print, online, and on the Web. Flare is designed for authors who need a quick and easy way to compose content with all of the re-use and flexibility of the XML format. Fortunately, no knowledge of XML is required in order to use Flare. This allows you to focus on creating content and keep pace with the industry evolution to XML, without having to learn any programming.

This manual is a very shortened version of the Flare *Getting Started Guide*, sort of a CliffsNotes® version. It is ideal for individuals who simply want brief explanations of the most vital parts of the Flare process. For steps and details not found in this manual, please refer to the *Getting Started Guide* or the online Help.

## How It Works

As the following diagram shows, the Flare workflow can be boiled down to just a handful of steps. The real strength of Flare is that you can maintain your content in one place but generate all of it (or portions of it) in whatever output format works best for you.



Following are the basic steps for developing a project:

1. **Start projects** Create a project from scratch, or start a project by importing existing content from a variety of sources. See "Starting Projects" on page 7.
2. **Add "stuff"** Add information and elements to your project, such as topics, text, a table of contents, cross-references, navigation, page layouts, and all of the other "stuff" necessary to help your end users. See "Adding Stuff to Projects" on page 9.
3. **Make your project look good** Through the use of elements such as style sheets and skins, you can develop a look and feel for your output. See "Making It Look Good" on page 13.
4. **Develop outputs** Decide the type(s) of output format(s) that you want to generate (DotNet Help, HTML Help, WebHelp, WebHelp Plus, WebHelp AIR, Adobe PDF, XHTML, Microsoft XPS, Microsoft Word, Adobe FrameMaker, or DITA), and design targets accordingly to meet your needs. See "Developing Outputs" on page 15.
5. **Build output** Generate the file(s) that you will deliver to users. See "Building Output" on page 21.
6. **Distribute output** Make the output accessible to your end users. See "Distributing Output" on page 23.

## Major Benefits

Following are some of the major benefits of using Flare. For more information about each feature, see the online Help.

- **Efficient topic-based authoring and single-sourcing** Flare was designed around the concept of single-sourcing, which essentially means that you can create content once and reuse it in many places and in many ways. In Flare, the focus is on developing pieces of information (topics) that are useful to readers, rather than focusing on creating one enormous document. In other words, you can first focus on the content and later worry about how it should be delivered to users; the two concepts are separate. Topic-based content development allows you to take full advantage of Flare's many powerful single-sourcing features. After you are finished creating topics, you can send them out individually for review (another benefit of small topics), and when the review process is finished, you can decide how to use the topics in the output.

### EXAMPLE

Let's say you write 200 topics in your project. You might send some of these topics to one person for review, and other topics to a different person for review. When that is finished and you are ready, you could output 100% of the topics to one manual, 80% of them into a second manual, and 45% of them into a third manual. Just pick and choose the content you need for each manual.

- **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.
- **Snippets and variables** Sometimes you might have very small pieces of content or text that need to be reused in many topics. Rather than retyping or copying and pasting that content, you can create snippets and variables and maintain the information in just one place.
- **DITA** You can import content from DITA files, work with relationship tables, and generate DITA code output. In addition, you can edit style classes that result from imported DITA elements.
- **Build output at the command line** You can easily generate output from the interface by building individual targets. However, you can also build targets from your operating system's command line. Using this method, you do not have to open Flare at all. You can build a single target or all targets in your Flare project in one batch. In fact, you can use the Microsoft Windows "Scheduled Tasks" feature in conjunction with a batch file that you create in order to automatically generate your output at specific times. For example, you might schedule the

output to be generated every night while you sleep so that you do not need to stare at your computer screen and wait for it to finish.

- **Easily translate content with MadCap Lingo integration** Flare is tightly integrated with MadCap Lingo, which was especially designed to make it as easy as possible to translate a Flare project into other languages.



## CHAPTER 2 Starting Projects

---

The first step in developing a project after you launch Flare is to start a project.

### Ways To Start Projects

You can start a project in one of several ways. For more information, see the online Help.

- **Create a new project from scratch** Use the Start New Project Wizard to create a new project, basing it on a factory template or one that you have created. Select **File>New Project**.
- **Import RoboHelp** Create a new project by importing a RoboHelp project (MPJ or XPJ file). Select **File>Import Project>Import (Non-Flare) Project**.
- **Import HTML Help** Create a new project by importing an HTML Help project (HHP file). Select **File>Import Project>Import (Non-Flare) Project**.
- **Import Microsoft Word** Create a project by importing Microsoft Word documents. In addition, after you create a project (using any method), you can create new topics in that project by importing Word documents. Select **File>Import Project>Import MS Word Documents**.
- **Import Adobe FrameMaker** Create a project by importing Adobe FrameMaker documents. 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. In addition, after you create a project (using any method), you can create new topics in that project by importing FrameMaker documents. Select **File>Import Project>Import FrameMaker Documents**.
- **Import DITA content** Create a project by importing DITA file content. When you do this, the DITA tags are converted to HTML tags (and if you later generate DITA output, they are then converted back to DITA tags). In addition, after you create a project (using any method), you can create new topics in that project by importing DITA file content. Select **File>Import Project>Import DITA Document Set**.
- **Import from source control** Import a Flare project that has been added to a source control application, such as Microsoft Visual SourceSafe or Microsoft Team Foundation Server. Select **File>Import Project>Import From Source Control**.
- **Open a project** Start a project by opening an existing one. Select **File>Open**.

## **Sharing Files Between Projects**

If you create more than one Flare project and need to share some of the same files or content in those projects, it is useful to import linked files from a global project. This is called "Global Project Linking." For more information, see the online Help.

## CHAPTER 3 Adding Stuff To Projects

---

As soon as you start a project, you can do any number of things with it. Technically, you could build the final output immediately. However, if it is a new project, building the output right away would not do your end users much good, since the output does not yet have any real substance. The project needs topics, content, cross-references, navigation, and all of the other "stuff" necessary to help your end users.

Following is a list of features that you can create and insert into your Flare project. Some of these features can be used in online output as well as print-based output, some are for online output only, and some are for print output only. Creating topics is essential for any project, but otherwise you can pick and choose which of the following to include and which to leave out. Also, it is not necessary that you add these items and features in any particular order.

For more information about each feature and step-by-step procedures, refer to the Flare *Getting Started Guide* or the online Help.

- **Topics** A topic is simply a chunk of information about a particular subject. Topics are the most important part of a Flare project. Everything else is contained within topics (e.g., cross-references, text, pictures) or points toward topics (e.g., tables of contents, indexes, browse sequences). The very reason end users open a Help system or manual is to find information, a little direction. They find that help within individual topics.
- **Audio** To enhance your online output, you can insert audio into content, using Flare's tight integration with MadCap Echo. With Echo, you can create audio files that become accessible in a Flash-based, customizable audio player. When users click the audio link, the Audio Player opens with buttons for accessing the sound.
- **Browse sequences** For online output, if you have several topics that you think end users should read in order, you can create browse sequences. When users view the compiled online output, they can use your browse sequences to "browse" through topics in a particular order.
- **Characters and symbols** You can insert special characters or symbols (e.g., a non-breaking space, a copyright symbol, an ellipsis, a monetary symbol) into a topic.
- **Context-sensitive Help** Context-sensitive Help (CSH) is a way to "tie" your existing topics with specific dialogs or windows in a software application. When users open a particular dialog or window in a software application, they can quickly open a topic pertaining to it.
- **Footnotes** A footnote is a comment that is used to explain a specific area of the text. It is used primarily for print-based output. Both the area in the text and the comment contain a number or symbol that ties the two together. A footnote comment is typically placed at the end of a page where the corresponding number or symbol is placed in the text. Otherwise, you

can place a comment later in the manual, such as at the end of a document, chapter, section, or book; in this case, the comment is usually referred to as an endnote.

- **Glossaries** A glossary is a feature that you can add to your output to help users understand the meaning of individual terms. You can include a glossary in both online and print-based output.
- **Index markers** You can create an index by inserting index keywords into individual topics. The index is generated automatically in the output when you build a target. For print-based output, you also need to create a topic with an index proxy in order to make sure the index is included in the output.
- **Master pages** A master page is an element that you can create in your project in order to apply certain content to multiple topics. A master page is useful in online outputs, as well as print-based outputs. However, the benefits are somewhat different for online output than they are for print output. For example, you might use a master page in online output to apply features such as breadcrumbs, mini-TOCs, or footer text to multiple topics, or even all topics in a target. For print-based output, a master page allows you to determine page specifications (such as size or orientation) and to apply certain content (such as header text or page numbers) to many (or all) topics in a manual.
- **Movies** Not only can you explain concepts and tasks to users in your project, but you can also show them through the use of movies. You can insert links to MadCap Mimic movies, or you can insert SWF movie files directly.
- **Navigation links** A navigation link is a feature that points to additional information from a specific area in a topic. The link may open information in the same topic, a different topic, or even a file outside of the project altogether. With print-based output, the link can electronically open the destination if the user is viewing the manual online, depending on the type of output you create (e.g., XPS, PDF, XHTML, Word). In addition, cross-reference links can be customized to refer to specific content and page numbers in the printed manual (e.g., See "My Topic" on page 32). In Flare, you can create links such as the following: cross-references, text hyperlinks, text popups, topic popups, image hyperlinks, togglers, and bookmarks.
- **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.

- **Pictures** You can insert a picture into a topic to help explain something. Flare supports the following types of raster and vector image files: BMP, EMF, EXPS, GIF, HDP, JPG, JPEG, PNG, SWF, TIF, TIFF, WDP, WMF, XAML, XPS.
- **Rulers (horizontal lines)** A ruler is a horizontal line (or bar) that you can insert into a topic. You might use a ruler, for example, as a design element to separate your topic title from the content.
- **Scripts** If you are experienced with using JavaScript, VBScript, or JScript, you can insert such scripts into your topics. If you can create a script that can be used in a website, you can create it in Flare as well.
- **Search** You can add the search feature to any of your online output targets. When users want to find information about a specific subject, they enter key words in the Search field. A search engine looks through every topic in your project to find the term(s) entered by the user. When it finds the terms, it presents the user with a list of topics to open. Search results are ranked, not listed alphabetically.
- **Snippets** 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.
- **Tables** A table in Flare is much like it is in any word processing program, such as Microsoft Word, or in a printed textbook. A table is a group of intersecting columns and rows that you can add to a topic for various purposes, such as comparing different elements.
- **Tables of contents** You can create a TOC by adding books and items with links (to topics, external files, other Help systems, movies, etc.) in a structure that you think would be useful for the individual. End users then browse through a TOC to find information. For print-based output, you also need to create a topic with a TOC proxy in order to make sure the generated TOC is included in the output.
- **Text boxes** You can insert a box into a topic and add content to it. The text box is separate from the regular text in the topic and can be positioned in a variety of places on a page (e.g., aligned left on the page, outside frame, center of column). You might add a text box, for example, to create an example or case study that runs alongside the main text in a topic, in order to enhance the reader's understanding of the main text.
- **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.



## CHAPTER 4 Making It Look Good

---

There are numerous features for "dressing up" your output. The two basic methods for accomplishing most formatting tasks are the use of styles and local formatting.

- **Styles and style sheets** A style is a formatting designation that you can use to quickly change the appearance of your content (such as characters, paragraphs, or tables) in multiple places at once. They allow you to separate your content from its presentation. Styles are stored in cascading style sheet files, and can be found in the Resources\Stylesheets subfolder in the Content Explorer.

### EXAMPLE

You might apply an `<h2>` style to subheadings in your topics in order to separate your content into easy-to-read segments. The `<h2>` style contains all kinds of information (e.g., font type, font size, spacing above and below) that determines how the style appears in the output. Rather than setting the look of one of these headings at each place where it occurs in the project, you can set the look on the style used for the headings; therefore, the look is automatically applied to every heading that uses that style.

- **Local formatting** This is a way to change the look and feel of content directly so that the changes are applied only to that specific content (as opposed to applying the changes throughout your project via the use of styles). Many easy-to-use tools are provided for formatting topics directly in the XML Editor, without having to know XML at all. However, it is recommended that you use styles instead of local formatting whenever possible.

Following are some of the primary ways that you can make your project look good, through the use of styles or local formatting:

- **Auto-numbers** Just as the name implies, auto-numbers are numbers that are associated with content automatically. For example, you might want to create a PDF manual that is divided into volume and chapter numbers. Auto-numbering lets you keep this numbering consistent throughout, without having to apply the numbers manually. There is also a great deal of flexibility in determining what information auto-numbers include and how they look.
- **Fonts** Flare lets you change various font properties to affect the way that your text looks.
- **Headings** When you are creating headings in print-based output, you have a lot of options as to how you can configure them, including the ability to position them (e.g., create side headings). You can simply use the <h1> through <h6> style tags provided in Flare, and you can modify the style settings to meet your needs.
- **Lists** You can create simple and multi-level lists (both numbered and bulleted) for content. These are useful for step-by-step procedures.
- **Object positioning** You can precisely specify the placement of objects, such as images and text boxes. For example, you can float objects to the left or right of a frame in a page. You might do this, for example, if you want text to wrap around an image, or if you want to create a case study (using a text box) and make it display to the left of the flow of normal text in a chapter.
- **Paragraph formatting** There are many ways that you can affect the look of paragraphs that you create. This includes initial caps, drop caps, hyphenation, page column breaks, short line elimination, spacing, widows, orphans, and more. You can also float paragraph content (e.g., position headings to the left of page layout frames).
- **Skins** A skin is a pre-designed look and feel for your online output. In a skin, you control the appearance, size, tabs or buttons, and styles associated with the output window that you generate. Think of it as a suit or a dress in your closet. You can alter your entire look just by changing your clothes. A skin is simply a change of clothes for your output. After you edit the settings for a skin, you associate the skin with a target that you want to build. If you have additional targets that you want to build, you can add more skins to your project, editing their settings, and associating them with the appropriate targets.

For more information, see the online Help or the *Flare Styles Guide*.

## CHAPTER 5 Developing Outputs

---

In Flare, you can determine what kind and how many types of output you want to provide for your end users. This can involve different tasks, depending on what you want to accomplish.

### Important Concepts

Following are some of the more important concepts for developing output.

#### Output Type

There are several types of online output (DotNet Help, HTML Help, WebHelp, WebHelp Plus, WebHelp AIR) and several types of direct print-based output (Adobe PDF, XHTML, Microsoft XPS, Microsoft Word, Adobe FrameMaker) that you can produce in Flare. There is also an output type that lets you generate DITA code from your project (DITA code can be used to create either online or print-based output). Each output type has its own set of advantages.

There is a fine line between what is called "online output" and what is called "print-based" output. The truth is that topics in virtually any of Flare's online output types can be sent to a printer (and therefore considered "print-based"). Similarly, any of the print-based output types can be viewed electronically (and therefore considered "online"). The real distinction between online and print-based outputs has to do with their primary purpose. Online outputs are usually intended to be viewed on a screen, rather than on a printed page. The idea is to show only small pieces of content at a time and allow users to jump around to other topics or elements of the output. On the other hand, print-based output follows a more traditional format that you would find in an actual book or manual—with the pieces of the output following one after the other on pages until the end of the book (e.g., title page, table of contents, preface, chapters, index, appendixes—with page numbers, as well as header or footer content, shown along the way).

#### Target

It is easy to confuse output types with targets, but they are two different concepts. A target is one instance of an output type. When you build your final output, you are essentially building one or more of the targets in your project.

By default, when you create a new project using one of MadCap Flare's factory templates, one new target is added to your project—depending on the format that you selected for your primary target.

However, this target is just a starting point for you. You can rename it to reflect the nature of your project. For example, if you are writing a Help system for a software program called DoohickeyPro,



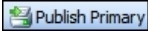
you could rename the default target to "DoohickeyPro." Also, just because only one target was added when you first created the project, this does not mean that you are limited to just that target in your project. You can add as many new targets as you need, using any of the available formats. You can also make as many copies of an existing target as you want. Each target has properties that you adjust to change the way the target behaves, as well as the way it looks and feels.

## Primary Target

You can have as many targets in your project that you want, and any of your targets can be built (generated) whenever you like. However, chances are that you will have one target that you work with more than the others. You can set it as your primary target.

A primary target is treated just like any of your other targets, with one exception. There are certain shortcuts in Flare that let you build, view, or publish your primary target more quickly.

### Primary target shortcuts:

- Build the primary target (click  or press **F6**)
- View the primary target (click  or press **SHIFT+F6**)
- Publish the primary target (click  press **CTRL+F6**)

## Condition Tags

Condition tags are markers that you can apply to different files or areas of your content so that some sections show up in some of your outputs but not in others.

### **E X A M P L E**

Let's say you have two different audiences—beginners and advanced users. The content in your project is the same in most places for both audiences. However, there are sections that apply only to the beginners, and other sections that apply only to the advanced users. You can use one condition tag to mark the sections for the beginners only, and you can use another condition tag to mark the sections for advanced users only. This lets you create one output for the beginners and another output for the advanced users without having to create two separate projects.

## Single-Sourcing

"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.


One way to single-source content is to take advantage of multiple output formats and condition tags. How does it work? Each target in your project is a potential output (using a specific output format, such as DotNet Help or WebHelp). You can create and apply condition tags to content. Then you associate the condition tags to your different targets as necessary, so that some content appears in some targets but not in other targets. This way, you do not need to create a separate project for each output that you want to produce. If most of the content for your outputs is similar, there is no need to rewrite it in another project. Simply specify which sections to include or exclude in which targets through the use of condition tags.

You can also single-source images. If you have MadCap Capture installed and use it to create images for your project, you can provide one group of settings for online outputs and another group of setting for print-based outputs. You can do this from just one image, and you do not need to use condition tags for it.

For more information, including other methods for single-sourcing content, see the online Help.

## Basic Tasks

When developing outputs through the use of targets, there are many possible tasks that you might perform, depending on your situation. You may not need to perform all of the following tasks—just those that fit your needs. You also may not necessarily perform these tasks in the following order. For more information, see the online Help.

- **Determine output type** The first task in developing output for your project is to determine which type of output is most appropriate for your needs. You might even need to produce multiple outputs and require more than one output type. There are several types of online output (DotNet Help, HTML Help, WebHelp, WebHelp Plus, WebHelp AIR) and several types of direct print-based output (Adobe PDF, XHTML, Microsoft XPS, Microsoft Word, Adobe FrameMaker) that you can produce in Flare. There is also an output type that lets you generate DITA code from your project (DITA code can be used to create either online or print-based output). Each output type has its own set of advantages. The output type can be specified on the Basic tab of the Target Editor.
- **Add or make copies of targets** Every target in a project has particular output type assigned to it. You can add multiple targets to a project, and you can make as many copies of existing targets that you want. For example, your project might end up containing three targets that are all based on the WebHelp output type. You can add targets by selecting **Project>Add Target**. After you add a target, it is stored in the Targets folder in the Project Organizer.
- **Rename targets** It is helpful to rename targets that you use to reflect the nature of your project (especially if you are using multiple targets with the same output type). To rename a target, open the **Targets** folder in the Project Organizer, right-click the target you want to rename, and then select **Rename**. Then type a new name for the target and press **Enter**.
- **Set primary target** You can select a primary target when you are in the process of creating a new project. Otherwise, you can open the **Targets** folder in the Project Organizer, right-click the target that you want to specify as the primary, and select **Make Primary**.
- **Create condition tags** If you are creating multiple outputs from the same project and you want the various outputs to contain different content, you can create condition tags. To create a condition tag, you can open the **Conditional Text** folder in the Project Organizer, double-click a condition tag set, and click the **New Item** button  in the Condition Tag Set Editor.
- **Apply condition tags** After you create condition tags, you can apply them to content. There are many ways to apply condition tags, depending on the element you are working on. For example, you can highlight text in a topic and select **Format>Conditions**. In the Condition Tags dialog, you can select the condition tag(s) to be applied to that content.
- **Associate condition tags with targets** After creating and applying condition tags, you need to tell Flare what your target should do with the condition tags that you have created and applied. Should content with a particular condition tag be included in or excluded from that target? To associate condition tags with a target, open the **Targets** folder in the Project Organizer, and double-click the target that you want to open. Then in the Target Editor,

select the **Conditional Text** tab and click the appropriate check boxes next to the condition tags that you want to include in that target or exclude from it.

- **Edit target settings** Using the Target Editor, you can perform tasks such as changing the output type, setting the output file name, selecting a master style sheet, improving the processing performance of the target, and more. To edit target settings, open the **Targets** folder in the Project Organizer, double-click the target that you want to edit, and use the various tabs in the Target Editor to specify settings.

### Output Type Comparison Table—Online Output Types

Following are the various online output types available, with the distinguishing features of each.

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Plus
<b>.NET integration</b>	✓				
<b>Context-sensitive Help (CSH)</b>	✓	✓	✓		✓
<b>Embedded CSH (embed CSH in application)</b>	✓				
<b>Feedback statistics and reporting features</b>	✓	✓	✓	✓	✓
<b>Language support for Help interface</b>	✓		✓	✓	✓
<b>Platform—Desktop-based</b>	✓	✓		✓	
<b>Platform—Web-based</b>			✓		✓
<b>Server—advanced features</b>					✓
<b>Single output file</b>		✓		✓	
<b>Main entry file extension</b>	.mchelp	.chm	.htm	.air	.htm
<b>Output window</b>	MadCap Help Viewer	Microsoft HTML Help Viewer	Browser window	Application window	Browser window

	<b>DotNet Help</b>	<b>HTML Help</b>	<b>WebHelp</b>	<b>WebHelp AIR</b>	<b>WebHelp Plus</b>
<b>Other requirements</b>	MadCap Dot-Net Help Viewer	Internet Explorer		Java Runtime Environment (JRE) Adobe AIR	IIS ASP.NET MS Indexing Service

### Output Type Comparison Table—Print-based Output Types


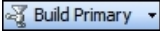
Following are the various print-based output types available, with the distinguishing features of each.

	<b>PDF</b>	<b>XHTML</b>	<b>XPS</b>	<b>MS Word</b>	<b>FrameMaker</b>
<b>Primary benefits</b>	Familiar to many users	Good for large, custom, enterprise level systems	Smaller file size than PDF Better print quality Easy file sharing Enhanced security	Work with output in Word	Work with output in Frame-Maker
<b>Main entry file extension</b>	.pdf	.htm	.xps	.xml (default)	.fm or .book
<b>Output files necessary to distribute</b>	.pdf file only	All files generated	.xps file only	All files generated	All files generated
<b>Program used to view output</b>	MadCap Book Viewer, browser window, or Adobe PDF Reader	MadCap Book Viewer or browser window	MadCap Book Viewer or browser window	Microsoft Word	Adobe Frame-Maker

## CHAPTER 6 Building Output

---

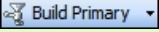
Building the final output is very easy in Flare. It involves generating one or more targets in your project, usually with just the click of a button or two. Following are the different ways you can build your final output:

- **Primary target** Do this if you are only concerned about building the primary target for your project. To build the primary target, do one of the following: (1) In the Project toolbar, click the **Build Primary** button , (2) press **F6** on your keyboard, or (3) select **Build>Build Primary [Name of Target]**.
- **Single target** Do this if you want to build a target that is not designated as your primary target. To build a single target, click the down arrow in the **Build Primary** button , and from the menu, select **Build [Name of Target]**.
- **Command line/batch** Do this if you want to build 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 this 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. For steps, see the online Help.

When you build a target, Flare creates output files and places them in a folder named after the target, which is stored in a subfolder of your project called "Output." For example, let's say your project is stored here: C:\MyProject. In that case, after you generate output, the files would be stored here:

C:\MyProject\Output\MyName\TargetName

Depending on the output type associated with the target, the generated output might consist of many files.

**Note:** If you do not see the **Build Primary** button , open the Project toolbar. To do this, select **View>Toolbars>Project**.



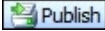
## CHAPTER 7 Distributing Output

---

You can publish the output files manually by simply copying them from the Output folder and pasting them wherever needed. You can easily find the output files by selecting **Build>Open Output Folder** and navigating to the subfolder named after the target you generated.

Alternatively, you can publish output files automatically by using a destination file in Flare.

### How to publish output automatically with a destination file

1. Select **Project>Add Destination**, provide a name, and click **Add**.
2. Open the Project Organizer, expand the **Destinations** folder, and double-click the destination file.
3. Complete the fields in the Destination Editor.
4. From the Project Organizer, open the target that you want to build.
5. In the Target Editor, click  **Publish**, and in the dialog select **Start Publishing**. (You may be informed that the output is out of date, in which case you need to rebuild it before publishing.)

For more information about destinations, see the online Help.

Which output files do you need to be concerned with? It depends on the output type.

### **DotNet Help**

This output type requires that all files and folders in your project file be included for your end users.

**Note:** If you distribute DotNet Help, you also need to provide the MadCap Help Viewer to users. This is a free download from MadCap Software.

### **HTML Help**

There is only one file that needs to be included—the CHM file, which is [Name of your project].chm (e.g., DoohickeyPro.chm).

### **WebHelp and WebHelp Plus**

These output types require that all files and folders in your project file be included for your end users.

### **WebHelp AIR**

There is only one file that needs to be included—the AIR file, which is [Name of your project].air (e.g., DoohickeyPro.air).

**Note:** WebHelp AIR also requires users to install Adobe AIR, which can be downloaded from <http://get.adobe.com/air/>.

### **PDF**

PDF output consists of a collection of files that you can print or distribute to users. This includes:

- A file with a **.pdf** extension. This is the file that contains the consolidated topic content from your project. This is the main file and the only one that you need to distribute.
- A file with an **.mcbook** extension. This is the main file that allows you to view the document with additional navigation in MadCap's built-in Book Viewer. It is necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need this file.
- A file with an **.ftoc** extension. This is the file that is generated if you have incorporated a table of contents into the book. It is necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need this file.
- A Resources folder with **various ancillary files**, such as style sheets and images. They are necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need these files.

## **XPS**

XPS output consists of a collection of XPS files that you can print or distribute to users. This includes:

- A file with an **.xps** extension. This is the file that contains the consolidated topic content from your project. This is the main file and the only one that is essential. This is the file that you would provide to a printer or distribute to end users.
- A file with an **.mcbook** extension. This is the main file that allows you to view the document with additional navigation in MadCap's built-in Book Viewer. It is necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need this file.
- A file with an **.ftoc** extension. This is the file that is generated if you have incorporated a table of contents into the book. It is necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need this file.
- A Resources folder with **various ancillary files**, such as style sheets and images. They are necessary only for viewing the document in the MadCap Book Viewer. Otherwise, you do not need these files.

If you want users to download an XPS document from a server, you need to enable the server to do this by registering the MIME types and file extensions. For steps, see the online Help.

## **XHTML**

XHTML consists of a collection of XHTML files that you can print or distribute to users. This includes:

- A file with an **.htm** extension. This is the XHTML file that contains the consolidated topic content from your project.
- A file with an **.mcbook** extension. This file is used to display the chapters in the MadCap Book Viewer.
- A file with an **.ftoc** extension. This is the file that is generated if you have incorporated a table of contents into the book.
- A Resources folder with **various ancillary files**, such as style sheets and images.

If you want to make XHTML output accessible for others, you need to include all of the files in the output mentioned in this list. Otherwise, when they view the output, certain elements (e.g., images) might be missing from the pages.

## **Word**

Depending on your settings, Microsoft Word consists of one or more XML, DOC, DOX, XPS, or PDF files that you will distribute to users.

## **FrameMaker**

Depending on your settings, Adobe FrameMaker consists of one or more BOOK, FM, or PDF files that you will distribute to your users.

**Note:** When you build Word or FrameMaker output, Flare creates an extra folder called Resources (if your project contains ancillary files, such as pictures). If you move the Word or FrameMaker output files to another location than your Output folder, you must remember to also move the Resources folder. Otherwise, the referenced images will no longer be displayed in the output files. If you convert the Word or FrameMaker files to PDF, the images will then be incorporated into the PDF file, and you can move that single file anywhere you like.

# Index

---

## A

Adobe FrameMaker	20
importing	7
output	15, 18
Adobe PDF	15, 18, 20
AIR file	19
Audio	9
Audio Player	9
Auto-numbers	14

## B

BMP file	11
BOOK file	20
Browse sequences	9
Building	
output	21
primary target	16, 21
single target	21
targets using command line	4, 21
Bulleted lists	See <i>Lists</i>

## C

Chapters	
variables	11
Characters	
inserting	9
CHM	See <i>HTML Help</i>
Classes	See <i>Styles</i>
Command line	
building targets	4
Compiling	See <i>Building</i>

Condition tags	16
content	18
creating	18
targets	18

Content	
editing and formatting	13
Context-sensitive Help	9, 19
CSH	See <i>Context-sensitive Help</i>

## D

Distributing output	23
DITA	4, 15, 18
importing	7
DOCX file	15, 18
DotNet Help	15, 18-19
distributing	24

## E

Echo	See <i>MadCap Echo</i>
Embedded Help	19
EMP file	11
Endnotes	9
EXPS file	11
Extensible Markup Language	See <i>XML</i>

## F

Feedback	19
FM file	20
Fonts	14
Footers	10
Chapter Number variables	11
Heading variables	11
Running Head variables	11

## "Footnotes" through "MPJ file"

---

Section Number variables	11	Importing	
Volume Number variables	11	DITA files	7
Footnotes	9	files from projects	4
Formatting		FrameMaker documents	7
paragraphs	14	RoboHelp projects	7
topic content	13	source control projects	7
FrameMaker	See <i>Adobe FrameMaker</i>	Word documents	7
<b>G</b>		Indexes	10
Generating	See <i>Building</i>	<b>J</b>	
GIF file	11	JavaScript	11
Global Project Linking	4	JPEG file	11
Glossaries	10	JPG file	11
Graphics	See <i>Pictures</i>	JScript	11
<b>H</b>		<b>L</b>	
HDP file	11	Language support	5
Headers	10	interface	19
Chapter Number variables	11	Unicode	1
Heading variables	11	Lines	See <i>Rulers (horizontal lines)</i>
Running Head variables	11	Links	10
Section Number variables	11	Lists	14
Volume Number variables	11	Localization	See <i>Language support</i>
Heading		<b>M</b>	
positioning	14	MadCap Echo	9
side	14	MadCap Help Viewer	19
text boxes	14	Master pages	10
variables	11	Heading variables	11
HHP file	7	Running Head variables	11
Horizontal lines	See <i>Rulers (horizontal lines)</i>	MCHELP file	19
HTM file	19-20	Microsoft HTML Help	See <i>HTML Help</i>
HTML Help	15, 18-19	Microsoft Visual SourceSafe	7
distributing	24	Microsoft Word	20
importing	7	importing	7
Viewer	19	output	15, 18
<b>I</b>		Microsoft XPS	15, 18, 20
Images	See <i>Pictures</i>	Movies	10
		MPJ file	7

Multi-authoring	7	setting	18
		viewing output	16
<b>N</b>		Projects	9
Navigation links	10	creating	7
Numbered lists	See <i>Lists</i>	files	4
		Global Project Linking	4
		importing	4
		opening	7
		starting	7
		Publishing	
		targets	16
<b>O</b>		<b>Q</b>	
Object positioning	14	Quick characters	
Outline TOC	See <i>Tables of contents</i>	inserting	9
Output	15	<b>R</b>	
building targets	4, 21	Raster images	11
distributing	23	RoboHelp	
look and feel	13	importing	7
main entry file	19-20	Rulers (horizontal lines)	11
skins	14	Running Head variables	11
types	15, 18	<b>S</b>	
<b>P</b>		Schemas	
Page footers	10	DITA	4
Page headers	10	Screen captures	See <i>Pictures</i>
Page layouts	10	Scripts	11
Chapter Number variables	11	Search	11
Heading variables	11	Sections	
Running Head variables	11	variables	11
Section Number variables	11	Selectors	See <i>Styles</i>
Volume Number variables	11	Side headings	14
Page numbering	10	Single-sourcing	4, 15
Paragraphs		Skins	
formatting	14	adding	14
positioning	14	Snippets	4, 11
Pictures	11	Source control	
raster	11	importing projects	7
vector	11		
PNG file	11		
Positioning			
headings	14		
objects	14		
paragraphs	14		
Primary target	16		
building	16, 21		
publishing	16		

Microsoft Visual SourceSafe	7	Viewing	
SourceSafe	See <i>Microsoft Visual SourceSafe</i>	primary target output	16
Styles	13	Visual SourceSafe	See <i>Microsoft Visual SourceSafe</i>
style sheets	13	Volumes	
SWF file	11	variables	11
Symbols	See <i>Characters</i>	<b>W</b>	
<b>T</b>		WDP file	11
Tables	11	WebHelp	15, 18-19
Tables of contents	11	distributing	24
Tags	See <i>Styles</i>	WebHelp AIR	19
Targets	15	distributing	24
adding	18	WebHelp Plus	15, 18-19
building output	4, 16, 21	distributing	24
condition tags	18	WMF file	11
copying	18	Word	See <i>Microsoft Word</i>
editing	18	<b>X</b>	
primary	16, 18	XAML file	11
publishing	16	XHTML	15, 18
renaming	18	output	20
viewing output	16	XML	
Text boxes	11	file	20
headings	14	XPJ file	7
TIF file	11	XPS	See <i>Microsoft XPS</i>
TIFF file	11		
TOCs	See <i>Tables of contents</i>		
Topics	4, 9		
<b>V</b>			
Variables	4, 11		
Chapter Number variables	11		
Heading variables	11		
Running Head variables	11		
Section Number variables	11		
Volume Number variables	11		
VBScript	11		
Vector images	11		
Version control	See <i>Source control</i>		
Videos	See <i>Movies</i>		