MadCap Software

WebHelp Guide

Flare 10

Copyright 2014 MadCap Software. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of MadCap Software.

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

THIS GUIDE WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1 WebHelp Outputs 5

About WebHelp Output 6
About WebHelp Plus Output 7
About WebHelp Mobile Output 9
About WebHelp AIR Output 13

CHAPTER 2 Skin Tasks for WebHelp Outputs 15

Excluding Accordion Titles from WebHelp Output 16
Selecting an Image for the About Box 18
Showing Navigation Links in Standalone Topics 20
Specifying Browser Settings 21
Specifying Navigation Pane Settings for HTML5 and WebHelp Outputs 22
Specifying Web Toolbar Settings—WebHelp Outputs 25

CHAPTER 3 CSH Calls for WebHelp 35

What You Need to Do 36
CSH Calls for WebHelp and WebHelp Plus—Developers 37

CHAPTER 4 CSH Calls for WebHelp Mobile 43

What You Need to Do 44
CSH Calls for WebHelp Mobile—Developers 45

CHAPTER 5 Merging Output from Flare WebHelp 47

CHAPTER 6 More About WebHelp Plus 55

Enabling WebHelp Plus Output 56
Installing IIS and ASP.NET 57
Setting Up a WebHelp Plus Target 67
Configuring IIS on Production Server 69
Starting Microsoft Indexing Service 71

Starting Microsoft Windows Search 72
Enabling WebHelp Plus Search 73
Testing WebHelp Plus Search 74
Viewing WebHelp Plus Output 75
Merging Output at Runtime Using WebHelp Plus 79
Including Non-XHTML Files in Search 87
WebHelp Plus Scenario 91
FAQs—WebHelp Plus 96

APPENDIX A PDF Guides 103

APPENDIX B Output Type Comparison Tables 105

General 106
Accessibility 108
Context-Sensitive Help 109
Feedback Statistics and Reporting Features 110
Generated Content 111
Glossaries 113
Horizontal Rules 114
Images 115
Indexes 116
Language Support 117
Master Pages and Page Layouts 119
Merging Output 120
Miscellaneous Features 121
Multimedia 123
Navigation Links 124
Pulse (Socially Enabled Output) 125
Search 126
Skins 133
Tables of Contents and Mini-TOCs 138
Target Settings 140
Variables 143

CHAPTER 1

WebHelp Outputs

Flare lets you choose from many kinds of online output formats. One of these formats is called "WebHelp," and there are multiple flavors of WebHelp. These output formats are similar in many ways, but each has its unique characteristics and advantages.

This chapter discusses the following:

About WebHelp Output 6
About WebHelp Plus Output 7
About WebHelp Mobile Output 9
About WebHelp AIR Output 13

About WebHelp Output

WebHelp is a Web-based Help format that can run on almost any browser or platform. You can use WebHelp to create online documentation for the Internet or an intranet, as well as for desktop applications.

The WebHelp output consists of a collection of files that you will distribute to users. The output will be displayed in the user's Internet browser window. The main entry file has an .htm extension.

Key Features

Following are some of the key features of the WebHelp format:

- **Multiple Browsers** This format is a good choice if your users have different Internet browsers on their systems.
- **Multiple Languages** You can produce an output interface in various languages. This is possible through the use of language skins.
- **Multiple Platforms** This format is a good choice if your users are working on different platforms (operating systems).
- **Web-Based** This format is a good choice if you are writing online documentation for distribution on the Internet or an intranet.
- **Sitemap/Search Engine Optimization (SEO)** You can generate a sitemap when compiling your output. This helps with search engine optimization (SEO), making it easier for search indexing services (i.e., spiders, crawlers, or bots) to find your output. Therefore, the entire output is indexed and search engine results are improved.

Note: The following browsers are supported for WebHelp: Internet Explorer 7.0 or later, Firefox 3.5 or later, Safari 4.0 or later, Google Chrome, and other browsers that support scripts and framesets.

About WebHelp Plus Output

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

The output consists of a collection of files that you will distribute to users by publishing output to a Microsoft IIS web server. The output will be displayed in the user's Internet browser window. The main entry file has an .htm extension.

Key Features

Following are some of the key features of the WebHelp Plus format.

- **WebHelp Features** This format lets you take advantage of features and benefits available with the regular WebHelp output type.
- **Sitemap/Search Engine Optimization (SEO)** You can generate a sitemap when compiling your output. This helps with search engine optimization (SEO), making it easier for search indexing services (i.e., spiders, crawlers, or bots) to find your output. Therefore, the entire output is indexed and search engine results are improved.
- **Multiple Platforms** This format is a good choice if you publish your output to a machine running Windows XP, Windows Server 2003, Windows Server 2008, Windows 7, or Windows Vista.
- **Searching of Non-XHTML Content** When end users perform a search in your online output, you can ensure that non-XHTML files (e.g. PDF, DOC, XLS) are included in that search.When you build WebHelp Plus output, a subfolder named "AutoSearch" is created and placed in the generated output folder. You can place non-XHTML files within the published AutoSearch subfolder (whether the non-XHTML files are linked to content from your Flare project or not). When users perform a search in your output, those non-XHTML files will also be scanned and become accessible to the users. See "Including Non-XHTML Files in Search" on page 87.
- **Faster Searching** Another benefit of generating and publishing WebHelp Plus output to a web server running Microsoft IIS is that users will find the task of performing a search to be much faster than it is otherwise. This is especially useful if you have a very large Help system.
- **Automatic Runtime Merging of Flare Projects** This is an easy way to merge the output from multiple WebHelp Plus Flare targets into one Help system. These targets can be originated from the same Flare project or from different Flare projects. You simply place the output files in the correct location on the server (i.e., within your master project's AutoMerge folder). Flare then automatically merges the output from all of the targets when users access the Help. From the end user's perspective, the results are seamless, appearing as one large Help system. All of the TOCs, browse sequences, indexes, glossaries, and search capabilities for the projects are merged. See "Merging Output at Runtime Using WebHelp Plus" on page 79.

Note: If you want to test WebHelp Plus on your local computer,the advanced search features of WebHelp Plus are not operable.

Note: The following browsers are supported for WebHelp Plus: Internet Explorer 7.0 or later, Firefox 3.5 or later, Safari 4.0 or later, Google Chrome, and other browsers that support scripts and framesets.

Tasks Associated with WebHelp Plus

Following are the primary tasks that you are likely to perform when working with WebHelp Plus.

- **Enable** If you want to take advantage of the advanced server-side features of WebHelp Plus (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable WebHelp Plus output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 56.
- **View the Output** You can view the WebHelp Plus output on the server as well as on your local machine. This is a necessary step if you are testing the output on your local machine. See "Viewing WebHelp Plus Output" on page 75.
- **Merge Multiple Outputs at Runtime** If you want to merge multiple outputs so that they appear as one big Help system to end users, you can do so easily with WebHelp Plus. The outputs from the different projects will be merged at runtime when users open the Help. See "Merging Output at Runtime Using WebHelp Plus" on page 79.
- **Add Non-XHTML Files for Searching** When end users perform a search in your online output, you can ensure that non-XHTML files (e.g. PDF, DOC, XLS) are included in that search. See "Including Non-XHTML Files in Search" on page 87.

Scenario

You can read through a detailed scenario that provides a real-life example of how someone might set up a local computer and server for producing WebHelp Plus output. See "WebHelp Plus Scenario" on page 91.

About WebHelp Mobile Output

WebHelp Mobile 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, Browse Sequences.

The WebHelp Mobile output consists of a collection of files that you distribute to users by placing them on a web server. End users then use their mobile device to open those files, just as they would open any website. The output will be displayed in the user's mobile browser. The main entry file has an .htm extension.

Key Features

Following are some of the key features of the WebHelp Mobile format.

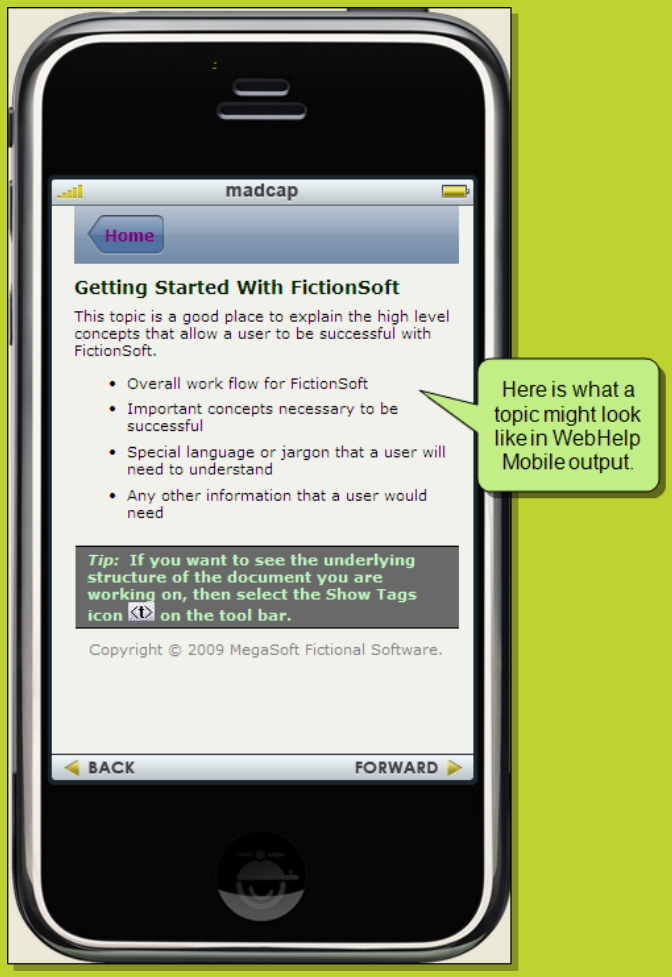
- **WebHelp Features** This format lets you take advantage of features and benefits available with the regular WebHelp output type.
- **Mobile** This format is a good choice if your end users are on the move and need to be able to access your documentation on their mobile devices.
- **Multiple Languages** You can produce an output interface in various languages. This is possible through the use of language skins.
- **Multiple Platforms** This format is a good choice if your users are working on different mobile platforms. Some of the major platforms supported are iPhone, iPad, Microsoft Mobile, LG WebOS (Palm WebOS), Blackberry, and Android.
- **Sitemap/Search Engine Optimization (SEO)** You can generate a sitemap when compiling your output. This helps with search engine optimization (SEO), making it easier for search indexing services (i.e., spiders, crawlers, or bots) to find your output. Therefore, the entire output is indexed and search engine results are improved.

Note: Some older mobile browsers do not support certain features (e.g., DHTML, search), whereas newer mobile browsers do support them. Other features, such as "mouse over," are not supported in either older or newer mobile browsers.

E X A M P L E

Let's say you want to create WebHelp Mobile output so that end users can see your documentation on their cell phones. In the skin associated with the target, you specify that the output should include a TOC, an index, a glossary, and search. After generating the output, the home page might look something like this:

madcap
Home
Index
A
Address
C
Company
Concepts
Contact
Address
Email
Phone
BACK
FORWARD
Here is the Home button.
Index entries let you quickly jump to associated topics.



Here is what a topic might look like in WebHelp Mobile output.

About WebHelp AIR Output

WebHelp AIR 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.

The WebHelp AIR output that you generate consists of a single file with an .air extension, which you distribute to users. When users access this file, they are taken through an installation process, and as a result an executable file with an .exe extension is created on their local drive and saved in their C:\Program Files directory. In other words, the output becomes its own application. The output will be displayed in the application window that is part of the AIR installation.

Key Features

Following are some of the key features of the WebHelp AIR format:

- **WebHelp Features** This format lets you take advantage of features and benefits available with the regular WebHelp output type.
- **Local** This format is a good choice if you want users to store and open the output locally, rather than from a server, such as a website.
- **Single File** This format is a good choice if you want to create output that has just one file.
- **Windows, Mac, and Linux** This format is a good choice if you want users to be able to install the file not only in Windows, but in a Macintosh and Linux environment as well.

Additional Installations

In order for you and your end users to take advantage of this output, both you and they must perform additional installations.

JAVA RUNTIME ENVIRONMENT INSTALLATION (YOU)

As the individual compiling the output, you need to install the Java Runtime Environment (JRE) before generating output.

You can download the JRE from:

http://java.sun.com/javase/downloads/index.jsp

ADOBE AIR INSTALLATION (YOU AND END USERS)

Anyone who wants to view the generated WebHelp AIR output needs to install Adobe AIR first. This means that both you and your end users must run this installation.

You can download Adobe AIR from:

http://get.adobe.com/air/

Specifying WebHelp AIR Options

If you are generating WebHelp AIR output, there are several options that you can set in order to affect the output.

HOW TO SPECIFY WEBHELP AIR OPTIONS

1. Open a target that is using the WebHelp AIR format.

   If necessary, you can change the output type for a target.

2. Select the **Adobe AIR Options** tab.
3. Complete the fields in the dialog.
   - **Program Name** The WebHelp AIR format creates a file with an .air extension. When this file is accessed, an executable file with an .exe extension is created and placed on the user's hard drive. By default, the executable file will be named after the target that you are generating. However, if you want the output to use a different file name, you can enter it in this field.
   - **ID** You can enter an identifier for the application, which uniquely identifies your application along with the publisher ID (which AIR derives from the certificate used to sign the application package). You cannot use spaces in this field.
   - **Copyright** You can enter a copyright in this field.
   - **Digital Certificate** If your company has a digital certificate, you can click the browse button next to this field and select it. By using a digital certificate, the installation will not display "unknown" for the name of the publisher when the user installs the application (i.e., opens the online output).
   - **Certificate Password** Enter the password for the digital certificate.
   - **Program Title** You can enter a title for the online output. When a user installs the application to open the online output, this title will be displayed in one of the installation windows. It is also displayed when you uninstall the application.
   - **Version** You can use this field to enter a version number for your online output.
   - **Description** You can enter a description for your online output. When a user installs the application to open the online output, this description will be displayed in one of the installation windows, under the title (if you specify one).
   - **Install Folder** By default, when a user installs the application to view your online output, the executable file is placed in their c:\Program Files directory, within a subfolder named after the target. If you want the EXE file to be placed in a subfolder with a different name, you can enter it in this field.
   - **Program Menu Folder** You can enter a folder name, which dictates where the program menu option displays in the Start menu. If you do not enter a folder name, the program option is displayed at the root level of the Start menu.
4. Click to save your work.

CHAPTER 2

Skin Tasks for WebHelp Outputs

There are various skin-related tasks unique to WebHelp outputs that you can perform.

This chapter discusses the following:

Excluding Accordion Titles from WebHelp Output 16

Selecting an Image for the About Box 18

Showing Navigation Links in Standalone Topics 20

Specifying Browser Settings 21

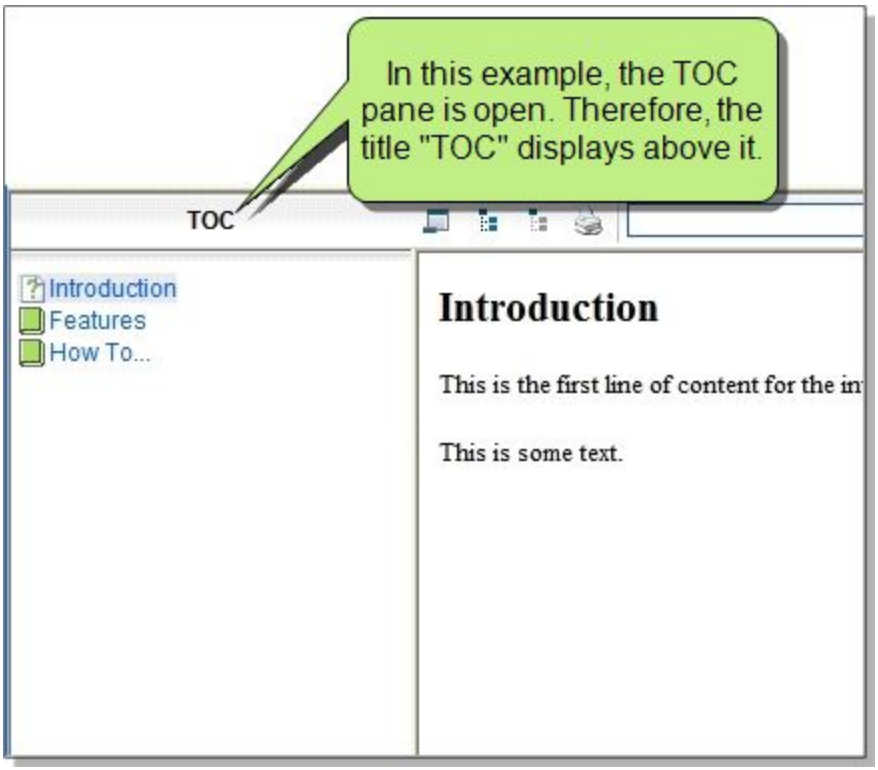
Specifying Navigation Pane Settings for HTML5 and WebHelp Outputs 22

Specifying Web Toolbar Settings—WebHelp Outputs 25

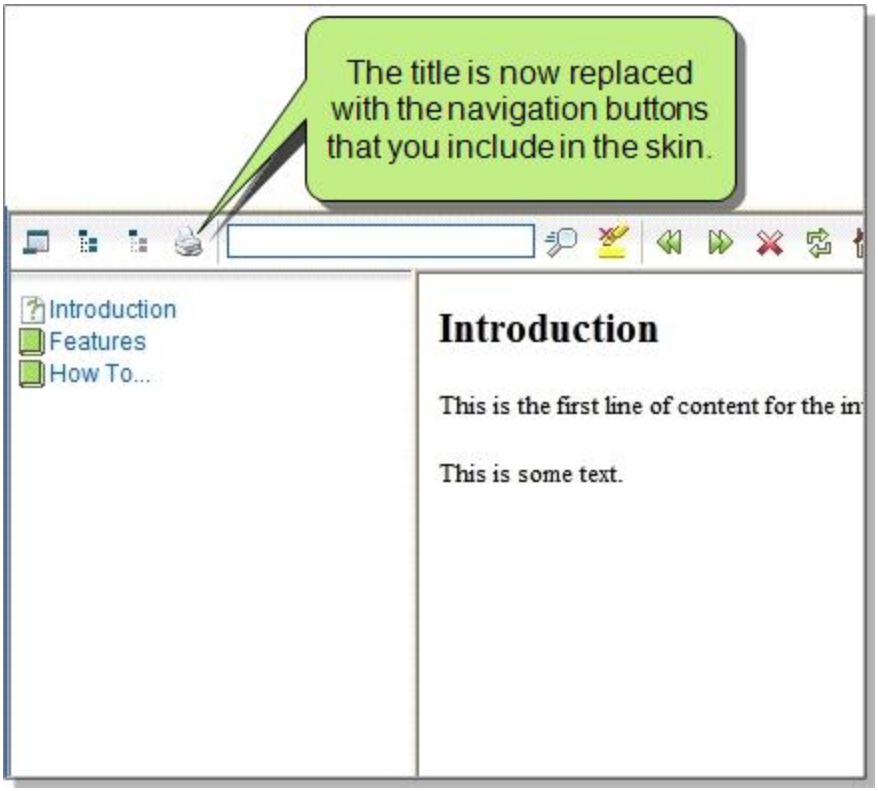
Excluding Accordion Titles from WebHelp Output

This feature is supported in WebHelp, WebHelp AIR, and WebHelp Plus output.

If you generate output using WebHelp, WebHelp AIR, or WebHelp Plus, the title of the active accordion bar displays by default in the output, like this:



You have the option of excluding this accordion title from the output. This shifts the navigation buttons for the output to the left, like this:



HOW TO EXCLUDE ACCORDION TITLES FROM WEBHELP OUTPUT

1. Open a Standard skin.
2. Select the **WebHelp Setup** tab.
3. Select **Exclude Accordion Title**.
4. Click to save your work.

Selecting an Image for the About Box

This feature is supported in Microsoft HTML Help, WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus output.

You can select an image to be used for the "About box" in the output window. You can use this About box for any purpose you like. When users click the icon…

… your image is displayed. When users click the image, it disappears.

HOW TO SELECT AN IMAGE FOR THE ABOUT BOX

1. Open a Standard or Mobile skin.
2. Select the **About** tab.
3. Click **Include About in output**.
4. Click the **Browse** button.
5. In the Open dialog, find and select the image file that you want to use for the About box. A preview of the image is shown at the bottom of the tab.
6. Click to save your work.

Note: If you want to remove the image, click the **Reset** button.

HOW TO SELECT A DIFFERENT IMAGE FOR THE ICON

What if you want to use an image other than the default MadCap icon (the image that users click to open the "About" box image")?

Use the following steps to accomplish this.

1. In the skin select the **Styles** tab.
2. In the Styles section, expand the **Toolbar Item** node.
3. Under it, click **Logo**.
4. Select the **Property Groups** option.
5. In the **Properties** section, click in the cell to the right of **Icon**.

6. Select **Browse for Image**.
7. In the dialog that opens, find and double-click the image that you want to use.
8. Click to save your work.

Showing Navigation Links in Standalone Topics

This feature is supported in HTML5, WebHelp, WebHelp AIR, and WebHelp Plus output.

You can add a navigation link to the top or bottom of topics in HTML5, WebHelp, WebHelp AIR, or WebHelp Plus outputs. This navigation link will not display unless the output topic is opened as a standalone (outside of the main navigation framework of the output). By clicking the link, a user can view the standalone topic in the main navigation framework.

You can also modify the skin styles for these navigation links to change the way they look or the words used in the links. To do this in a Standard skin, open the Skin Editor, select the **Styles** tab, expand the **Control** group, and set the values for the **Navigation Link Bottom** and **Navigation Link Top** properties. To do this in an HTML5 skin, open the Skin Editor, select the **Styles** tab, expand the **Navigation Link** group, and set the values for the **Bottom** and **Top** properties.

HOW TO SHOW NAVIGATION LINKS IN WEBHELP STANDALONE TOPICS

1. Open a Standard or HTML5 skin.
2. If you opened a Standard skin, select the **WebHelp Setup** tab. If you opened an HTML5 skin, select the **Setup** tab.
3. In the **Topic Settings** section, click **Show navigation link at top of topic** or **Show navigation link at bottom of topic**.
4. Click to save your work.

Specifying Browser Settings

This feature is supported in HTML5, WebHelp, and WebHelp Plus output.

You can specify which features will be used in the output window when a browser is involved.

HOW TO SPECIFY BROWSER SETTINGS FOR WEBHELP OUTPUT

1. Open a Standard or HTML5 skin.
2. Select the **WebHelp Setup** tab. If you opened an HTML5 skin, select the **Setup** tab.
3. Select the options that you want to include in the browser output window.
   - **Use Browser Default Settings** Click this check box if you want the output window to use the settings specified in the user's Internet browser. If you do not select this check box, you can select specific browser features below to include in the output window.

   If you do not select "Use Browser Default Settings" above, click the check boxes next to each browser feature that you want to include in the output window.

   - **Toolbar** Displays the browser toolbar at the top of the output window.
   - **Menu** Displays the browser menu at the top of the output window.
   - **LocationBar** Displays the browser location bar (or address bar) at the top of the output window.
   - **StatusBar** Displays the browser status bar at the bottom of the output window.
   - **Resizable** Allows users to resize the output window.
4. Click to save your work.

Important: If you specify browser settings for WebHelp, WebHelp Plus or HTML5, you *must* provide your users with a proper link to open your online Help. Otherwise the browser features mentioned below will not work properly. For details, see "CSH Calls for WebHelp and WebHelp Plus—Developers" on page 37.

Specifying Navigation Pane Settings for HTML5 and WebHelp Outputs

This feature is supported in HTML5, WebHelp, WebHelp AIR, and WebHelp Plus output.

You can specify navigation settings for HTML5, WebHelp, WebHelp AIR, or WebHelp Plus output. The navigation pane is used to hold the TOC, Index, Search, Glossary, Browse Sequences, and Favorites in an accordion-type structure.

HOW TO SPECIFY NAVIGATION PANE SETTINGS FOR WEBHELP OUTPUT

1. Open a Standard or HTML5 skin.
2. Select the **WebHelp Setup** tab. If you opened an HTML5 skin, select the **Setup** tab.
3. Select the appropriate options.

   PANE POSITION

   Select a location (Left, Right, Top, Bottom) in the output window for the navigation pane. A small preview to the right changes as you make your selection.

   PANE SIZE

   Enter the width of the navigation pane for the output window (in pixels). You can type a number or use the up and down arrows.

   VISIBLE ACCORDION ITEMS

   Specify how many navigation items (e.g., TOC, Index, Search) you want to be included at full size in the navigation pane. (This option is not available in HTML5 skins.)

   E X A M P L E

   If you include six tabs in the output window (using the General tab) and set this field to "4," the first four navigation items will be shown with full-sized links. The remaining two items will still be accessible in the navigation pane, but their links will not be full-sized, but rather smaller icons.

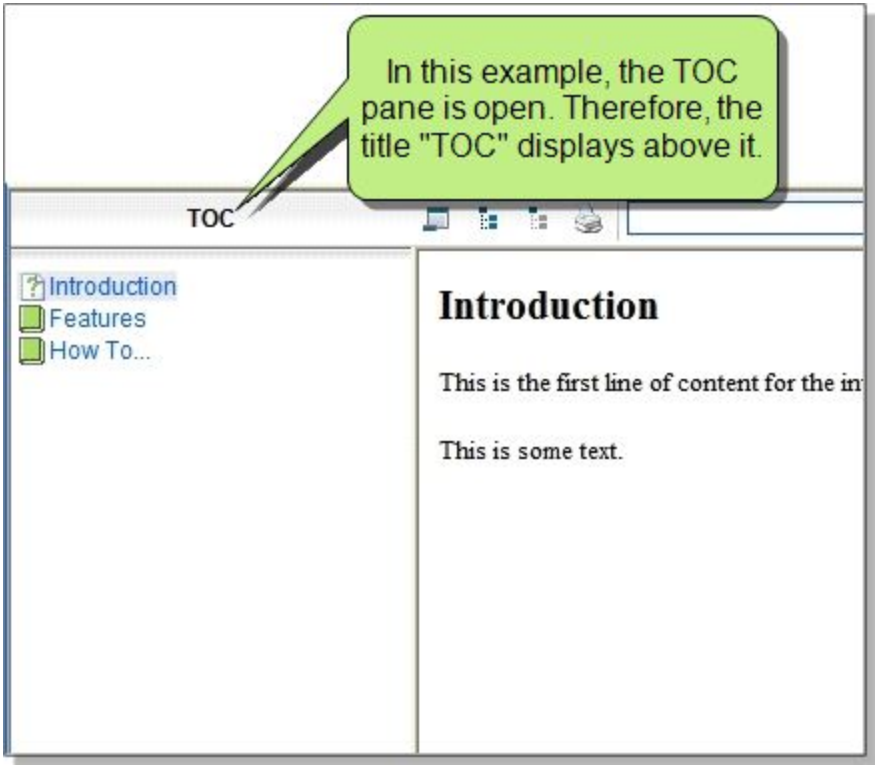
   HIDE NAVIGATION PANE ON STARTUP

   Click this check box if you do not want the navigation pane to be immediately displayed when the Help is accessed in this skin. You might use this option, for example, if you are creating a skin to be used for context-sensitive Help (CSH) topics (as opposed to your main Help system with full navigation). With CSH topics, the navigation pane is not usually necessary because the user wants to see information only for a very specific area.

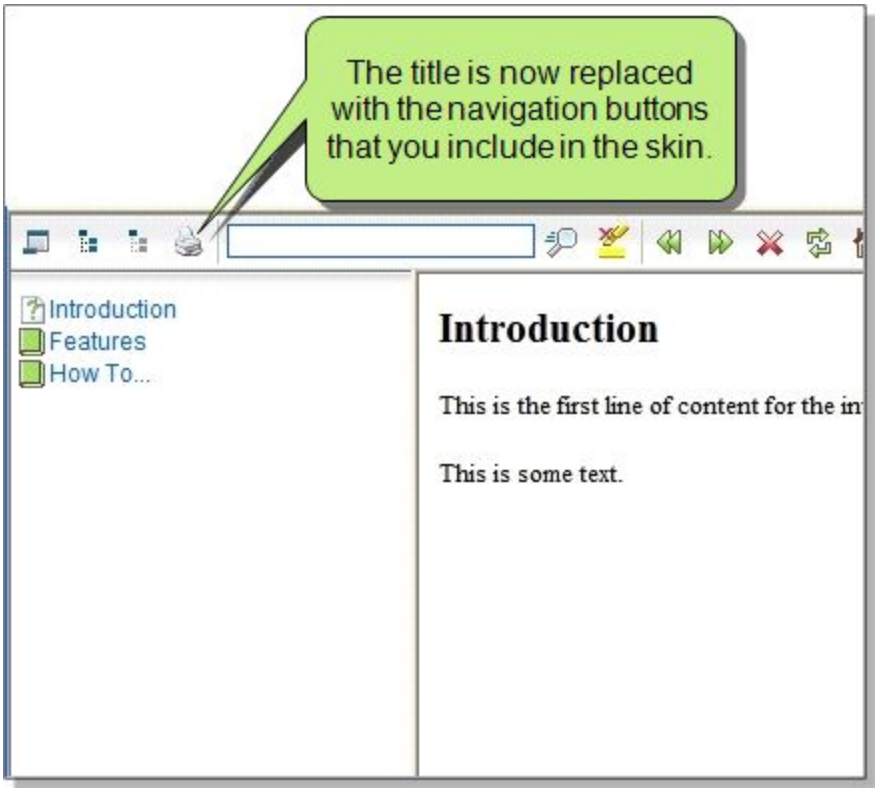
EXCLUDE ACCORDION TITLE

Removes the accordion title from the WebHelp output. (This option is not available in HTML5 skins.)

If you generate output using WebHelp, WebHelp AIR, or WebHelp Plus, the title of the active accordion bar displays by default in the output, like this:



You have the option of excluding this accordion title from the output. This shifts the navigation buttons for the output to the left, like this:



4. Click to save your work.

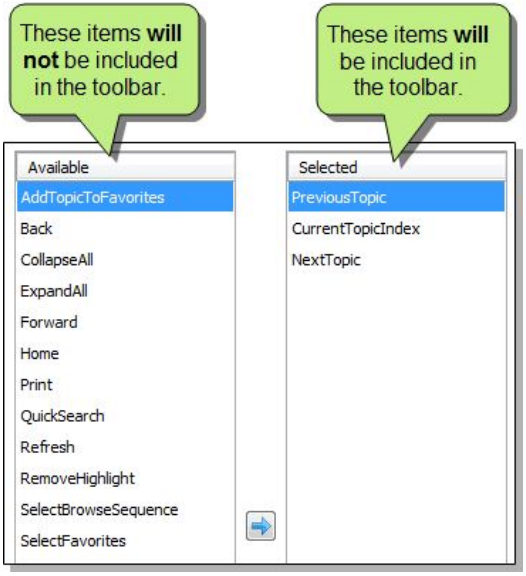
Specifying Web Toolbar Settings—WebHelp Outputs

This feature is supported in WebHelp, WebHelp AIR, and WebHelp Plus output.

You can specify web toolbar settings for some online outputs. This includes determining which buttons are displayed in the toolbar. You can also add custom JavaScript for the toolbar.

HOW TO SPECIFY CUSTOM TOOLBAR SETTINGS FOR WEBHELP OUTPUTS

1. Open a Standard skin.
2. Select the **WebHelp Toolbar** tab.
3. Select options to include specific features (buttons) in the toolbar. Items that will be included in the toolbar are displayed in the Selected section on the right. Items that will not be included in the toolbar are displayed in the Available section on the left. You can use the right arrow button to move items from the Available section to the Selected section.



Below are descriptions of the various items.

ADD TOPIC TO FAVORITES

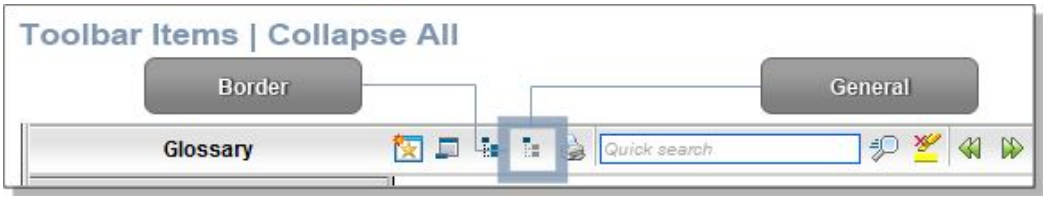
Lets users add the active topic to the Favorites pane so that they can quickly access the topic in the future.

BACK

Lets users open the topic that was viewed previously.

COLLAPSE ALL

Lets users collapse all elements such as togglers, drop-down effects, and expanding text effects in a topic (if they are expanded).



CURRENT TOPIC INDEX

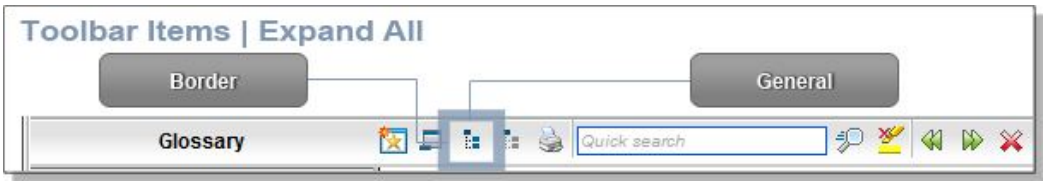
Lets users see the display for the current topic in the sequence. This is typically used in conjunction with the "Next Topic" and "Previous Topic" buttons when the topic in question exists in a browse sequence. Flare finds the current topic in a browse sequence and display its position in relation to the rest of the topics in that sequence. (The root node is used to determine the count; in other words, if there are "child" topics in the sequence structure under the current topic, they will be included in the count.)

EDIT USER PROFILE

Lets users edit their Pulse or Feedback registration profile settings.

EXPAND ALL

Lets users expand all elements such as togglers, drop-down effects, and expanding text effects in a topic (if they are not yet expanded).



FORWARD

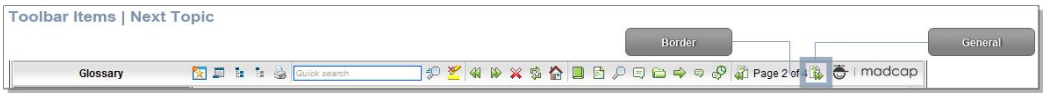
Lets users open the next topic in a previously viewed sequence.

HOME

Lets users open a designated home page (such as your company's website).

NEXT TOPIC

Lets users open the next topic in the sequence. First, Flare attempts to find the current topic in a browse sequence and navigate to the next topic from there. If you do not have a browse sequence, Flare looks at the position of the current topic in your table of contents (TOC) and opens the next topic after it.

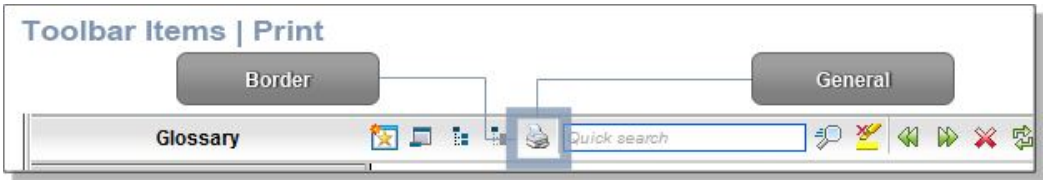


PREVIOUS TOPIC

Lets users open the previous topic in the sequence. First, Flare attempts to find the current topic in a browse sequence and navigate to the previous topic from there. If you do not have a browse sequence, Flare looks at the position of the current topic in your table of contents (TOC) and opens the previous topic before it.

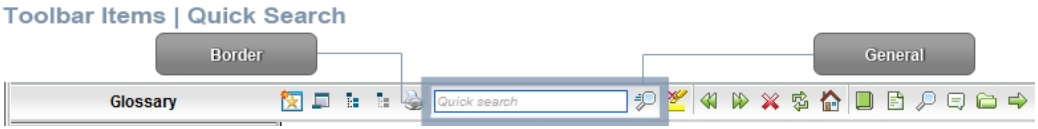
PRINT

Lets users open the Print dialog so that they can send the open topic to the printer.



QUICK SEARCH

Lets users initiate a search in the active topic for the word or phrase that has been typed in the field next to the button.



REFRESH

Lets users update the content of the topic being viewed.

REMOVE HIGHLIGHT

After a user performs a quick search in a topic, the search text found in the topic is highlighted. This button lets users turn the highlights off.

SELECT BROWSE SEQUENCE

Lets users open the Browse Sequence feature.

SELECT FAVORITES

Lets users open the Favorites feature.

SELECT GLOSSARY

Lets users open the Glossary feature.

SELECT INDEX

Lets users open the Index feature.

SELECT SEARCH

Lets users open the Search feature.

SELECT TOC

Lets users open the TOC feature.

STOP

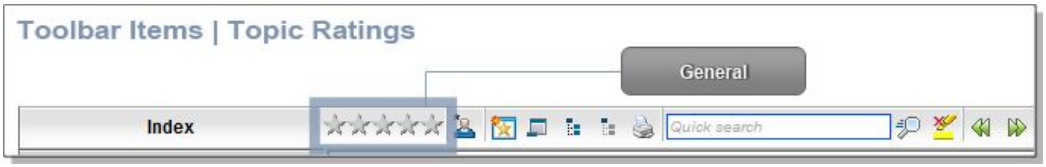
Lets users prevent a page from downloading file information.

TOGGLE NAVIGATION PANE

Lets users toggle between hiding and showing the navigation pane in the output window.

TOPIC RATINGS

Lets users submit ratings for a topic (if you have incorporated your output with MadCap Pulse or Feedback).



4. (Optional) You can use the buttons at the bottom of the tab to perform additional tasks.

	Opens the New Toolbar Button dialog, which lets you add a custom button to the toolbar. After creating the custom button, select it and click to move it to the Selected section. Unless you specify otherwise, in the output the new button will display the text for the name you gave it. However, you can change this by editing the control type for the new button. To do this, you can select the **Styles** tab in the Skin Editor, expand the **Toolbar Item** node, select one of the styles listed above, expand the **Type** property group, and from the **Control Type** field select **Button**. To choose an image for the button, expand the **General** property group on the Styles tab and select image files in the **Icon** fields. If you want to provide a command for the custom button (e.g., so that clicking the button opens a website), you can select the **Styles** tab in the Skin Editor, expand the **Toolbar Item** node, select one of the styles listed above, expand the **Type** property group, and enter a command in the **OnClick** field. For example, the following command opens the MadCap Software website when a user clicks the button: `window.open("http://www.madcapsoftware.com");` Alternatively, you can use the JavaScript area on the WebHelp toolbar tab to enter a command for the button. See the next step.
	Removes the selected custom button from the Available section.
	Moves the selected item up in the order in the Selected section. In the output, buttons are displayed left to right according to the order in which they appear on this tab, with the item at the top being the first button shown on the left in the output.
	Moves the selected item down in the order in the Selected section.

	Removes the selected item(s) from the Selected section.
	Adds a separator to the Selected section. A separator is the divider between the toolbar buttons and the navigation pane.
	Resets the toolbar to the default settings.

5. (Optional) If you want to include custom JavaScript in the toolbar, click **Edit** in the area to the right. This opens the Toolbar JavaScript dialog, which you can use to enter or load custom JavaScript for the toolbar.

   For example, the following command opens the MadCap Software website when a user clicks the button:

```
function MyFunction()
{
window.open("http://www.madcapsoftware.com");
}
```

6. Click to save your work.

Note: You can also specify custom toolbar settings for HTML5 output.

Note: The Section 508 standard states that "Documents shall be organized so they are readable without requiring an associated stylesheet." This means that some users need to view output with styles disabled in the browser. However, if styles are disabled, the frame containing the toolbar in WebHelp output may not be fully visible, and no scroll bar is provided because it was explicitly set not to display. Therefore, in order to be in compliance with Section 508, you can use an option in a Standard skin in Flare to set scrolling values for the toolbar and navigation frames in WebHelp and WebHelp Plus output.

CHAPTER 3

CSH Calls for WebHelp

You can work with your developer to connect a WebHelp or WebHelp Plus system to a software application or to open specific parts of your online documentation from a simple web link. The application or web link(s) can be connected to the basic Help output, to specific topics in the output (context-sensitive Help), or both.

- **Basic Help** The developer can connect the application or web link to your basic WebHelp or WebHelp Plus output, rather than to a specific topic. The Help will open in the browser window, displaying the startup topic that you designate and the navigation elements that you include.
- **Context-sensitive Help (CSH)** The developer can use CSH to connect the application or web link(s) to specific topics in the WebHelp or WebHelp Plus output (as long as you have created CSH in your Flare project and share the header file information with the developer).

This chapter discusses the following:

What You Need to Do 36

CSH Calls for WebHelp and WebHelp Plus—Developers 37

What You Need to Do

1. Work with your developer to determine how you want to connect the online output to the application or web link (s).
2. Create and build your WebHelp or WebHelp Plus target in Flare.
3. Provide the developer with the output files, as well as the CSH header file.
4. Provide the developer with the information in the following topic: "CSH Calls for WebHelp and WebHelp Plus—Developers" on the next page .

CSH Calls for WebHelp and WebHelp Plus—Developers

Information for Developers

Use the following information if you are producing WebHelp or WebHelp Plus and want to incorporate context-sensitive Help (CSH) into the software application.

There are two methods you can use.

- **Method 1—JavaScript** Using this method requires calling a JavaScript function that Flare provides.
- **Method 2—URL** Using this method, you can create a hyperlink to launch the Help system.

Which Method is Best for You?

Each method has its unique benefits. Generally speaking, the JavaScript method lets you have more control, whereas the URL method is a bit more quick and simple.

One reason to choose the JavaScript method is to better control the window size and location. With the URL method, the browser window automatically starts to open at the same size and location as the previous time that browser window was opened. But if you have specified a different size and location for your output window, the window will visibly move and resize accordingly. The JavaScript method prevents this type of behavior by opening the window directly to the size and location you specified. You would set the window size and location in the skin. Then in the JavaScript call you would specify the appropriate skin.

Another benefit to using the JavaScript method is that it is required in order for the Browser Settings option to take effect. This option can be found on the WebHelp Setup tab of the Skin Editor.

HOW TO USE JAVASCRIPT TO OPEN WEBHELP OR WEBHELP PLUS

1. **Author** Add a header file to your project.
2. **Author** Add an alias file to your project.
3. **Author** In the Alias Editor, create and assign an identifier (ID) for the topic that you want to link to from the CSH call.

   For example, if you have a topic called "Welcome.htm," you might create a new ID and also name it "Welcome." Then you might assign a value of 1000 to it. You would make sure that the topic Welcome.htm is assigned to the identifier.

   Finally, an optional step in the Alias Editor is to assign a skin to that ID. However, you can bypass this step and specify a skin later when you create the actual JavaScript, or you can choose not to select a skin at all.

4. **Author** Build your Help system using a WebHelp or WebHelp Plus target and publish the output files to the final destination.
5. **Developer** Add a reference to the JavaScript file (which is created automatically when the author builds the output). This .js file is named after the WebHelp or WebHelp Plus output file and placed at the root level of the output folder. For example, if the output file is named "MyFirstHelp.htm," the JavaScript file is called "MyFirstHelp.js." The reference to the JavaScript file should use the following format: <script type="text/javascript" src="path and name of file.js"></script>.

   Note: Make sure you use forward slashes (/) in the src path to the file, even if the file is referenced locally.

6. **Developer** Create a trigger and add the command to let users open the Help system. Here is a format that you can use to add a button.

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

   You can change the input type and the value if necessary. The most important parts that you will adjust are the elements within parentheses (ID, skin name, search string, first pick search string value).

   - **ID** This is the CSH ID that the author created in Flare (see Step 3 above). This can be either the identifier name or value. The topic and skin associated with the ID will be used. If no skin is associated with the ID in Flare, the skin name that you provide in this command will be used. Alternatively, the ID may contain a topic path. In this case, the specified topic will be loaded with the skin that is specified in this command. The topic path must be relative to the Content folder of the Flare project. You also have the option of entering "null" instead of an ID to use the Help system's default starting topic.
   - **Skin Name** This is the name of the skin to use when opening the Help system. If a skin has been assigned to the ID in Flare (see Step 3 above) *and* you enter a skin name in this command, the skin name in the command will take precedence. You also have the option of entering "null" instead of a skin name if you want to use the Help system's default skin or to use the skin that is associated with the CSH ID in Flare.
   - **Search String** This is an optional element that automatically performs a search for a specific string.
   - **First Pick Search String Value** This element can be used in conjunction with the search string. If you use the first pick option, you can include a true or false value. If the value is true, the first topic found with the specified search string will be opened automatically. If the value is false, the search results will simply be displayed; the first topic will not be opened automatically.

In the following example, the topic and skin associated with "Welcome" will be used. No search string information is included.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp('Welcome', null,
null, null );" />
```

In the following example, the topic associated with "Welcome" will be used. "BlueSkin" will override the skin associated with "Welcome." No search string information is included.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp('Welcome',
'BlueSkin', null, null );" />
```

In the following example, the topic and skin associated with the ID value 1000 will be used. No search string information is included.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp(1000, null, null,
null );" />
```

In the following example, the topic associated with the ID value 1000 will be used. "BlueSkin" will override the skin associated with ID value 1000. No search string information is included.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp(1000, 'BlueSkin',
null, null );" />
```

In the following example, "Company/Employees.htm" will be used with the default skin. No search string information is included.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp('Com-
pany/Employees.htm', null, null, null );" />
```

In the following example, both the default topic and skin will be used. A search will automatically be performed for the words "quarterly report," but the first topic found in the search *will not* be opened automatically.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp(null, null,
'quarterly report', false );" />
```

In the following example, the default topic will be used with "BlueSkin." A search will automatically be performed for the words "quarterly report," and the first topic found in the search *will* be opened automatically.

```
<input type="button" value="Open Help System" onclick="FMCOpenHelp(null, 'BlueSkin',
'quarterly report', true );" />
```

HOW TO USE A URL TO OPEN WEBHELP OR WEBHELP PLUS

1. **Author** (Optional) Add a header file to your project.
2. **Author** (Optional) Add an alias file to your project.
3. **Author** (Optional) In the Alias Editor, create and assign IDs for the topics to which you want to provide links. If you do not want to create a header file, alias file, and IDs for topics, you can instead use the file names for the topics to which you want to link.

   An optional step in the Alias Editor is to assign a skin to that ID. However, you can bypass this step and specify a skin later when you create the actual link, or you can choose not to select a skin at all.
4. **Author** Build your Help system using a WebHelp or WebHelp Plus target and publish the output files to the final destination.
5. **Developer** Create a trigger and add a link to let users open a specific area of the Help system.

   There is a certain amount of flexibility in terms of how you create the link and what you can include in it. Here is the basic structure of the link.

   ```
   [main entry file]_CSH.htm?[search string]|FirstPick#[ID or topic file name.htm]
   |[skin name]
   ```

   - **Main Entry file_CSH.htm** This is the main entry file for your output. The file name is determined by whatever you enter into the **Output File** field in the **General** tab of the Target Editor. If you do not provide a name in this field, the name "Default" will be used. After the file name, it is important that you add an underscore followed by "CSH."

Note: You do not need to include the "_CSH" portion in the file name of the topic in Flare. You only need to add " _CSH" to the web page hyperlink connecting to that topic.

- **Search String** This is an optional element that automatically performs a search for a specific string.
- **First Pick** This element can be used in conjunction with the search string. If you include the first pick option, the first topic found with the specified search string will be opened automatically. If you do not include this element, the search results will simply be displayed; the first topic will not be opened automatically.
- **ID** This is the CSH ID that you created in Flare (see Step 3 above). This can be either the ID name or value. The topic and skin that is associated with the ID will be used, unless you override it in this link by specifying a different skin name. If you do not want to use a map ID, you can use the topic file name (see below).
- **Topic Name** This is the name of the specific topic to which you want to link. If you use this element, you do not need to create an ID as described above. The topic path must be relative to the Content folder of the Flare project. If you do not want to use the topic name, you can use an ID (see above).
- **Skin Name** If you use an ID in the link, the skin associated with that ID (if any) will be used to display the Help. However, you can specify a skin directly in the link, which will override the skin associated with that topic in the alias file.

EXAMPLES

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

<a href="http://my.mycompany.com/Default_CSH.htm#1000">Click here to open</a>

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

<a href="http://my.mycompany.com/Default_CSH.htm?World Cup Standings#1000">Click
here to open</a>

<a href="http://my.mycompany.com/Default_CSH.htm?World Cup Stand-
ings|FirstPick#Soccer">Click here to open</a>
```

In these examples, the following were used.

- Default.htm = main entry file name
- 1000 = CSH ID value

- Soccer = CSH ID name
- Soccer.htm = topic in the project, at the root level of the Content Explorer
- World Cup Standings = search term
- Green = skin name

CHAPTER 4

CSH Calls for WebHelp Mobile

You can work with your developer (or you can function as the developer yourself) to open specific parts of your WebHelp Mobile documentation from web links. In order to accomplish this, you need to have created context-sensitive Help (CSH) in your Flare project and share the header file information with the developer.

This chapter discusses the following:

What You Need to Do 44

CSH Calls for WebHelp Mobile—Developers 45

What You Need to Do

1. Work with your developer to determine how you want to connect the mobile output to the web link(s).
2. Create and build your WebHelp Mobile target in Flare.
3. Provide the developer with the output files, as well as the CSH header file.
4. Provide the developer with the information in the following topic: "CSH Calls for WebHelp Mobile—Developers" on the next page.

CSH Calls for WebHelp Mobile—Developers

Information for Developers

Use the following information if you are producing WebHelp Mobile and want to incorporate context-sensitive Help (CSH). This lets users open specific parts of the output from web links.

HOW TO CREATE CSH CALLS FOR WEBHELP MOBILE

1. **Author** (Optional) Add a header file to your project.
2. **Author** (Optional) Add an alias file to your project.
3. **Author** (Optional) In the Alias Editor, create and assign IDs for the topics that you want to provide links to. If you do not want to create a header file, alias file, and IDs for topics, you can instead use the file names for the topics to which you want to link.
4. **Author** Build your Help system using a WebHelp Mobile target and publish the output files to the final destination.
5. **Developer** Create a trigger and add a link to let users open a specific area of the Help system.

   There is a certain amount of flexibility in terms of how you create the link and what you can include in it. Here is the basic structure of the link.

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

   After the question mark, you can specify any combination of the parameters (cshid, searchQuery, firstPick), separted by ampersands (&). The order of the parameters does not matter.

   - **Main entry file** Provide the path to the main entry file for your output. The file name is determined by whatever you enter into the **Output File** field in the **General** tab of the Target Editor. If you do not provide a name in this field, the name "Default" will be used.
   - **cshid** This is the CSH ID that you created in Flare (see Step 3 above). This can be either the identifier name or value. Alternatively, you can enter the path and name of the specific topic to which you want to link. If you use this element, you do not need to create an ID as described above. The topic path must be relative to the Content folder of the Flare project.
   - **searchQuery** This is an optional element that automatically performs a search for a specific string.
   - **firstPick** This element can be used in conjunction with the search string. If you include the first pick option, the first topic found with the specified search string *will* be opened automatically. If you do not include this element, the search results will simply be displayed; the first topic *will not* be opened automatically.

EXAMPLES

```
<a href="http://mycompany.com/Default.htm?cshid=1000&searchQuery=World Cup Stand-
ings&firstPick=true">Click here to open</a>

<a href="http://mycompany.com/Default.htm?cshid=Soccer&searchQuery=World Cup Stand-
ings&firstPick=true">Click here to open</a>

<a href="http://mycompany.com/Default.htm?cshid=Soccer.htm&searchQuery=World Cup Stand-
ings&firstPick=true">Click here to open</a>
```

In these examples, the following were used.

- Default.htm = main entry file name
- 1000 = CSH ID value
- Soccer = CSH ID name
- Soccer.htm = topic in the project, at the root level of the Content Explorer
- World Cup Standings = search term

CHAPTER 5

Merging Output from Flare WebHelp

This feature is supported in all WebHelp outputs (HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus).

Use this merging method if you are developing one of the WebHelp outputs (WebHelp, HTML5, WebHelp AIR, WebHelp Mobile, WebHelp Plus) and you want to merge the output files from your parent project with the WebHelp output files from an external Flare project. This method is useful, for example, if another author is working on the external WebHelp project to which you are linking and you only have access to the output files (not the project files).

Before your master project can be merged with an external Help system, the output folders and files for the external (or "child") project need to be copied to a folder called "Subsystems" in the same location where the master project is being published. The child's target folder containing these output files also needs to be included at this location.

E X A M P L E

For example, if the master project's output files are being published to C:\MyHelpSystem\Master\Master.mcwebhelp, the output folders and files for the child project should be copied to the folder C:\MyHelpSystem\Master\webSubsystems (e.g., C:\MyHelpSystem\Master\Subsystems\AnotherHelpSystem\AnotherHelpSystem.mchelp). In this example, "AnotherHelpSystem" is the target folder that you generated, which holds the output folders and files, and "AnotherHelpSystem.mcwebhelp" is the main output file for the child project that was created. Not only does the main .mcwebhelp output file need to be included in this folder, but also the ancillary output folders and files (e.g., Content, Data, Skin).

You can merge WebHelp outputs using an automatic or manual method.

HOW TO MERGE OUTPUT FROM WEBHELP PROJECTS (AUTOMATIC METHOD)

1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.

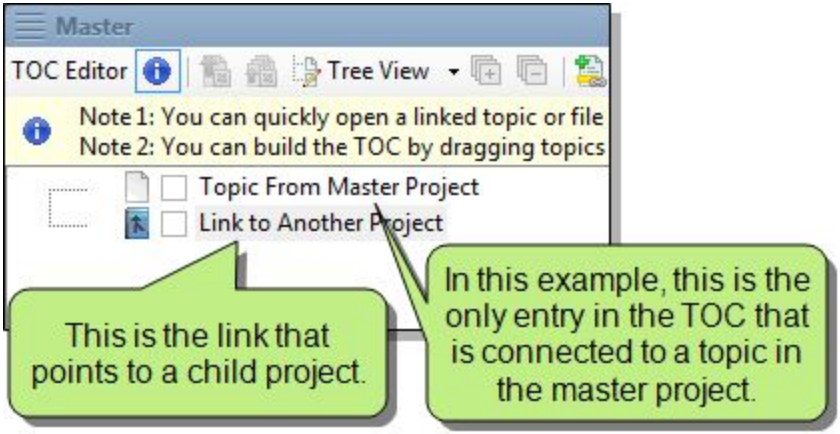
   Note: You can also merge projects by linking them from a browse sequence, as well as from a TOC. If you want to do this, simply follow these same steps after opening your browse sequence.

3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child WebHelp project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Click **Browse**.
7. In the **Files of type** field, click the down arrow and select **MadCap Web Help Files (*.mcwebhelp)**. Then find and select the MCWEBHELP output file to which you want to link.
8. Click **OK**.

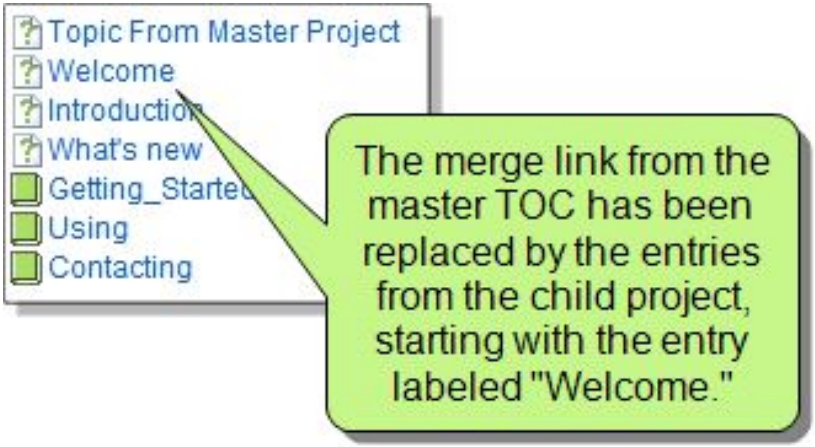
   The full path to the file is displayed in the field.

9. (Optional) If you want the merged TOC to replace the entry that you selected, do the following:
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

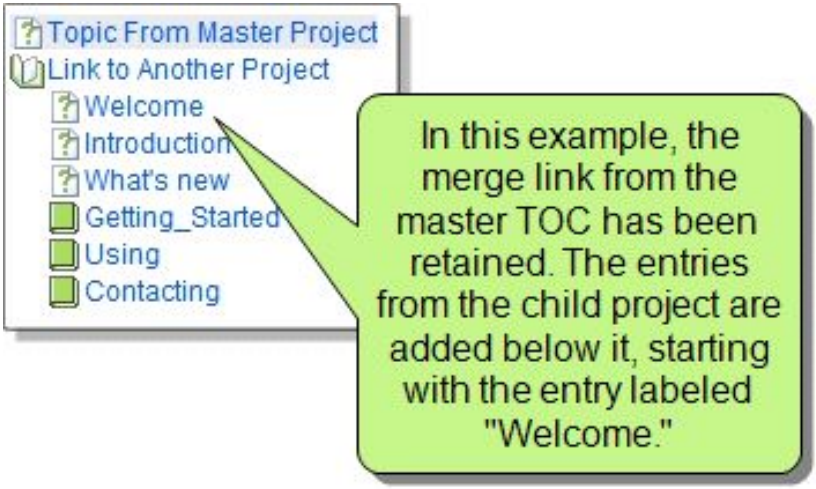
For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



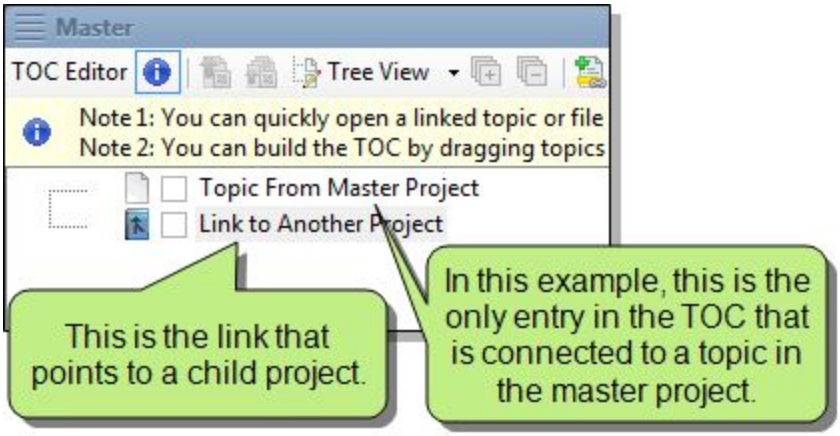
10. In the Properties dialog, click **OK**.
11. Build the output for the WebHelp target in your "master" project.

    Flare automatically creates the Subsystems folder for you. It also copies the target folder and .mcwebhelp file (along with the ancillary output files and folders) to the appropriate destination. When you open the TOC in the output, you will see the integrated TOC from the linked project.

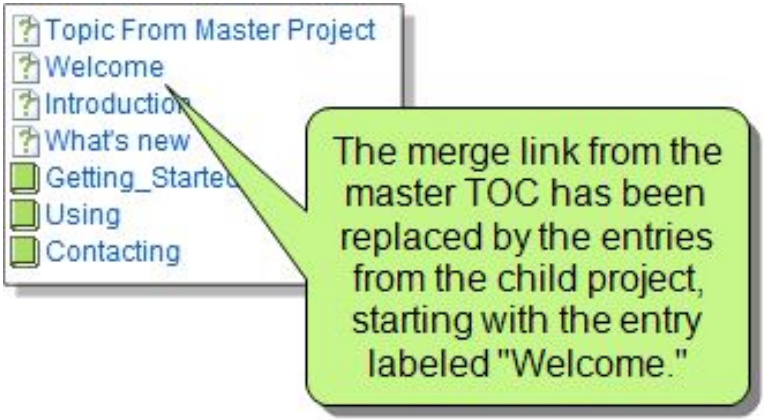
HOW TO MERGE OUTPUT FROM WEBHELP PROJECTS (MANUAL METHOD)

1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.
3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child WebHelp project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Type the name of the child target and main output file (e.g., Another-HelpSystem/AnotherHelpSystem.mcwebhelp).
7. Click **OK**.
8. (Optional) If you want the merged TOC to replace the entry that you selected, do the following:
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

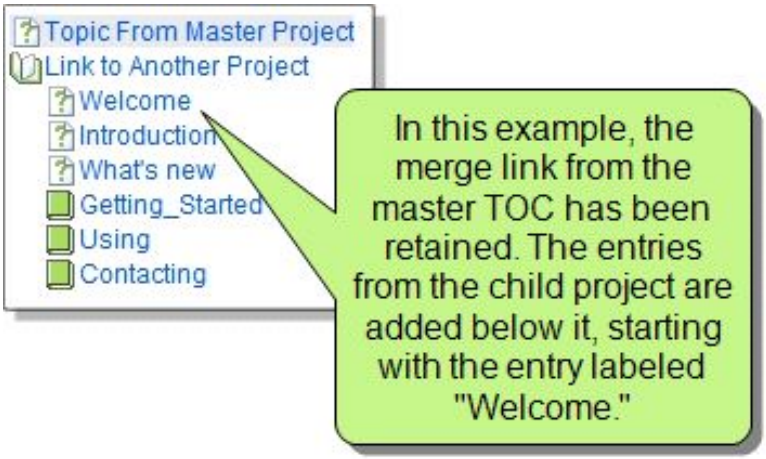
   For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



9. In the Properties dialog, click **OK**.
10. Build the output for the WebHelp target in your "master" project.
11. In the master Flare project, right-click on the target and select **Open Output Folder**. Within that folder, manually create another folder and name it Subsystems.
12. In the child Flare project, select **Project>Open Output Folder** (if using the ribbon view) or **Build>Open Output Folder** (if using the menu view).
13. Copy the target folder to be merged with the master project.
14. Paste the target folder within the Subsystems folder that you created in the master output folder.

Note: There are occasions when projects cannot be merged because the file name is the same in two or more of the linked projects (e.g., multiple DotNet Help projects all named "MyDotNetProject.mchelp," multiple WebHelp projects all named "MyWebProject.mcwebhelp," or multiple Microsoft HTML Help projects all named "MyHtmlHelp.chm"). The way to solve this dilemma is to make sure each linked project has a different output file name. To do this: (1) open the target, (2) select the **General** tab in the Target Editor, and (3) enter a unique name in the **Output File** field.

Note: If you plan to use Mark of the Web (MOTW) and are merging output from multiple Flare WebHelp projects, make sure you enable MOTW in all of the subprojects to be included in the merge. Otherwise, the output will not load smoothly into the browser window for the user. For more information about MOTW, see Microsoft's MSDN website.

Note: If you have previously merged Flare projects with WebHelp outputs and then decide to generate WebHelp Plus output from the master project instead, you must make sure that the child projects linked to the master contain at least one WebHelp Plus target each. Even if you keep the link from the master project to a WebHelp target in the child, the child project must also have a WebHelp Plus target.

Note: If you merge projects, synonym files will remain separate in each project. For example, if you create synonyms in Project A but not in project B, only the topics from Project A will use the synonyms when users perform searches in the output.

CHAPTER 6

More About WebHelp Plus

Because WebHelp Plus is a particularly powerful output type, there are several additional tasks to consider when using this format.

This chapter discusses the following:

Enabling WebHelp Plus Output 56
Installing IIS and ASP.NET 57
Setting Up a WebHelp Plus Target 67
Configuring IIS on Production Server 69
Starting Microsoft Indexing Service 71
Starting Microsoft Windows Search 72
Enabling WebHelp Plus Search 73
Testing WebHelp Plus Search 74
Viewing WebHelp Plus Output 75
Merging Output at Runtime Using WebHelp Plus 79
Including Non-XHTML Files in Search 87
WebHelp Plus Scenario 91
FAQs—WebHelp Plus 96

Enabling WebHelp Plus Output

If you want to take advantage of the advanced server-side features of WebHelp Plus (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable WebHelp Plus output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling WebHelp Plus search.

It is not necessary to install these in order to simply build WebHelp Plus output. However, if you want to be able to view and display WebHelp Plus output, the following steps are necessary. You must perform these steps on the machine that will be hosting the published WebHelp Plus output. If you want to view the output on your local machine to test the results, then you also need to perform these steps on your local computer, except for the procedure "Configuring IIS on Production Server."

For a complete scenario that illustrates these and other WebHelp Plus steps in a real-life situation, see "WebHelp Plus Scenario" on page 91.

Installing IIS and ASP.NET

The following steps show you how to install IIS and ASP.NET, depending on the operating system.

WINDOWS 8

The following steps show you how to install IIS and ASP.NET 4.5 for Windows 8.

INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Select **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. Select **Internet Information Services**.
5. Expand **Internet Information Services>Web Management Tools>IIS 6 Management Compatibility**.
6. Click **IIS Metabase and IIS 6 configuration compatibility**.

INSTALLING ASP.NET 4.5

7. Under **Internet Information Services** expand **World Wide Web Services**.
8. Expand **Application Development Features**.
9. Click the check box next to **ASP.NET 4.5** to add a check mark.
10. Click **OK**.

WINDOWS 7

The following steps show you how to install IIS and ASP.NET for Windows 7.

INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Select **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. Select **Internet Information Services**.
5. Expand **Internet Information Services>Web Management Tools>IIS 6 Management Compatibility**.
6. Click **IIS 6 Metabase and IIS 6 configuration compatibility**.

INSTALLING ASP.NET

7. Under **Internet Information Services** expand **World Wide Web Services**.
8. Expand **Application Development Features**.
9. Click the check box next to **ASP.NET** to add a check mark.
10. Click **OK**.
11. Click **Start**.
12. In the search field enter `run`.
13. Click **Run**. The Run dialog opens.
14. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

    32-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
    ```

    64-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
    ```

15. Click **OK**.

    A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

WINDOWS SERVER 2012

The following steps show you how to install IIS and ASP.NET 4.5 for Windows Server 2012.

INSTALLING IIS

1. Open the Server Manager.
2. Click **Add roles and features**. The Add Roles and Features Wizard opens.
3. Click **Next** two times until you ge to the Server Roles page.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Features**.

INSTALLING ASP.NET 4.5

6. In the Add Roles Wizard click **Next**.
7. Expand **.NET Framework 4.5 Features** and click **ASP.NET 4.5**.
8. Click **Next**.
9. At the Web Server Role (IIS) tab, click **Next** again.
10. Expand **Management Tools>IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
11. On the Confirm Installation Selections page of the wizard, click **Install**.
12. After the installation is completed, the Installation Results page opens. Click **Close**.

WINDOWS SERVER 2008 R2

The following steps show you how to install IIS and ASP.NET for Windows Server 2008 R2.

INSTALLING IIS

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**. The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Required Features**.
6. In the Add Roles Wizard click **Next**.
7. Click **Next** again.
8. On the "Select Role Services" page of the wizard, expand **Management Tools>IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
9. On the Confirm Installation Selections page of the wizard, click **Install**.
10. After the installation is completed, click **Finish**.

INSTALLING ASP.NET

1. In the Server Manager dialog, expand **Roles**.
2. Select **Web Server IIS**.
3. In the **Role Services** section, click **Add Role Services**.
4. Click the check box next to **ASP.NET** to add a check mark.
5. Click **Next**.
6. Click **Install**.
7. Click **Close**.
8. In the search field enter `run`.
9. Click **Run**.The Run dialog opens.

10. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

    32-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
    ```

    64-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
    ```

11. Click **OK**.

    A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

WINDOWS SERVER 2008

The following steps show you how to install IIS and ASP.NET for Windows Server 2008.

INSTALLING IIS

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**. The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Web Server (IIS)**.
5. In the dialog that opens click **Add Required Features**.
6. In the Add Roles Wizard click **Next**.
7. Click **Next** again.
8. On the "Select Role Services" page of the wizard, expand **Management Tools>IIS 6 Management Compatibility** and click **IIS 6 Metabase Compatibility**.
9. On the Confirm Installation Selections page of the wizard, click **Install**.
10. After the installation is completed, click **Finish**.

INSTALLING ASP.NET

1. In the Server Manager dialog, expand **Roles**.
2. Select **Web Server IIS**.
3. In the **Role Services** section, click **Add Role Services**.
4. Click the check box next to **ASP.NET** to add a check mark.
5. Click **Next**.
6. Click **Install**.
7. Click **Close**.

WINDOWS SERVER 2003

The following steps show you how to install IIS and ASP.NET for Windows Server 2003.

INSTALLING IIS VIA THE MANAGE YOUR SERVER DIALOG

1. From the Start menu open the Manage Your Server dialog.
2. Click **Add or remove a role**.The Configure Your Server Wizard opens.
3. Click **Next**.
4. Select **Application server (IIS, ASP.NET)** and click **Next**.
5. Click **Enable ASP.NET** and click **Next**.Your selections are summarized.
6. Click **Next**.The Windows Components Wizard opens, displaying the status of the installation.
7. After the installation is completed, click **Finish**.

INSTALLING IIS VIA THE CONTROL PANEL

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Add or Remove Programs**.The Add or Remove Programs window opens.
3. Click **Add/Remove Windows Components**.The Windows Components Wizard opens.
4. In the list of components select **Application Server** and then click **Details**.The Application Server Dialog opens.
5. Click the check box next to **Internet Information Services (IIS)** to add a check mark.
6. In the Windows Component Wizard, "Application Server" should now be selected. Click **Next**.
7. Click **Finish**.

INSTALLING ASP.NET

1. Click **Start>Run**.The Run dialog opens.
2. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

   32-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
   ```

   64-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
   ```

3. Click **OK**.

   A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

> **Note:** ASP.NET pages by default are prohibited in IIS. Therefore, you must set these pages to be allowed if you are running Windows Server 2003. To do this: (1) In Internet Information Services (IIS) Manager, select **Web Service Extensions** from the left pane; (2) in the right pane, make sure that **ASP.NET v4.0.30319** is set to **Allowed**. If it is not, select it and click the **Allow** button.

WINDOWS XP

The following steps show you how to install IIS and ASP.NET for Windows XP.

INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Add or Remove Programs**.The Add or Remove Programs window opens.
3. Click **Add/Remove Windows Components**.The Windows Components Wizard opens.
4. In the list of components select **Application Server** and then click **Details**.The Application Server Dialog opens.

> **Note:** The options in step 4 do not apply to the process for installing IIS on Windows XP Professional. If you are installing IIS on Windows XP Professional, skip step 4 and move on to step 5.

5. Click the check box next to **Internet Information Services (IIS)** to add a check mark.
6. In the Windows Component Wizard, "Application Server" should now be selected. Click **Next**.
7. Click **Finish**.

INSTALLING ASP.NET

1. Click **Start>Run**.The Run dialog opens.
2. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

   32-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
   ```

   64-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
   ```

3. Click **OK**.

   A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

WINDOWS VISTA

The following steps show you how to install IIS and ASP.NET for Windows Vista.

INSTALLING IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Programs and Features**.
3. Click **Turn Windows features on or off**.
4. Select **Internet Information Services**.
5. Expand **Internet Information Services>Web Management Tools>IIS 6 Management Compatibility**
6. Click **IIS 6 Metabase and IIS 6 configuration compatibility**.

INSTALLING ASP.NET

7. Under **Internet Information Services** expand **World Wide Web Services**.
8. Expand **Application Development Features**.
9. Click the check box next to **ASP.NET** to add a check mark.
10. Click **OK**.
11. Click **Start>Run**.The Run dialog opens.
12. In the **Open** field, copy and paste one of the following into the field, depending on whether you have a 32-bit or 64-bit system:

    32-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
    ```

    64-BIT

    ```
    C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
    ```

13. Click **OK**.

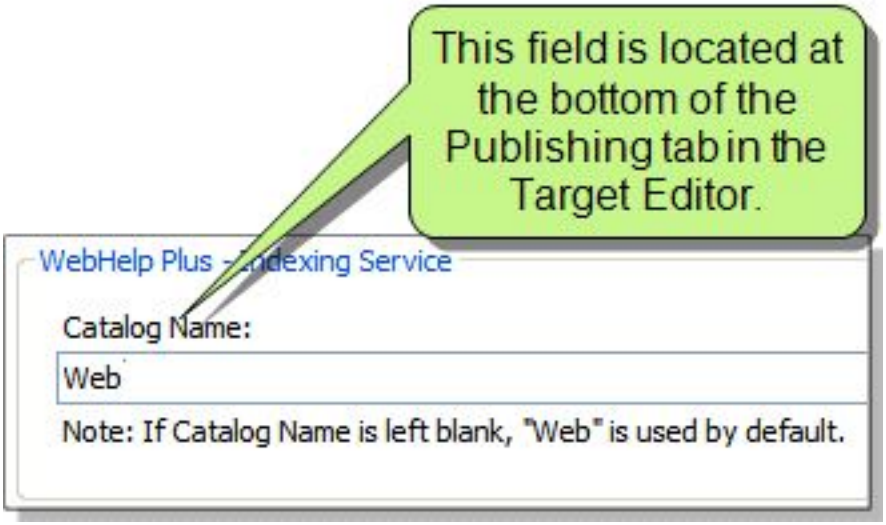
    A window opens, displaying the installation progress of ASP.NET. The window will close automatically when the installation finishes.

Setting Up a WebHelp Plus Target

The following steps show you how to set up your WebHelp Plus target for server-based output.

HOW TO SET UP THE WEBHELP PLUS TARGET

1. In Flare, open the WebHelp Plus target.The Target Editor opens.
2. Select the **Publishing** tab.
3. In the **Indexing Service Catalog Name** field, type the catalog that you are using for the output. In most cases, this will be `Web`, which is the default value. However, if you or someone in your company (e.g., network administrator) creates a custom catalog, you need to enter that name in the field.



What is a catalog? Microsoft Indexing Service stores all of its index information in catalogs. A catalog comprises index information and stored properties for a particular group of file system directories. When Indexing Service is installed with Windows XP, it automatically builds a catalog, called the System catalog, listing contents of all permanently attached disk drives. The System catalog contains an index for all documents except certain system and temporary files. If Internet Information Services (IIS) is installed, the Indexing Service also creates a web catalog, which contains an index of IIS, the default virtual server of the World Wide Web.

Note: This step is necessary only for Windows XP and Windows Server 2003.

4. Click to save your work.
5. Generate the target.
6. Publish the output to any location on the web server. If you want to use the default location that was created after you installed IIS, you can publish the output to C:\Inetpub\wwwroot. If you do this, you can use the "Default Web Site" folder that you will encounter when performing the next set of steps (i.e., configuring IIS on the production server).

   If you are working on your local machine for testing purposes, you do not need to publish the output. You can simply view your WebHelp Plus output. When you view WebHelp Plus output on your local computer, Flare creates a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder, and it places a copy of your WebHelp Plus output files in it. This enables you to test the advanced features of WebHelp Plus on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.)

Configuring IIS on Production Server

The following steps show you how to configure IIS, depending on the operating system. These steps are necessary only for the server where you will be publishing the final output. It is not necessary to perform these steps on your local computer for testing WebHelp Plus output.

WINDOWS SERVER 2003 AND WINDOWS XP

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Administrative Tools**.The Administrative Tools dialog opens.
3. Double-click **Internet Information Services**.The Internet Information Services dialog opens.
4. Find your output folder and right-click the **Service** subfolder. Then select **Properties**.The Service Properties dialog opens.
5. Select the **Virtual Directory** tab.
6. Next to the **Application name** field, click the **Create** button.

> **Note:** If this button has already been selected, it will display as "Remove" instead. In this case, you do not need to click the button.

7. In the **Execute Permissions** drop-down, make sure **Scripts only** is selected. It should already be selected by default.
8. Click **OK**.

WINDOWS SERVER 2008 AND 2008 R2, WINDOWS 2012, WINDOWS 7, WINDOWS 8, AND WINDOWS VISTA

1. Open the Control Panel.
2. Select **System and Security**.
3. Select **Administrative Tools**. The various operating systems have different ways to find this option.
4. Open the Internet Information Services (IIS) Manager.
5. Expand the node with the computer name.
6. Expand the **Sites** folder.
7. Expand the website folder corresponding to the one that you published.
8. Find your output folder and right-click the **Service** subfolder. Then select **Convert to Application**.

9. Make sure the the application pool is using the .NET 4 framework.
    a. In the dialog click **Select**.
    b. From the drop-down select an application pool that is using the .NET 4 framework.
    c. Click **OK**.
10. Click **OK**.

Starting Microsoft Indexing Service

The following steps show you how to start the Microsoft Indexing Service on machines that are running Windows XP and Windows Server 2003. This procedure is not necessary for the other operating systems.

WINDOWS SERVER 2003 AND WINDOWS XP—HOW TO START MICROSOFT INDEXING SERVICE

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Administrative Tools**.The Administrative Tools dialog opens.
3. Double-click **Computer Management**.The Computer Management dialog opens.
4. Expand **Services and Applications**.
5. Right-click **Indexing Service**.
6. In the context menu, click **Start**. The Indexing Service is now started.

Starting Microsoft Windows Search

The following steps show you how to start the Microsoft Windows Search on machines that are running Windows Server 2008. This procedure is not necessary for the other operating systems.

WINDOWS SERVER 2008—HOW TO START MICROSOFT WINDOWS SEARCH

1. From the Start menu open the Server Manager dialog.
2. Click **Add roles**.The Add Roles Wizard opens.
3. Click **Next**.
4. Click the check box next to **Files Services**.
5. Click **Next** twice.
6. Click the check box next to **Windows Search Service**.
7. Click **Next** twice.
8. Click **Install**.
9. Click **Close**.

Enabling WebHelp Plus Search

The following steps show you how to enable WebHelp Plus search. These steps must be done for each catalog that you are using on the server (or on your local machine, if you are viewing WebHelp Plus output on your computer for testing purposes).

HOW TO ENABLE WEBHELP PLUS SEARCH

1. Navigate to the output folder for the WebHelp PlusHTML5 target on the server or on your local machine.

   If you are enabling WebHelp PlusHTML5 on your local machine, you can quickly find the output folder by doing the following.

   a. Open the Project Organizer.
   b. Double-click the **Targets** folder.
   c. Right-click on the target and select **Open Output Folder**.
2. In the output folder, double-click the **Service** folder.
3. Double-click the folder labeled **Console.ConfigureSearch**.
4. Double-click **ConfigureSearch.exe**. A window appears very briefly and then disappears.
5. Create a folder called "AutoMergeCache" at the root of the site. Then you must set security preferences on the AutoMergeCache folder so the application can create and update files.
   a. Right-click on the root folder and choose **Explore** to open Windows Explorer.
   b. Right-click on **AutoMergeCache** and choose **Properties**.
   c. On the **Security** tab click the **Edit** button.
   d. Click **Add**.
      i. Type `Everyone` and click **Check Names**, making sure it gets underlined.
      ii. Click **OK**.
   e. Make sure **Everyone** is highlighted and check the option for **Full Control**.
   f. Click **Apply**.
   g. Click **OK** to exit the Permission dialog.
   h. Click **OK** to exit the Properties dialog.

Testing WebHelp Plus Search

The following steps are optional for testing WebHelp Plus search (for Microsoft Windows Search).

HOW TO TEST SEARCH

1. In IIS right-click on the directory where your WebHelp Plus project is published to and choose **Explore**.
2. In the upper-right corner of the window you will see a Search input box. Perform a search for a term that is commonly found in your project.
3. If you see results, your project is ready to go live.

Viewing WebHelp Plus Output

You can view WebHelp Plus output on your local machine for testing purposes, and you can view the output on the server where the files are published (where users access them). Before viewing the output, however, you must first follow all of the steps for enabling WebHelp Plus. See "Enabling WebHelp Plus Output" on page 56.

Basics of Viewing WebHelp Plus Output

When you view WebHelp Plus output on your local computer, Flare creates a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder, and it places a copy of your WebHelp Plus output files in it. This enables you to test the advanced features of WebHelp Plus on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.)

When you test the output on your local computer, the output initially needs to be viewed by using the Flare interface (in order to generate the MCPreview folder). After that, you can view the output by using the Flare interface or by opening the file copies from the MCPreview folder. It is recommended that you use the MCPreview folder for subsequent viewings of WebHelp Plus output, rather than the Flare interface.

For a complete scenario that illustrates these and other WebHelp Plus steps in a real-life situation, see "WebHelp Plus Scenario" on page 91.

If you want to test WebHelp Plus on your local computer,the advanced search features of WebHelp Plus are not operable.

Testing Output Locally—Windows Vista and Windows 7

Windows Vista and Windows 7 include a feature called User Access Control (UAC). By default, this feature is enabled in Windows Vista and Windows 7, which means that all users, including administrators, run programs in an under-privileged state. It is not possible to view for testing on your local computer if Windows Vista or Windows 7 is running in this under-privileged state. If this occurs, you have two options: (1) disable UAC on the local machine, or (2) run Flare with elevated permission without globally disabling UAC.

OPTION 1—DISABLE UAC ON THE LOCAL MACHINE

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Select **User Accounts** (select it twice if you are in Category View).
3. (Windows Vista) Click **Turn User Account Control on or off**. Click the check box to remove the check mark. Click **OK**.

   (Windows 7) Click **Change User Account Control Settings** and then move the slider to **Never notify** and click **OK**.
4. Restart the computer.

   UAC is now disabled. However, in order to preview WebHelp Plus output, you must be part of the Administrator's group, because Standard users cannot preview the output. Check with your system administrator to determine if you are in this group.

OPTION 2—RUN FLARE WITH ELEVATED PERMISSION WITHOUT GLOBALLY DISABLING UAC

1. Make sure you are part of the Administrator's group. Check with your system administrator to determine if you are in this group.
2. In Windows, navigate to the Flare executable file (C:\Program Files\MadCap Software\MadCap Flare V10\Flare.app\Flare.exe).
3. Set yourself as an administrator for Flare (either temporarily or permanently).

   TO RUN ONLY A SINGLE INSTANCE OF FLARE AS AN ADMINISTRATOR

   a. Right-click on the executable file.
   b. Select **Run as administrator**.

   TO ALWAYS RUN FLARE AS AN ADMINISTRATOR

   a. Right-click on the executable file.
   b. Select **Properties**.
   c. Select the **Compatibility** tab.
   d. In the **Privilege Level** area, click in the check box labeled **Run this program as an administrator**.

Procedures for Viewing Output

Use the following procedures to view WebHelp Plus output on the production server (where you publish your final output files) and on your local computer (for testing purposes).

HOW TO VIEW OUTPUT ON THE PRODUCTION SERVER

1. Open a browser window on any computer.
2. In the address field, type the URL to the main entry file of your output files (where you published them).

   Unless you change the name of the output file (on the General tab of the Target Editor), the name of the main output file is Default.htm.

   EXAMPLE

   http://www.mycompanysite/help/myWHPtarget/Default.htm

3. Press **Enter**.

HOW TO VIEW OUTPUT ON THE LOCAL MACHINE FOR TESTING (INITIAL VIEWING)

After generating the output, do one of the following using the Flare interface.

» In the message that displays immediately at the end of the compilation process, click **Yes**.

OR

» Open the Target Editor and in the local toolbar click **View**.

HOW TO VIEW OUTPUT ON THE LOCAL COMPUTER (SUBSEQUENT VIEWINGS)

You can follow the same steps above for the initial viewing (using the Flare interface), or you can do the following to quickly access the files from the MCPreview folder.

1. Open a browser window.
2. In the address field, type the following:

   `http://localhost/MCPreview/[Name of Main Entry File].htm`

   For example: http://localhost/MCPreview/Default.htm

   > **Note:** The above address pertains to Windows XP and Windows Server 2003 only. If you are testing on a computer running Windows Vista, use the following instead: http://localhost:70/[Name of Project]/[Name of Main Entry File].htm. The portion ":70" refers to the port number. For example, if testing on Vista, you might enter http://localhost:70/MyProject/Default.htm.

3. Press **Enter**.

> **Note:** If you are testing WebHelp Plus output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

Merging Output at Runtime Using WebHelp Plus

This is an easy way to merge the output from multiple WebHelp Plus Flare targets into one Help system. These targets can be originated from the same Flare project or from different Flare projects. You simply place the output files in the correct location on the server (i.e., within your master project's AutoMerge folder). Flare then automatically merges the output from all of the targets when users access the Help. From the end user's perspective, the results are seamless, appearing as one large Help system. All of the TOCs, browse sequences, indexes, glossaries, and search capabilities for the projects are merged.

For a complete scenario that illustrates these and other WebHelp Plus steps in a real-life situation, see "WebHelp Plus Scenario" on page 91.

Tasks Associated with Merging Output Using WebHelp Plus

Following are basic tasks involved with merging output using WebHelp Plus.

HOW TO MERGE OUTPUT AT RUNTIME USING WEBHELP PLUS

1. **Enable WebHelp Plus Output** If you want to take advantage of the advanced server-side features of WebHelp Plus (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable WebHelp Plus output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 56.
2. **Determine Master Output** Decide which of your project outputs will serve as the "master." This is the main output that users will open. All other outputs will be accessed from that output, although it will appear as one large Help system to end users.
3. **(Optional) Specify TOC and/or Browse Sequence Locations for Automerge** By default, the TOCs and browse sequences from the secondary outputs will be appended at the end of the master output's TOC and browse sequence. However, if you want them to be appended somewhere within the master output's TOC or browse sequence, you can specify the exact location. See "Specifying the Automerge Location in a TOC for WebHelp Plus" on page 81 and "Specifying the Automerge Location in a Browse Sequence for WebHelp Plus" on page 83.
4. **(Optional) Specify Order of Merged Outputs** When you automerge WebHelp Plus outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Specifying the Order of Automerged WebHelp Plus Outputs" on page 85.
5. **Generate Outputs** Build the output for each of the targets to be included in the automerge. You must use WebHelp Plus for all of the targets.

6. **Publish Output to Server** Publish the output for the master project to the web server running Microsoft IIS. The output files for the secondary targets need to be published to the "AutoMerge" subfolder located in the master target's main output folder (e.g., http://www.xyzincorporated.com/help systems/MyTargetName/AutoMerge).

   When readers open the online output, they will see all of the WebHelp Plus outputs merged into one Help system. They can easily navigate and use the index and search across all of the output files.

Note: If you are testing WebHelp Plus output on your local machine, you need view the output at least one time. When you view WebHelp Plus output on your local computer, Flare creates a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder, and it places a copy of your WebHelp Plus output files in it. This enables you to test the advanced features of WebHelp Plus on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.) When testing the automerge feature on your computer, you need to place the secondary ("child") outputs in the AutoMerge subfolder at this location (as opposed to the output folder that was generated where your Flare project is located).

Note: If you are testing WebHelp Plus output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

Note: If you want to test WebHelp Plus on your local computer,the advanced search features of WebHelp Plus are not operable.

Specifying the Automerge Location in a TOC for WebHelp Plus

You can determine where other Flare project outputs are merged relative to your "master" project's TOC if you are generating WebHelp Plus output and you are publishing the files to a web server running Microsoft IIS.

By default, the other WebHelp Plus outputs will be merged at the end of your master project's TOC. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A TOC

1. Open the TOC in the Flare project that will serve as the parent project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the TOC (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new TOC item. To do this:
     a. Place your cursor in the TOC where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the TOC.
3. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected TOC entry.
   - **After** The automerge will occur immediately after the selected TOC entry.
   - **First Child** The automerge will occur at the first location directly after the selected TOC book (i.e., before any other entries within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.
   - **Last Child** The automerge will occur at the last location after the selected TOC book (after the last entry within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.

» **Replace** The automerge will occur at the location of the TOC entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master TOC so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Automerge Location in a Browse Sequence for WebHelp Plus

You can determine where other Flare project outputs are merged relative to your "master" project's browse sequence if you are generating WebHelp Plus output and you are publishing the files to a web server running Microsoft IIS.

By default, the other WebHelp Plus outputs will be merged at the end of your master project's browse sequence. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A BROWSE SEQUENCE

1. Open the browse sequence in the Flare project that will serve as the "master" project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the browse sequence (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new browse sequence item. To do this:
     a. Place your cursor in the browse sequence where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the browse sequence.
3. In the local toolbar of the Browse Sequence Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected browse sequence entry.
   - **After** The automerge will occur immediately after the selected browse sequence entry.
   - **First Child** The automerge will occur at the first location directly after the selected browse sequence book (i.e., before any other entries within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.

- **Last Child** The automerge will occur at the last location after the selected browse sequence book (after the last entry within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.
- **Replace** The automerge will occur at the location of the browse sequence entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master browse sequence so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Order of Automerged WebHelp Plus Outputs

When you automerge WebHelp Plus outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Merging Output at Runtime Using WebHelp Plus" on page 79.

E X A M P L E

Let's say you have a master project with a target named "Main" and three smaller projects with targets named "First Child," "Second Child," and "Third Child," respectively. If you place the three smaller project outputs in the AutoMerge subfolder where "Main" is published, their TOCs will be appended to the master project's TOC in the following order:

1. First Child
2. Second Child
3. Third Child

But what if you want "Third Child" to be appended first? You can use the steps in this topic to position it above the other targets. Therefore, the result would be:

1. Third Child
2. First Child
3. Second Child

Now let's say that you add two more targets (called "Final Child" and "Another Child") to the AutoMerge folder. If you do not adjust your custom order, Flare will automerge these outputs in alphabetical order at the end of your custom order. Therefore, the final result would be:

1. Third Child
2. First Child
3. Second Child
4. Another Child
5. Final Child

HOW TO SPECIFY THE ORDER OF AUTOMERGED OUTPUTS

1. Create an XML file and name it SortOrder.xml. You can do this by opening an editor such as Notepad. When you save the file, type the xml file extension at the end of the name.
2. Enter the following code into the blank file.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>ProjectA</Item>

<Item>ProjectB</Item>

<Item>ProjectC</Item>

</SortOrder>
```

3. Replace "ProjectA," "ProjectB," and "ProjectC" with the names of your own child targets. You can add or remove line items as necessary.

   If you were to use the example at the top of this topic, you would enter the following.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>Third Child</Item>

<Item>First Child</Item>

<Item>Second Child</Item>

</SortOrder>
```

4. Save the file.
5. Copy and paste the SortOrder.xml file into the AutoMerge directory of your published WebHelp Plus output. This file sits beside the child output folders that you are using for the automerge.

Including Non-XHTML Files in Search

This feature is supported in WebHelp Plus output.

When end users perform a search in your online output, you can ensure that non-XHTML files (e.g. PDF, DOC, XLS) are included in that search. The files do not even need to be linked to any of the content in your Flare project. This feature is available if you publish WebHelp Plus output to a web server running Microsoft Internet Information Services (IIS).

For a complete scenario that illustrates these and other WebHelp Plus steps in a real-life situation, see "WebHelp Plus Scenario" on page 91.

HOW TO INCLUDE NON-XHTML FILES IN SEARCH

1. **Enable WebHelp Plus Output** If you want to take advantage of the advanced server-side features of WebHelp Plus (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable WebHelp Plus output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 56.
2. **Generate WebHelp Plus Output** In Flare, generate a target using the WebHelp Plus output format.
3. **Publish WebHelp Plus Output to Server** Publish the WebHelp Plus output to your Microsoft IIS Web server.
4. **(Optional) Add IFilter** File types that are automatically supported for WebHelp Plus search include: Microsoft Office files, HTML files, and TXT files. For other non-supported file types, such as PDF documents, you can download and install a simple IFilter in order to include that file type in the search. To find and download the IFilter for the file type that you want to include, go to a website that provides these (such as **http://www.ifilter.org**) and follow the instructions for downloading IFilters. You must install the appropriate IFilters on the server where the output files will be published, and if you are testing WebHelp Plus output on your computer, you need to install the IFilter(s) locally as well.
5. **Copy and Paste Non-XHTML Files to Appropriate Folder** Copy all of the non-XHTML files that you want to be included in the search. Then paste them in the appropriate subfolder of the WebHelp Plus output that you published. By default, WebHelp Plus will find any files located in the project output folder, as well as the "AutoSearch" subfolder. Therefore, you can paste your extra non-XHTML files in the AutoSearch subfolder.

E X A M P L E

http://www.xyzincorporated.com/help systems/MyTargetName/AutoSearch.

However, if there are any other folders where you want either XHTML or non-XHTML files to be stored (instead of, or in addition to, AutoSearch), you can perform some extra steps to allow this.

Let's say you also want searches in the Help system to provide results from files in the folder C:\ExternalSearchFolder. Here are the steps to allow this:

1. This folder must be accessible by IIS. Therefore, create a virtual directory in IIS for it. Following are steps for doing this.

    a. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.

    b. Double-click **Administrative Tools**. The Administrative Tools dialog opens.

    c. Double-click **Internet Information Services**. The Internet Information Services (IIS) Manager dialog opens.

    d. Expand the **Web Sites** folder.

    e. Right-click on the **Default Web Site** folder and select **New>Virtual Directory**.

    f. Click **Next**.

    g. In the **Alias** field, enter a name for the subfolder (e.g., SearchFolder).

    h. Click **Next**.

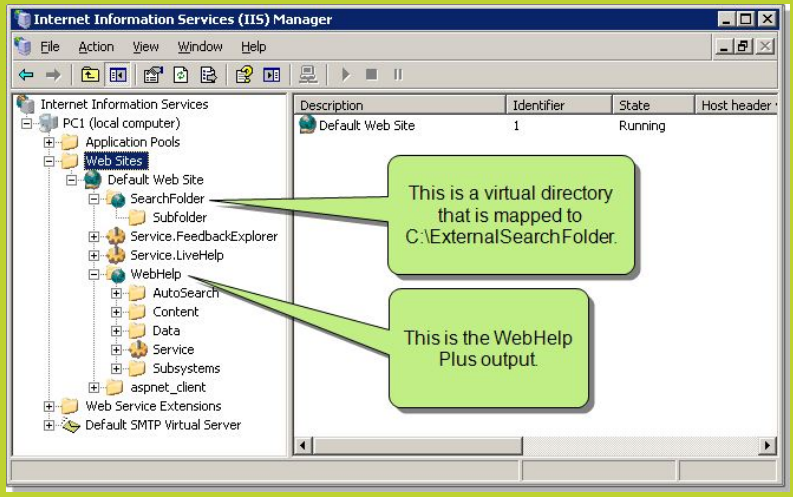
    i. Click the **Browse** button, then find and select the folder holding the content that you want to publish (e.g., C:\ExternalSearchFolder).

    j. Click **Next**.

    k. Make sure a check mark is next to **Read**.

l. Click **Next**.

m. Click **Finish**.



2. Create an XML file called "SearchFolders.xml" and place it in the "AutoSearch" folder in the WebHelp Plus output directory. The contents of this file will specify any virtual directories that WebHelp Plus should search. Following is what you would type in the SearchFolders.xml file. This example specifies just one folder, but you can add references to as many virtual directories as you need.

```
<?xml version="1.0" encoding="utf-8"?>

  <SearchFolders>

     <Url>/SearchFolder</Url>

  </SearchFolders>
```

When users perform a search in your output, those non-XHTML files will also be scanned and become accessible to the users.

Note: If you are testing WebHelp Plus output on your local machine, you need to view the output at least one time. When you view WebHelp Plus output on your local computer, Flare creates a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder, and it places a copy of your WebHelp Plus output files in it. This enables you to test the advanced features of WebHelp Plus on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.) When testing this search feature on your computer, you need to place the non-XHTML files in the AutoSearch subfolder at this location (as opposed to the output folder that was generated where your Flare project is located).

Note: If you are testing WebHelp Plus output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

Note: If you want to test WebHelp Plus on your local computer,the advanced search features of WebHelp Plus are not operable.

WebHelp Plus Scenario

The following scenario provides a real-life example of how someone might set up a local computer and server for producing WebHelp Plus output. For specific steps of the tasks represented in this scenario, see "Enabling WebHelp Plus Output" on page 56 and "Viewing WebHelp Plus Output" on page 75.

S C E N A R I O

Here is the situation: An author is running Windows XP on her computer. She has a Flare project with a target named "Jackpot" and plans to publish the output to the company website (xyzincorporated.com), using the WebHelp Plus format. In addition, the author has 20 Microsoft Word documents and 15 PDF files that she wants to be included in user searches. Finally, she has four other Flare projects and wants to merge them with the output from the Jackpot target, which will serve as the "parent" target. The author wants to publish the output to this directory on the server: http://www.xyzincorporated.com/help systems. But before she publishes the final output, she wants to test the output on her local computer. The following steps in this scenario require the completion of some tasks by the author and the completion of other tasks by the web administrator, who is responsible for the production server where the final output files will be placed.

1. Author On her local computer, the author completes the steps for installing Microsoft Internet Information Services (IIS) and ASP.NET.

   See "Installing IIS and ASP.NET" on page 57.

2. Author In Flare, the author follows the steps for setting up the Jackpot target, which is based on the WebHelp Plus output type. This includes entering "Web" as the catalog name on the Publishing tab (we're assuming the intention is to use the default value, as opposed to a custom catalog).

   See "Setting Up a WebHelp Plus Target" on page 67.

3. Author The author generates the Jackpot target in Flare. At the end of the compilation process, she selects "Yes" to view the output. As soon as she does this, the MCPreview folder is automatically created on her computer at C:\Inetpub\wwwroot, and Flare places a copy of the Jackpot output files in that folder.

4. Author The author follows the steps for starting Microsoft Indexing Service.

   See "Starting Microsoft Indexing Service" on page 71.

5. Author In order to enable the advanced WebHelp Plus search features on her computer, the author opens the output folder and double-clicks the file ConfigureSearch.exe, which is located in the output's

Service\Console.ConfigureSearch subfolder. In the case of this author, she finds her executable file here:

C:\Documents and Settings\AuthorName\Documents\My Projects\JackpotProject\Output\authorname\ Jackpot\Service\Console.ConfigureSearch\ConfigureSearch.exe

See "Enabling WebHelp Plus Search" on page 73.

6. Author The author generates her four "child" Flare targets (those that will be merged with the parent Jackpot target). In Flare, she selects **Project>Open Output Folder** and copies each output folder related to a child target. She then pastes these output folders in the following location:

   C:\Inetpub\wwwroot\MCPreview\AutoMerge

   The author is careful not to view the output of any of the child targets from Flare immediately after she builds them. If she were to do this, the output files for the parent Jackpot project would be overwritten in the MCPreview folder.

   You can also choose specific places in the table of contents or browse sequence where child projects are merged. See "Specifying the Order of Automerged WebHelp Plus Outputs" on page 85, "Specifying the Automerge Location in a TOC for WebHelp Plus" on page 81, and "Specifying the Automerge Location in a Browse Sequence for WebHelp Plus" on page 83. In this scenario, let's assume the author simply uses the default merge order.

7. Author Now the author wants to add the Word and PDF files to the output, so that they can be included in searches. However, an initial step needs to be completed first. Certain files (Microsoft Office, text, and HTML) do not need any attention because they can be included in the search just the way they are. PDF documents, however, require an IFilter. Therefore, the author downloads an IFilter from Adobe and installs it on her computer.

   See "Including Non-XHTML Files in Search" on page 87.

8. Author The author copies and pastes the Word and PDF documents to:

   C:\Inetpub\wwwroot\MCPreview\AutoSearch

9. Author The author views the WebHelp Plus output on her computer to test the merged output and the search for Word and PDF content. However, this time she does not initiate the view from within Flare. This time, she opens a browser window and types the following path into the address bar:

   http://localhost/MCPreview/Default.htm

She does this for two reasons: (1) the output loads faster, and (2) if she had initiated the view from within Flare, the output files in the MCPreview folder would have been replaced with new files, which would have overwritten the merged child output, Word documents, and PDF files that were added manually.

See "Viewing WebHelp Plus Output" on page 75.

10. Author Let's say the author is now finished testing her output and is ready to publish the final output to the server. Therefore, in Flare she creates a publishing destination for the parent Jackpot target that points to http://www.xyzincorporated.com/help systems/Jackpot. She generates the output for the parent Jackpot target again and publishes the Jackpot WebHelp Plus output to that location.

11. Author/Web Administrator The author generates the four child Flare target outputs again. In Flare, she selects **Project>Open Output Folder** and copies each output folder related to a child target. The author can provide these folders to the web administrator or (if the author has access) directly paste them in the following location:

    http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge

    For example, if the author has child targets named "Child1," "Child2," "Child3," and "Child4," those output folders would be added to the website as follows:

    http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child1

    http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child2

    http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child3

    http://www.xyzincorporated.com/help systems/Jackpot/AutoMerge/Child4

    Alternatively, the author (or web administrator) could manually create subfolders on the website that are named after each target folder; these subfolders would be contained within the AutoMerge folder. The author can then create publishing destinations that point to the appropriate subfolders and automatically publish each child target accordingly.

12. Web Administrator On the server where the output files will be published, the company's web administrator completes the steps for installing Microsoft Internet Information Services (IIS) and ASP.NET.

    See "Installing IIS and ASP.NET" on page 57.

13. Web Administrator On the server where the output files will be published, the company's web administrator completes the steps for configuring IIS on the production server.

    See "Configuring IIS on Production Server" on page 69.

14. Web Administrator On the server where the output files will be published, the company's web administrator completes the steps for starting Microsoft Indexing Service.

    See "Starting Microsoft Indexing Service" on page 71.

15. Web Administrator The web administrator opens the output folder on the server to enable WebHelp Plus search. He double-clicks the file ConfigureSearch.exe, which is located here:

    http://www.xyzincorporated.com/help systems/Jackpot/Service/Console.ConfigureSearch

    See "Enabling WebHelp Plus Search" on page 73.

16. Web Administrator In order to include the PDF files in the search, the web administrator installs the appropriate IFilter on the server.

    See "Including Non-XHTML Files in Search" on page 87.

17. Author/Web Administrator The author (if she has access) or web administrator copies and pastes the 20 Word documents and 15 PDF files to:

    http://www.xyzincorporated.com/help systems/Jackpot/AutoSearch

18. Author To view the published output, the author opens a browser window and types the following path into the address bar:

    http://www.xyzincorporated.com/help systems/Jackpot/Default.htm

    If the author had specified a custom file name for the Jackpot target, such as "JackpotOnline," the address instead would be:

    http://www.xyzincorporated.com/help systems/Jackpot/JackpotOnline.htm

Note: If you are testing WebHelp Plus output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

FAQs—WebHelp Plus

Following are some frequently asked questions about WebHelp Plus.

DOES WEBHELP PLUS SUPPORT 32- AND 64-BIT IIS?

Yes.

WHAT DOES CONFIGURESEARCH.EXE DO? WHAT SERVER SECURITY IS REQUIRED TO ALLOW THIS PROGRAM TO PERFORM CORRECTLY?

WebHelp Plus uses Indexing Service for searching in Windows XP and Windows Server 2003. The content from a WebHelp Plus output is indexed by a catalog in the Indexing Service. This catalog must be specified in the WebHelp Plus target. The ConfigureSearch.exe utility configures the specified catalog for the WebHelp Plus output. It only needs to be run once per catalog; therefore, if another WebHelp Plus output on the same web server uses the same catalog, the utility does not need to be run again for that output.

WebHelp Plus uses Windows Search for searching in Windows Vista, Windows 7, and Windows Server 2008. The content from a WebHelp Plus output is indexed by Windows Search. The ConfigureSearch.exe utility adds the WebHelp Plus content to the Windows Search index .

E X A M P L E

Let's say you have a WebHelp Plus target and you are publishing to Windows XP. You have specified that it should use a catalog called "MyCatalog."

Generate the output and publish it to the destination web server. Then browse to the output folder on the web server. Within the output folder, locate the "Service\Console.ConfigureSearch" subfolder.

Run the ConfigureSearch.exe utility to configure the "MyCatalog" catalog.

Now let's say you have a second WebHelp Plus target that will be hosted on the same web server and is also configured to use the catalog called "MyCatalog." Since it is sharing the same catalog, you do not need to run the ConfigureSearch.exe utility again.

The user account that runs this utility needs to have "administrator" rights.

HOW CAN YOU TELL IF THE INDEXING SERVICE IS CONFIGURED CORRECTLY TO SUPPORT WEBHELP PLUS?

1. Open the Control Panel.
2. Open **Administrative Tools**.
3. Open **Computer Management**.
4. In the Computer Management window, navigate to **Services** and **Applications>Indexing Service**.
5. Expand **Indexing Service** and you will see an entry for your desired catalog.
6. Expand the entry for the catalog and click the entry called **Properties**. This will display a list of properties in the right pane.
7. Locate the two properties called **MadCapSearchable** and **MadCapConcept**. To sort the list by property name, click the **Property** column heading.
8. If both of those properties exist, the Indexing Service catalog is correctly configured.

HOW CAN YOU TELL IF THE WINDOWS SEARCH IS CONFIGURED CORRECTLY TO SUPPORT WEBHELP PLUS?

1. Open the Control Panel.
2. Open **Indexing Options**. The various operating systems have different ways to find this option.
3. If you see your output folder listed, it is configured correctly.

CAN MULTIPLE INSTANCES OF WEBHELP PLUS BE PRESENT (I.E., DIFFERENT FLARE PROJECTS) ON THE SAME IIS WEBSITE?

Yes. In addition, for Windows XP and Windows Server 2003 each unique catalog that is used for different WebHelp Plus outputs on the same web server needs to be configured separately using the ConfigureSearch.exe utility.

THE TABLE OF CONTENTS AND SEARCH FUNCTIONS SEEM TO REQUIRE SPECIAL SUPPORT OR PARTICULAR CONFIGURATION ON THE WEB SERVER IN ORDER TO FUNCTION. DOES ANY OTHER PORTION OF WEBHELP PLUS REQUIRE SPECIAL SUPPORT?

The table of contents, index, search, favorites, glossary, and browse sequence panes use a web service that is part of the WebHelp Plus output.

IF THE TABLE OF CONTENTS DOES NOT WORK IN WEBHELP PLUS OUTPUT, WHAT DO YOU CHECK IN ORDER TO TROUBLESHOOT THE PROBLEM?

Verify the following.

VERIFY THAT ASP.NET HAS BEEN REGISTERED WITH YOUR WEB SERVER

1. Open a command prompt by navigating to **Start>Run**, type `cmd` into the text field, and click the **OK** button.
2. Execute one of the following commands, depending on whether you have a 32-bit or 64-bit system.

   32-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis -i
   ```

   64-BIT

   ```
   C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis -i
   ```

VERIFY THAT THE WEBHELP PLUS WEB SERVICE IS PROPERLY CONFIGURED

1. Open the Control Panel.
2. Open **Administrative Tools**. The various operating systems have different ways to find this option.
3. Open **Internet Information Services**.
4. Do the following (for Windows Server 2003 only).
   a. Browse to **Web Service Extensions**.
   b. In the pane on the right, select **ASP.NET v4.0.30319**.
   c. If its Status does not show "Allowed," click the **Allow** button.
5. Browse to the WebHelp Plus output folder under the website where it was published.
6. In the output folder, there is a folder called Service. This folder needs to be set up as a web application. Do the following, depending on the operating system.
   - (Windows XP and Windows Server 2003) Right-click the **Service** folder and select **Properties**.
   - (Windows 7, Windows Server 2008, and Windows Vista) Right-click the **Service** folder, select **Convert to Application**, and click **OK**. If you do not see this option, but rather "Manage Application," it is already configured. Skip to step 10.
7. Under the **Directory** tab, inspect the **Application Settings** area. If **Application name** is grayed out, click the **Create** button to its right.
8. The **Execute Permissions** drop-down should contain **Scripts only**.
9. Click the **OK** button.
10. The **Service** folder should have a special icon next to it rather than the usual icon that looks like a yellow folder.

11. Open a web browser and navigate to your WebHelp Plus output web service.

E X A M P L E

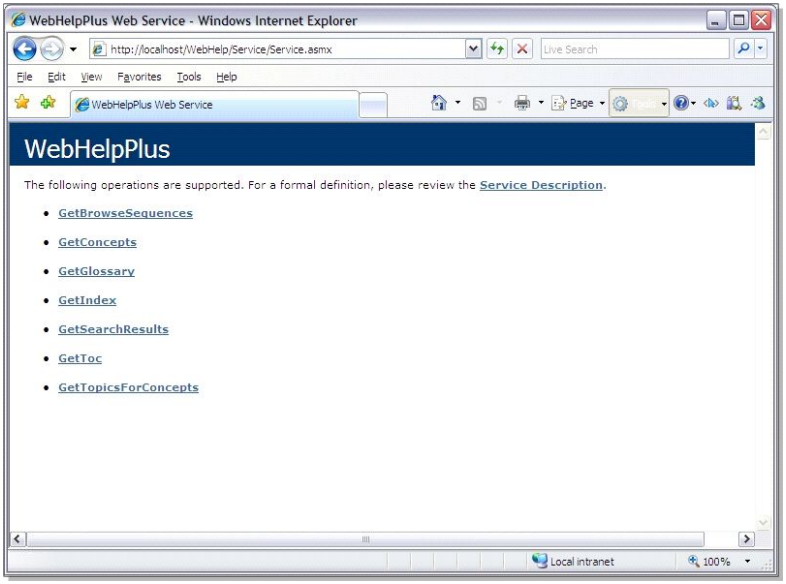
Let's say your output resides at:

http://localhost/MyWebHelpPlus

In that case, navigate to:

http://localhost/MyWebHelpPlus/Service/Service.asmx

12. If the web service is configured correctly, you will be taken to a web page that looks like the following:



In the WebHelp Plus output folder, there is an AutoMergeCache folder. This folder and the content inside it is automatically generated when the WebHelp panes (Table of Contents, Index, Glossary, etc.) need to be updated. It could be that the content in this folder was somehow corrupted. If so, delete the **AutoMergeCache** folder. Then view the WebHelp Plus output in a web browser so that the folder is re-created.

IF THE SEARCH FUNCTION DOES NOT WORK IN WEBHELP PLUS OUTPUT, WHAT DO YOU CHECK IN ORDER TO TROUBLESHOOT THE PROBLEM? (WINDOWS XP AND WINDOWS SERVER 2003)

Verify the following.

VERIFY THAT THE INDEXING SERVICE IS PROPERLY CONFIGURED

1. Open the Control Panel.
2. Open **Administrative Tools**.
3. Open **Computer Management**.
4. In the Computer Management window, navigate to **Services** and **Applications>Indexing Service**.
5. Expand **Indexing Service** and you will see an entry for your desired catalog.
6. Expand the entry for the catalog and click the entry called **Properties**. This will display a list of properties in the right pane.
7. Locate the two properties called **MadCapSearchable** and **MadCapConcept**. To sort the list by property name, click the **Property** column heading.
8. If both of those properties exist, the Indexing Service catalog is correctly configured.

VERIFY THAT THE INDEXING SERVICE HAS INDEXED THE CONTENT FROM YOUR OUTPUT

1. In the Computer Management window, select **Indexing Service**.
2. In the pane on the right, you will see a list of catalogs in the Indexing Service.
3. Check if the desired catalog still has documents to index. This can be seen in the **Docs to Index** column. If this number is not **0**, wait for the Indexing Service to finish indexing the files in the catalog.

VERIFY THAT YOU CAN PERFORM A SEARCH USING THE INDEXING SERVICE "QUERY THE CATALOG" UTILITY

1. In the Computer Management window, under the desired catalog in Indexing Service, there is a utility called **Query the Catalog**. Enter a term to search for that you know exists in the content of your WebHelp Plus output.
2. Click the **Search** button.
3. You should see content from your output in the list of search results.

VERIFY THAT YOU CAN PERFORM A SEARCH USING THE WEBHELP PLUS WEB SERVICE

1. From your web server, open a web browser and navigate to your WebHelp Plus output web service.

   E X A M P L E

   Let's say your output resides at:

   http://localhost/MyWebHelpPlus

   In that case, navigate to:

   http://localhost/MyWebHelpPlus/Service/Service.asmx

2. In the list of links, click the **GetSearchResults** link.
3. In the **searchString** field, enter a term to search for that you know exists in the content of your WebHelp Plus output.
4. Click the **Invoke** button. This will open a page containing search results in XML format.
5. You should see content from your output in the list of search results.

IS THERE SOME SORT OF DIAGNOSTIC OR SAMPLE FLARE OUTPUT TO DETERMINE IF WEBHELP PLUS WILL WORK OR IS CONFIGURED CORRECTLY?

Currently, there is not.

APPENDIX A

PDF Guides

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

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

Snippets Guide

Source Control Guide

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Variables Guide

WebHelp Outputs Guide

What's New Guide

APPENDIX B

Output Type Comparison Tables

Following are tables that show the various output types available, with the distinguishing features of each.

✓ Supported

✓ Supported with limitations

✗ Not supported

Note: If an item does not have a check mark, that does not necessarily mean the feature is unavailable in the output. Rather, it means that the feature cannot be added in Flare. For example, search cannot be added in Flare for EPUB output, but some ebook readers may have search built in.

General

Following are general features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
.NET Integration	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Main Entry File Extension	.mchelp	.xml or .jar	.chm	.htm	.htm	.htm	.htm	.air	.fm or .book	.pdf	.xml (default)	.htm	.xps	.epub or .mobi
Output Window or Application	MadCap Help Viewer	Eclipse Help Viewer	Microsoft HTML Help Viewer	Browser Window	Browser Window	Mobile Web Browser	Browser Window	Application Window	Adobe FrameMaker	Browser Window or Adobe PDF Reader	Microsoft Word	Browser Window	Browser Window	Ebook Reader or Web
Responsive Output	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Single Output File Possible	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✓

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Special Requirements	MadCap DotNet Help Viewer	Eclipse Standard 4.3.1 (Kepler) or higher Java (V7 Update 45) or later	Internet Explorer	✗	✗	✗	IIS ASP.NET MS Indexing Service	Java Runtime Environment (JRE) Adobe AIR	✗	✗	✗	✗	✗	Reader

Accessibility

Following are accessibility features (e.g., Section 508, WCAG) supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Accessibility Supported	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓
Compiler Warnings	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
Empty Alt Text for Images	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
Scrolling for Toolbars and Navigation Frames—Enable	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗

Context-Sensitive Help

Following are context-sensitive Help (CSH) features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
CSH Supported	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Embedded CSH Supported	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Feedback Statistics and Reporting Features

Following are Feedback statistics and reporting features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML 5	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Feedback Supported	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Search Results	✓	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗

Generated Content

Following are generated content features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Autonumbers	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Breadcrumbs	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Browse Sequences	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Concept Links	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Keyword Links	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
List of Concepts	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
List of Elements	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
List of Endnotes	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Lists (Numbered and Bul-leted)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Page Numbers	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗
Related Topics Links	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Relationship Links	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scripts	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Shortcut Controls	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Topic Toolbars	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

Glossaries

Following are glossary features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Glossaries—Created via Glossary File and Proxy	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Glossaries—Created via Glossary File and Skin Setting	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Auto-generate	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓

Horizontal Rules

Following are horizontal rule features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Horizontal Rules Supported	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓

Images

Following are image features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Images Supported	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Image Hyperlinks	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓
Image Maps	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Pre-Compile Resized	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓
Thumbnail Images	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Web-Safe—Convert from Non—Web-safe	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

Indexes

Following are index features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Indexes—Created Using Keywords and Proxy	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Indexes—Created Using Keywords and Skin Setting	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Auto-generate	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Binary Indexes	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Bookmarks in Index Entries	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Index Links	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Search—Can Exclude Index Entries from Search	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

Language Support

Following are language support features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Author and Generate Output in Left-to-Right (LTR) and Right-to-Left (RTL) Languages	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Invert Image Callouts to RTL	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Invert Page Layouts to RTL	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗
Invert Styles to RTL	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Output Interface—Display in LTR and RTL	✗	✓	✗	✓ Regular skin (UI Text tab)	✓ Language skin	✓ Language skin	✓ Language skin	✓ Language skin	✗	✗	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Output Interface—Display English, French, Japanese, or German	✓ Possible with Help Viewer (end user selects language)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Language Skins	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Separate UI Text tab in Skin Editor for Localization	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Master Pages and Page Layouts

Following are master page and page layout features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML 5	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Master Pages Supported	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗
Page Layouts Supported	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗

Merging Output

Following are merging output features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Merge Output Supported	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Merge Output at Runtime	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗

Miscellaneous Features

Following are miscellaneous features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Equations	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Footnotes	✓ Converted to popup	✓	✓ Converted to popup	✓ Converted to popup	✓ Converted to popup	✓ Converted to link	✓ Converted to popup	✓ Converted to popup	✓	✓	✓	✓	✓	✓
Object Positioning	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
QR Codes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Preview Topics in Output Type	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
Redacted Text	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Slideshows	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Snippets	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tables	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Text Boxes	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗

Multimedia

Following are multimedia features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Audio Supported	✓	✓	✓	✓	✓	✓ If mobile browser supports it	✓	✓	✗	✗	✗	✗	✗	✓ Depends on reader
Movies Supported	✓	✓ Does not support MP4 or MPG	✓	✓	✓	✓ If mobile browser supports it	✓	✓	✗	✗	✗	✗	✗	✓ Depends on reader
Movies—HTML5 Movie Format Supported (Depending on Browser)	✗	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

Navigation Links

Following are navigation link features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Cross-References Supported	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cross-References—Context-Sensitive	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Drop-down Text	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Expanding Text	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Text Hyperlinks	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Text Popups	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Togglers	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Topic Popups	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

Pulse (Socially Enabled Output)

Following are Pulse features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Pulse Integration	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗

Search

Following are search features supported in each online output type.

General Support

The table below summarizes search support in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML 5	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Search Supported Via Flare Integration	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

End User Search Features

The table below summarizes the key search features and capabilities for each output type. With HTML5 and WebHelp outputs, you can distribute either client- or server-based outputs. This is why the table below lists them twice—under client- and server-based output.

End user search features are supported by the Eclipse Help Viewer and search operations are provided by a plug-in that you develop using the Eclipse platform (see **org.eclipse.help.ui.searchEngine**). The Flare search engine does not handle Eclipse Help search operations.

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML 5	AIR	WebHelp Mobile	WebHelp Plus	HTML 5
Additional Requirements	Help Viewer SQL Server Compact (for asterisk (*) wildcard search and boolean operators)	Internet Explorer			Java Runtime Environment (JRE) Adobe AIR		IIS ASP.NET Microsoft Indexing Service or Windows Search	IIS ASP.NET Microsoft Indexing Service or Windows Search
Glossary Search—Find Matching Terms	✓	✓	✗	✓	✗	✗	✗	✓
Index Search—Find Matching Terms	✓	✓	✓	✓	✓	✗	✓	✓

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML	AIR	WebHelp Mobile	WebHelp Plus	HTML
Search Favorites—Save Queries to Favorites List	✓	✗	✓	✗	✓	✗	✓	✗
Search Query—Asterisk (*) Wildcard Character	✓ Requires SQL Server Compact.	✓	✓	✓	✗	✗	✗	✗
Search Query—Boolean Operators	✓ Requires SQL Server Compact.	✓ Select operator from built-in list.	✓	✓	✓	✓	✗	✗
Search Query—Full-text Search (not case sensitive)	✓	✓	✓	✓	✓	✓	✓	✓
Search Query—Enclose Terms in Quotes (" ")	✓	✓	✓	✓	✓	✓	✓	✓
Search Results—Narrow Search Scope Using Search Filter Sets	✓	✗	✓	✓	✓	✗	✓	✓

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML 5	AIR	WebHelp Mobile	WebHelp Plus	HTML 5
Search Results—Results Listed in Ranked Order and Show Rank Number	✓	✓ Results are listed in order. Rank number is not visible.	✓	✓ Results are listed in order. Rank number is not visible.	✓	✓ Results are listed in order. Rank number is not visible.	✓	✓ Results are listed in order. Rank number is not visible.
Search Results—Search Hits Highlighted in Topics	✓	✓	✓	✓	✓	✓	✓	✓

Content Developer Search Features

This table summarizes the key search features and capabilities that are available to you, the content developer, for each output type. With HTML5 and WebHelp outputs, you can distribute either client- or server-based outputs. This is why the table below lists them twice—under client- and server-based output.

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML5	AIR	WebHelp Mobile	WebHelp Plus	HTML5
Enable Search Feature	✓ Search cannot be disabled.	✓ Standard skin	✓ Standard skin	✓ HTML5 skin	✓ Standard skin	✓ Mobile skin	✓ Standard skin	✓ HTML5 skin
Search Highlighting—Set Highlight Color	✓	✓	✓	✓	✓	✓	✓	✓
Search Performance—Chunk Large Search Database Files	✓	✗	✓	✓	✓	✓	✓	✓
Search Performance—Enable Partial-word Searching and Set Minimum Word Size	✓	✗	✓	✓	✓	✓	✓	✓

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML 5	AIR	WebHelp Mobile	WebHelp Plus	HTML 5
Search Performance—Enable Stop Words in Search	✓	✓ Stop words always enabled.	✓	✓	✓	✓	✓	✓
Search Performance—Exclude Index Entries from Search	✓	✓ Index entries always excluded.	✓	✓	✓	✓	✓	✗
Search Performance—Exclude Non-words from Search	✓	✓ Non-words always excluded.	✓	✓	✓	✓	✓	✓
Search Performance—Include SQL Search Database	✓ Requires SQL Server Compact.	✗	✗	✗	✗	✗	✗	✗

	CLIENT-BASED OUTPUT						SERVER-BASED OUTPUT	
	DotNet Help	Microsoft HTML Help	WebHelp	HTML 5	AIR	WebHelp Mobile	WebHelp Plus	HTML 5
Search Performance—Pre-merge Search Database	✓	✗	✓	✓	✓	✓	✓	✓
Search Performance—Set N-Gram Size for Chinese, Japanese, and Korean Projects	✓	✗	✓	✓	✓	✓	✓	✓
Search Toolbar—Include and Customize a Quick Search Field and/or Select Search Button	✓	✗	✓	✗	✓	✗	✓	✗

Skins

Following are skin features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Skin Type	Standard	✗	Standard	HTML5	Standard	Mobile	Standard	Standard	✗	✗	✗	✗	✗	✗
About Box	✗	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Accordion Titles—Exclude	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Browser Settings	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Caption for Output Window	✗	✗ org.eclipse.help.base plug-in	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Elements (e.g., tabs, accordions)—Specify Default Element	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Elements (e.g., tabs, accordions)—Specify Which to Include	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Feedback Comments, Email Notifications, User Profile	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Generate All	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Language Skins	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Navigation Links in Stand-alone Topics	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Navigation Pane Settings	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Preview Skin for Output Type	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Separate UI Text Tab in Skin Editor for Localization	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Styles	✓ Only some styles (e.g., Feedback, toolbar) are supported.	✗	✓ Only some styles (e.g., Feedback, toolbar) are supported.	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
TOC Entry—Select Skin For	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Topic Tool-bar—Custom Settings	✓	✗	✓	✓ Topic tool-bar settings controlled in proxy or stylesheet, not in skin	✓	✓ Limited settings	✓	✓	✗	✗	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Web Toolbar—Add via Skin	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Web Toolbar—Custom Set-tings	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Window—Size/Position	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Window—User-Defined Size/Position	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Tables of Contents and Mini-TOCs

Following are table of contents (TOC) and mini-TOC features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
TOCs—Created via Headings and Proxy	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓ Some ebook readers have built-in TOC
TOCs—Created via TOC File and Skin Setting	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Auto-generate	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Binary TOCs	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mark as New	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗
Mini-TOCs	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Skin—Select for TOC Entry	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Synchronize With Topics	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗

Target Settings

Following are target features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Auto-generate Glossary, Index, TOC	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Characters and Spaces—Replace With Underscores	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
CMYK	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
Content Folder—Omit from Output	✗	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
Crop and Registration Marks—Include in Output	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
DOCTYPE Declaration	✓	✗	✓ Recommend disabling this option for non-English content	✗	✓	✓ Standard mobile DOCTYPE always used	✓	✓	✗	✗	✗	✗	✗	✗
Empty Pages—Inject Automatically	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
File Extensions—Custom	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Mark of the Web	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
Meta Tags—Custom	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Startup Topic	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Stylesheet Medium	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Warnings—Ignore	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Variables

Following are variable features supported in each output type:

	ONLINE								PRINT					OTHER
	DotNet Help	Eclipse Help	Microsoft HTML Help	HTML	WebHelp	WebHelp Mobile	WebHelp Plus	AIR	FM	PDF	WORD	XHTML	XPS	EPUB
Custom Variables	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Heading Variables	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Running HF Variables	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
System Variables	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓