



madcap
FLARETM

WebHelp Plus Guide

Version 6.1

Copyrights

Copyright 2010 MadCap Software. All rights reserved.

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

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

Contents

CHAPTER 1 About WebHelp Plus Output	1
Benefits of WebHelp Plus.....	2
Output Type Comparison Table—Online Output Types.....	3
Tasks Associated with WebHelp Plus.....	9
More Information About WebHelp Plus.....	9
CHAPTER 2 Enabling WebHelp Plus Output	11
Installing IIS.....	12
Setting Up WebHelp Plus Target.....	15
Configuring IIS on Production Server.....	17
Installing ASP.NET.....	18
Starting Microsoft Indexing Service.....	20
Starting Microsoft Windows Search.....	21
Enabling WebHelp Plus Search.....	22
CHAPTER 3 Viewing WebHelp Plus Output	25
Basics of Viewing WebHelp Plus Output.....	26
Testing WebHelp Plus Output Locally—Windows Vista and Windows 7.....	27
Procedures for Viewing WebHelp Plus Output.....	28
CHAPTER 4 Merging Output at Runtime Using WebHelp Plus	31
Tasks Associated with Merging Output Using WebHelp Plus.....	32
Specifying the Auto-Merge Location in a TOC.....	34
Specifying the Auto-Merge Location in a Browse Sequence.....	36
Specifying the Order of Auto-Merged Outputs.....	38
CHAPTER 5 Including Non-XHTML Files in Search	41
CHAPTER 6 WebHelp Plus Scenario	45
CHAPTER 7 FAQs—WebHelp Plus	51
Index	57

CHAPTER 1 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 XP, Windows Server 2003, Windows Server 2008, Windows 7, or Windows Vista. It also uses Microsoft Internet Information Services (IIS) and ASP.NET. To provide faster search, WebHelp Plus uses Microsoft Indexing Service (for Windows XP and Windows Server 2003) or Windows Search (for Windows Server 2008). 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 WebHelp Plus 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.

This chapter discusses the following:

Benefits of WebHelp Plus.....	2
Output Type Comparison Table—Online Output Types.....	3
Tasks Associated with WebHelp Plus.....	9
More Information About WebHelp Plus.....	9

Benefits Of WebHelp Plus

WebHelp Plus is a good choice if:

- You want all of the features and benefits available with WebHelp.
- You want users to experience faster searches in your output.
- You want non-XHTML files (e.g., PDFs, Excel spreadsheets) to be included in searches that users perform in your output (even if those non-XHTML files are not part of your project).
- You want to automatically merge the output for multiple Flare projects so that they appear as one large Help system to end users.
- You publish your output to a machine running Windows XP, Windows Server 2003, Windows Server 2008, Windows 7, or Windows Vista.

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

Output Type Comparison Table—Online Output Types

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

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
General						
.NET integration	✓					
Main entry file extension	.mchelp	.chm	.htm	.air	.htm	.htm
Output window	MadCap Help Viewer	Microsoft HTML Help Viewer	Browser window	Application window	Mobile Web browser	Browser window
Platform	Desktop	Desktop	Web	Desktop	Mobile Web	Web
Single output file		✓		✓		
Special requirements	MadCap DotNet Help Viewer	Internet Explorer		Java Runtime Environment (JRE) Adobe AIR		IIS ASP.NET MS Indexing Service
Context-sensitive Help (CSH)						
CSH supported	✓	✓	✓		✓	✓
Embedded CSH supported	✓					
Feedback statistics and reporting features						
Feedback supported	✓	✓	✓			✓
Search results	✓		✓			✓
Generated content						
Breadcrumbs	✓	✓	✓	✓	✓	✓
Browse sequences	✓	✓	✓	✓	✓	✓

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
Concept links	✓	✓	✓	✓	✓	✓
Glossaries	✓	✓	✓	✓	✓	✓
Indexes	✓	✓	✓	✓	✓	✓
Keyword links	✓	✓	✓	✓	✓	✓
Mini-TOCs	✓	✓	✓	✓	✓	✓
Related topics links	✓	✓	✓	✓	✓	✓
Relationship links	✓	✓	✓	✓	✓	✓
Scripts	✓	✓	✓	✓	✓	✓
Shortcut controls		✓				
TOCs	✓	✓	✓	✓	✓	✓
Topic toolbars	✓	✓	✓	✓	✓	✓
Language support						
Content—display in any left to right language	✓	✓	✓	✓	✓	✓
Output interface—display in any left to right language			✓	✓	✓	✓
Output interface—display English, French, Japanese, or German	✓ Possible with Help Viewer (end user selects language)					

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
Master pages						
Master pages supported	✓	✓	✓	✓	✓	✓
Merging output						
Merge output supported	✓	✓	✓	✓	✓	✓
Merge output at runtime						✓
Multimedia						
Audio	✓	✓	✓	✓	✓ If mobile browser supports it	✓
Movies	✓	✓	✓	✓	✓ If mobile browser supports it	✓
Search						
Search supported	✓	✓	✓	✓	✓	✓
Include/exclude topics in search	✓		✓	✓	✓ Only supported in newer mobile browsers	✓
Search filter sets	✓		✓	✓		✓
Search non-XHTML files						✓

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
Stop words	✓		✓	✓	Only supported in newer mobile browsers	Uses indexing service
Synonyms	✓		✓	✓	Only supported in newer mobile browsers	✓
Section 508						
Section 508 supported			✓	✓	✓	✓
Skin settings						
About box		✓	✓	✓	✓	✓
Accordion titles			✓	✓		✓
Browser settings			✓			✓
Caption for output window		✓	✓	✓	✓	✓
Elements (e.g., tabs, accordions)—specify default element		✓	✓	✓		✓
Elements (e.g., tabs, accordions)—specify which to include	✓	✓	✓	✓	✓	✓
Feedback user profile	✓	✓	✓			✓
Index—binary		✓				

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
Index—include bookmarks in index entries		✓				
Language skins		✓	✓	✓	✓	✓
Navigation links in standalone topics			✓	✓		✓
Navigation pane settings		✓	✓	✓		✓
Preview skin for output type	✓	✓	✓		✓	✓
Size/position—specify	✓	✓	✓	✓		✓
Styles	✓ Only some styles (e.g., Feedback, toolbar) are supported.	✓ Only some styles (e.g., Feedback, toolbar) are supported.	✓	✓	✓	✓
TOC—binary		✓				
TOC—synchronize with topics	✓	✓	✓	✓		✓
Topic toolbar custom settings	✓	✓	✓	✓	✓ Limited settings	✓
Target settings						
Characters and spaces—replace with underscores	✓	✓	✓	✓	✓	✓
Content folder—omit from output	✓	✓	✓	✓		✓

	DotNet Help	HTML Help	WebHelp	WebHelp AIR	WebHelp Mobile	WebHelp Plus
DOCTYPE declaration	✓	✓	✓	✓	Standard mobile DOCTYPE always used	✓
File extensions—custom			✓	✓	✓	✓
Images—pre-compile resized	✓	✓	✓	✓	✓	✓
Images—Web-safe	✓	✓	✓	✓	✓	✓
Language—select for output interface			✓	✓	✓	✓
Mark of the Web			✓			✓
Master page	✓	✓	✓	✓	✓	✓
Skins—generate all	✓	✓	✓	✓	✓	✓
Startup topic	✓	✓	✓	✓		✓
Style sheet medium	✓	✓	✓	✓	✓	✓

Tasks Associated With WebHelp Plus

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

- **Enable** In order 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), (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) installing ASP.NET, (5) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (6) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 11.
- **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 25.
- **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 31.
- **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 41.

More Information About WebHelp Plus

Following are additional sources of information about WebHelp Plus.

- **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 45.
- **FAQs** You can take a look at some frequently asked questions about WebHelp Plus. See "FAQs—WebHelp Plus" on page 51.

CHAPTER 2 Enabling WebHelp Plus Output

In order 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), (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) installing ASP.NET, (5) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (6) 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 45.

This chapter discusses the following:

Installing IIS	12
Setting Up WebHelp Plus Target	15
Configuring IIS on Production Server	17
Installing ASP.NET	18
Starting Microsoft Indexing Service	20
Starting Microsoft Windows Search	21
Enabling WebHelp Plus Search	22

Installing IIS

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

Windows Server 2003—How to install 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**.

Windows Server 2003—How to install 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**.

Windows Server 2008—How to install IIS

1. From the Start menu open the Manage Your Server 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**.

Windows 7—How to install 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**.
7. Click **OK**.

After IIS is configured and installed the dialog closes automatically.

Windows XP—How to install 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.
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**.

Windows Vista—How to install IIS

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Complete one of the following sets of steps, depending on the view you are using.

Home View:

- i. Select **Programs**.
- ii. Under **Programs and Features** click **Turn Windows features on or off**.

Classic View:

- i. Double-click **Programs and Features**.
 - ii. Select **Turn Windows features on or off**.
3. Select **Internet Information Services**.
 4. Expand **Internet Information Services>Web Management Tools>IIS 6 Management Compatibility**
 5. Click **IIS 6 Metabase and IIS6 configuration compatibility**.
 6. Click **OK**.

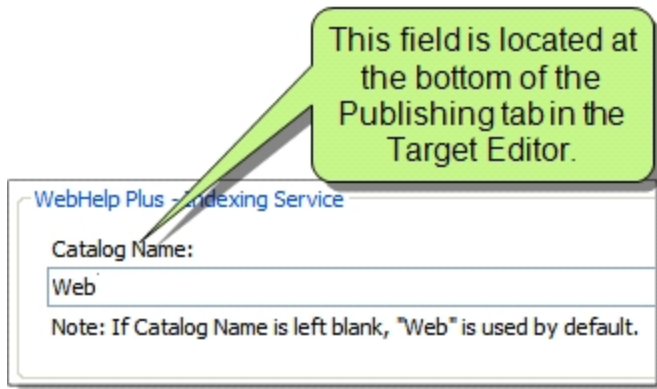
After IIS is configured and installed the dialog closes automatically.

Setting Up WebHelp Plus Target

The following steps show you how to set up your WebHelp Plus.


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 **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. Press **CTRL+S** or click  to save your work.
5. Generate the WebHelp Plus target.

6. Publish the WebHelp Plus 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 accomplish the same thing by simply viewing 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—How to configure IIS on the production server

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, Windows 7, and Windows Vista—How to configure IIS on the production server

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

Installing ASP.NET

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

Windows Server 2003 and Windows XP—How to install 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\v2.0.50727\aspnet_regiis -i
```

64-bit:

```
C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\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: If you are using Windows Server 2003, please note that 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 v2.0.50727** is set to **Allowed**. If it is not, select it and click the **Allow** button.

Windows Server 2008—How to install ASP.NET

1. From the Start menu open the Manage Your Server dialog.
2. Expand **Roles**.
3. Select **Web Server IIS**.
4. In the **Role Services** section, click **Add Role Services**.
5. Click the check box next to **ASP.NET** to add a check mark.
6. Click **Next**.
7. Click **Install**.
8. Click **Close**.


Windows 7 and Windows Vista—How to install ASP.NET

1. Open the Control Panel. On many computers, you can do this by clicking **Start>Control Panel**.
2. Double-click **Programs and Features**. The various operating systems have different ways to find this option.
3. Click **Turn Windows features on or off**.
4. Expand **Internet Information Services**.
5. Expand **World Wide Web Services**.
6. Expand **Application Development Features**.
7. Click the check box next to **ASP.NET** to add a check mark.
8. 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. Click the plus button  next to **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 Manage Your Server 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

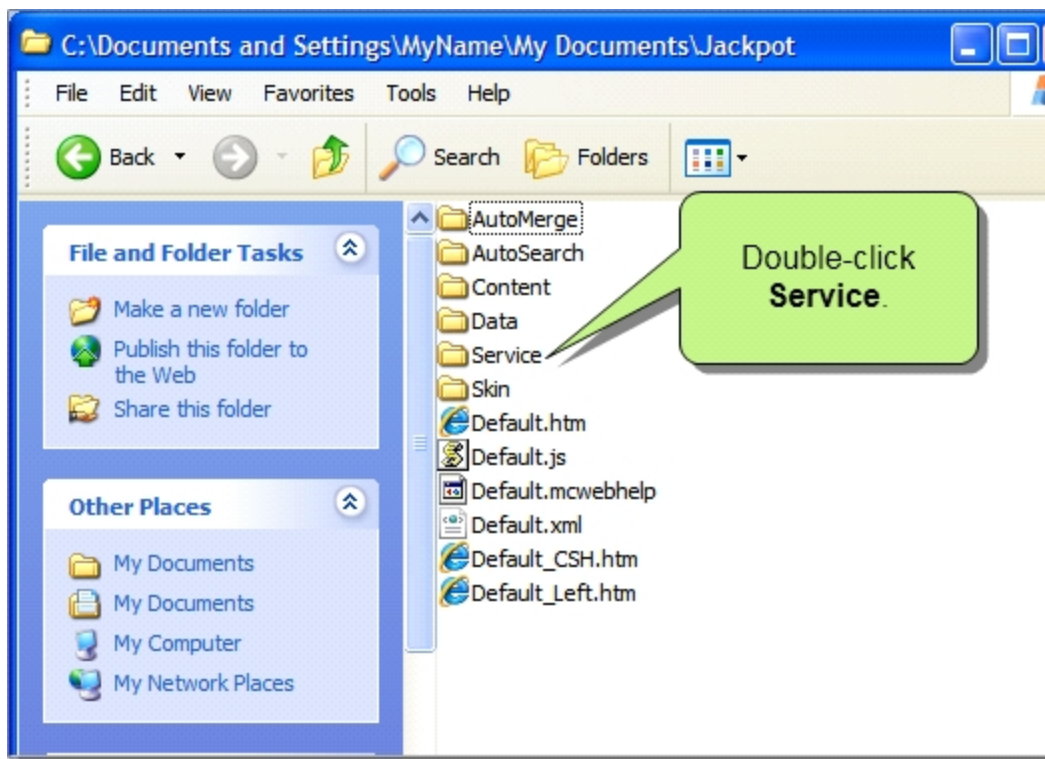
The following steps show you how to enable WebHelp Plus search. These steps must be done for each catalog that you are using on the server (or on your local machine, if you are viewing WebHelp Plus output on your computer for testing purposes).

How to enable WebHelp Plus search

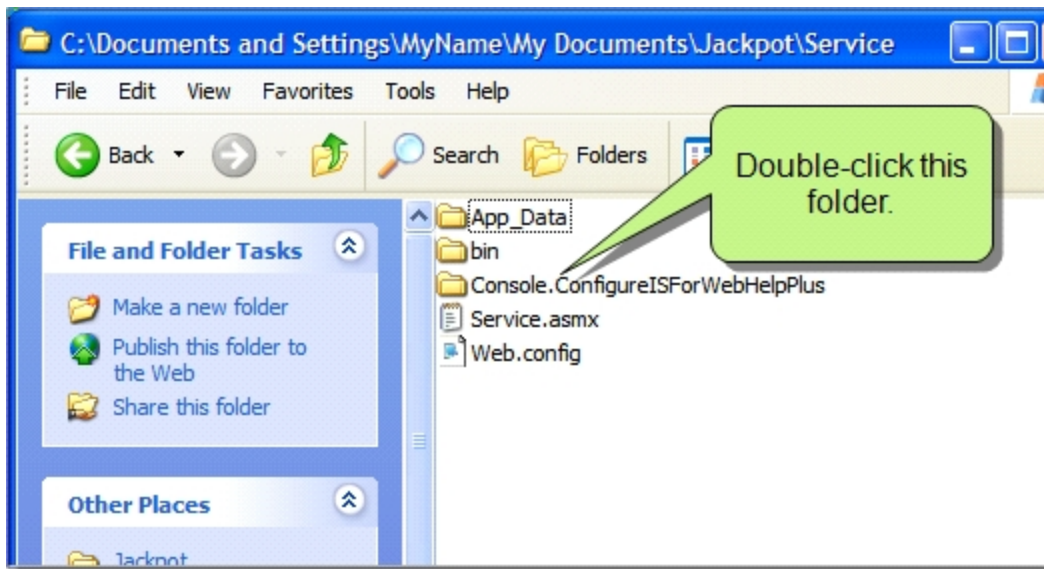
1. Navigate to the output folder for the WebHelp Plus target on the server or on your local machine.

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

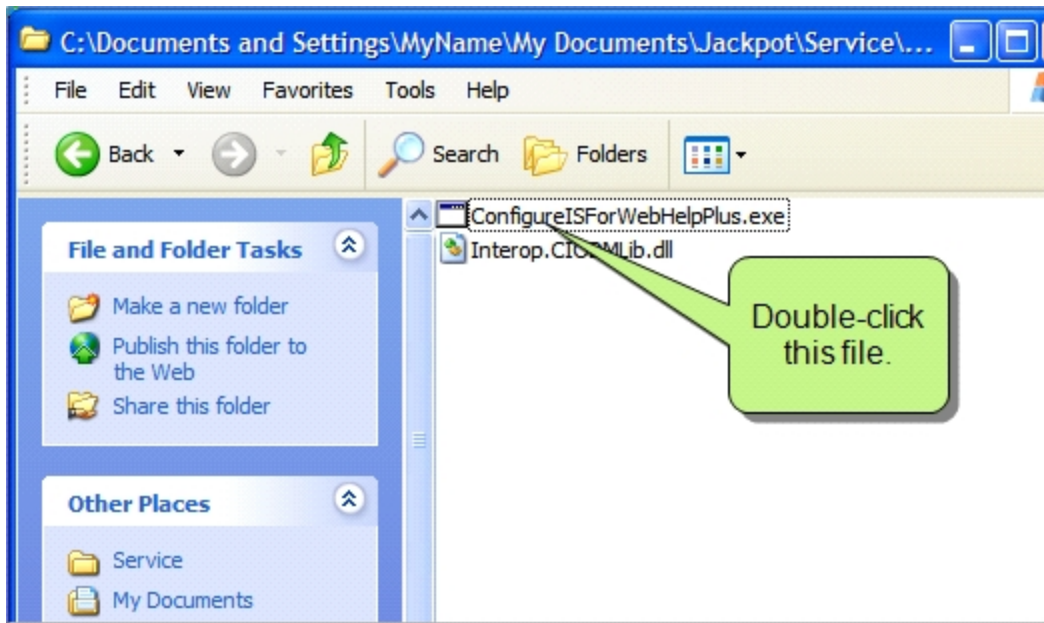
- a. Open the Project Organizer.
 - b. Double-click the **Targets** folder.
 - c. Right-click on the target and select **Open Output Folder**.
2. In the output folder, double-click the **Service** folder.



3. Double-click the folder labeled **Console.ConfigureISForWebHelpPlus**.



4. Double-click **ConfigureISForWebHelpPlus.exe**.



A window appears very briefly and then disappears.

CHAPTER 3 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 11.

This chapter discusses the following:

Basics of Viewing WebHelp Plus Output	26
Testing WebHelp Plus Output Locally—Windows Vista and Windows 7.....	27
Procedures for Viewing WebHelp Plus Output	28

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 45.

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

Testing WebHelp Plus 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 WebHelp Plus 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 V6\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 WebHelp Plus 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 WebHelp Plus 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


E X A M P L E

```
http://www.mycompanysite/help/myWHPTarget/Default.htm
```

3. Press **Enter**.

How to view WebHelp Plus 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
- In the Project toolbar, click  **View Primary**.

How to view WebHelp Plus 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.

CHAPTER 4 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 45.

This chapter discusses the following:

Tasks Associated with Merging Output Using WebHelp Plus.....	32
Specifying the Auto-Merge Location in a TOC.....	34
Specifying the Auto-Merge Location in a Browse Sequence.....	36
Specifying the Order of Auto-Merged Outputs.....	38

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** In order 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), (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) installing ASP.NET, (5) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (6) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 11.
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 auto-merge** 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 Auto-Merge Location in a TOC" on page 34 and "Specifying the Auto-Merge Location in a Browse Sequence" on page 36.
4. **(Optional) Specify order of merged outputs** When you auto-merge 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 Auto-Merged Outputs" on page 38.
5. **Generate outputs** Build the output for each of the targets to be included in the auto-merge. The master project must be generated using a WebHelp Plus target. The other targets can be generated using either WebHelp Plus or WebHelp. However, it is recommended that you use WebHelp Plus for all of the targets to ensure that you do not miss out on any features (such as search filtering).
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 auto-merge 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 Auto-Merge Location In A TOC


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 auto-merge location in a TOC

1. Open the TOC 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 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 **WebHelp Plus Auto-merge field**, select one of the following:
 - **Before** The auto-merge will occur immediately before the selected TOC entry.
 - **After** The auto-merge will occur immediately after the selected TOC entry.
 - **First child** The auto-merge 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 auto-merge 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 auto-merged.



- **Replace** The auto-merge 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 auto-merge occurs.
6. In the Properties dialog, click **OK**.
 7. Press **CTRL+S** or click  to save your work.

Specifying The Auto-Merge Location In A Browse Sequence


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 auto-merge 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 **WebHelp Plus Auto-merge field**, select one of the following:
 - **Before** The auto-merge will occur immediately before the selected browse sequence entry.
 - **After** The auto-merge will occur immediately after the selected browse sequence entry.
 - **First child** The auto-merge 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 auto-merge 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 auto-merge 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 auto-merge occurs.
6. In the Properties dialog, click **OK**.
 7. Press **CTRL+S** or click  to save your work.

Specifying The Order Of Auto-Merged Outputs

When you auto-merge 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 About Advanced Server Features in WebHelp Plus and "Merging Output at Runtime Using WebHelp Plus" on page 31.

EXAMPLE

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 auto-merge 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 auto-merged 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 auto-merge.

CHAPTER 5 Including Non-XHTML Files In Search

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 only 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 45.

How to include non-XHTML files in search

1. **Enable WebHelp Plus output** In order 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), (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) installing ASP.NET, (5) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (6) enabling WebHelp Plus search. See "Enabling WebHelp Plus Output" on page 11.
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.

EXAMPLE

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.

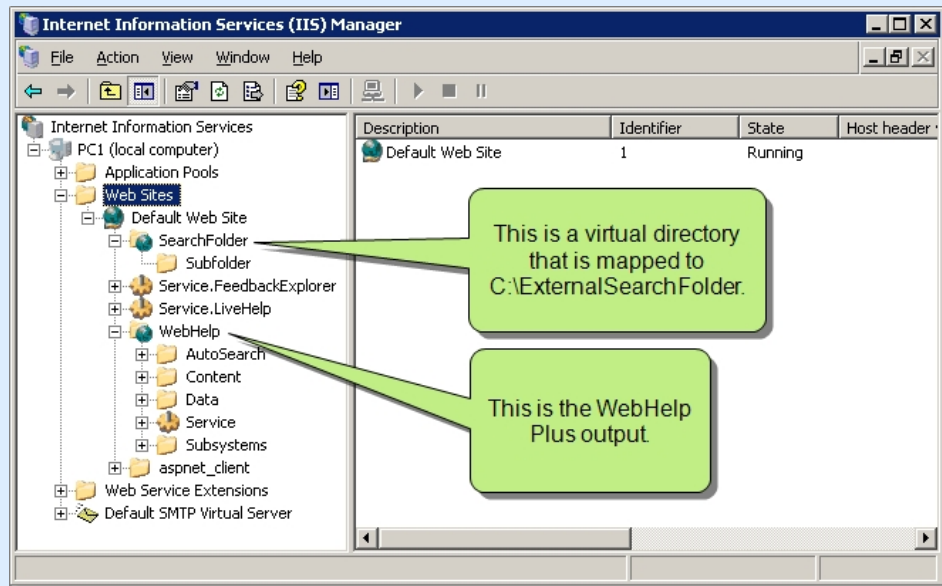
EXAMPLE

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.

CHAPTER 6 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 11 and "Viewing WebHelp Plus Output" on page 25.

SCENARIO

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).
See "Installing IIS" on page 12.
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 WebHelp Plus Target" on page 15.
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 installing ASP.NET and starting Microsoft Indexing Service.

See "Installing ASP.NET" on page 18 and "Starting Microsoft Indexing Service" on page 20.

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 `ConfigureISForWebHelpPlus.exe`, which is located in the output's `Service\Console.ConfigureISForWebHelpPlus` subfolder. In the case of this author, she finds her executable file here:

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

See "Enabling WebHelp Plus Search" on page 22.

6. **Author** The author generates her four "child" Flare targets (those that will be merged with the parent Jackpot target). In Flare, she selects **Build>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.

When you auto-merge 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. You can also choose specific places in the table of contents or browse sequence where child projects are merged. See "Specifying the Order of Auto-Merged Outputs" on page 38, "Specifying the Auto-Merge Location in a TOC" on page 34, and "Specifying the Auto-Merge Location in a Browse Sequence" on page 36. 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 41.

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 25.

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 **Build>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).
See "Installing IIS" on page 12.
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 17.
14. **Web Administrator** On the server where the output files will be published, the company's Web administrator completes the steps for installing ASP.NET.
See "Installing ASP.NET" on page 18.
15. **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 20.
16. **Web Administrator** The Web administrator opens the output folder on the server to enable WebHelp Plus search. He double-clicks the file ConfigureISForWebHelpPlus.exe, which is located here:
http://www.xyzincorporated.com/help_systems/Jackpot/Service/Console.ConfigureISForWebHelpPlus
See "Enabling WebHelp Plus Search" on page 22.
17. **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 41.
18. **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
19. **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.

CHAPTER 7 FAQs—WebHelp Plus

Following are some frequently asked questions about WebHelp Plus.

Does WebHelp Plus support 32- and 64-bit IIS?

Yes.

What does ConfigureISForWebHelpPlus.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 ConfigureISForWebHelpPlus.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 ConfigureISForWebHelpPlus.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.ConfigureISForWebHelpPlus" subfolder.

Run the ConfigureISForWebHelpPlus.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 ConfigureISForWebHelpPlus.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 ConfigureISForWebHelpPlus.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 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\v2.0.50727\aspnet_regiis -i
```

64-bit:

```
C:\WINDOWS\Microsoft.NET\Framework64\v2.0.50727\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 v2.050727**.
 - 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.

EXAMPLE

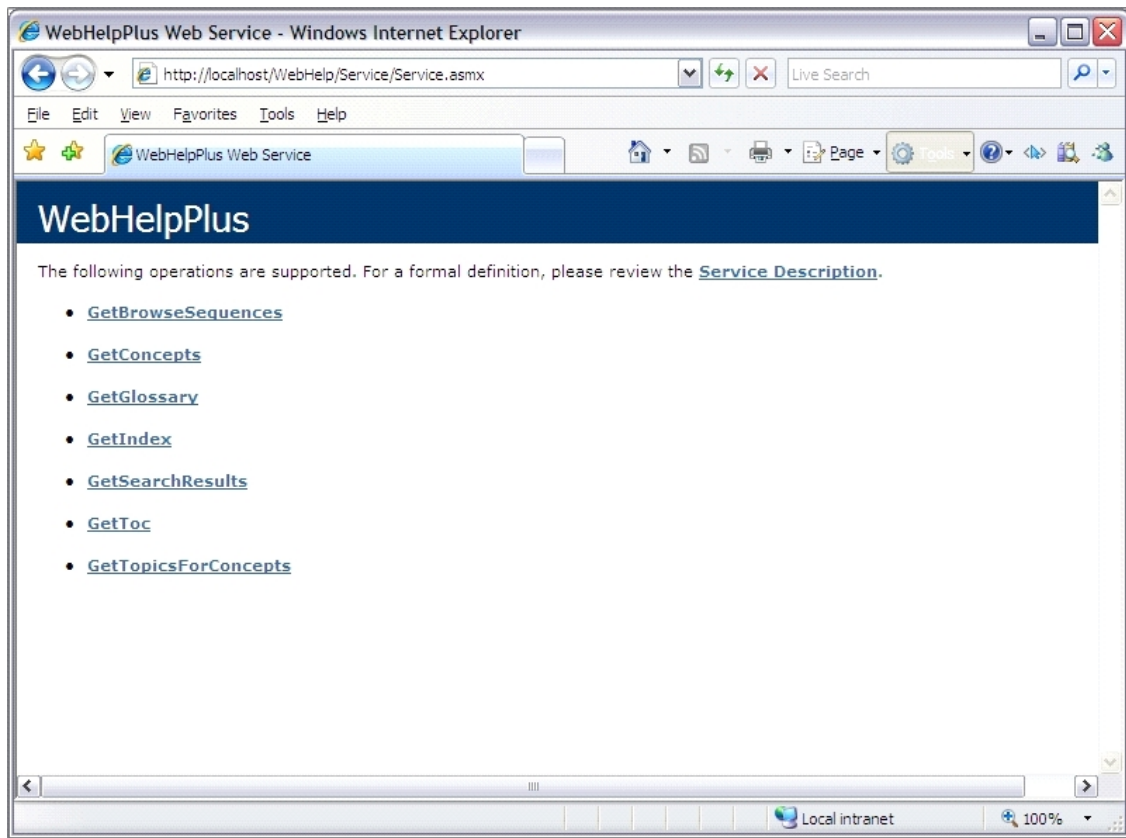
Let's say your output resides at:

`http://localhost/MyWebHelpPlus`

In that case, navigate to:

`http://localhost/MyWebHelpPlus/Service/Service.aspx`

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 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.

EXAMPLE

Let's say your output resides at:

`http://localhost/MyWebHelpPlus`

In that case, navigate to:

`http://localhost/MyWebHelpPlus/Service/Service.aspx`

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.

Index

A

AIR file 3
 ASP.NET 1, 9, 11, 32, 41, 53
 Auto-merging
 browse sequences 36
 order of outputs 38
 TOCs 34

B

Browse sequences
 auto-merge 36
 WebHelp Plus 52

C

Catalog 15
 CHM *See HTML Help*

D

DotNet Help 3

F

FAQs
 WebHelp Plus 9, 51
 Favorites
 WebHelp Plus 52
 Feedback 3

G

Glossaries
 WebHelp Plus 52

H

HTM file 1, 3

HTML Help 3
 Viewer 3

I

IFilter 41
 Indexes
 WebHelp Plus 52
 Indexing services *See Microsoft Indexing*
 Internet Information Services 1, 9, 11, 32, 41, 51

M

MadCap Help Viewer 3
 MCHelp file 3
 Merging
 projects 9, 11, 31-32, 34, 36, 38, 41
 WebHelp Plus 9, 11, 31-32, 34, 36, 38, 41
 Microsoft HTML Help *See HTML Help*
 Microsoft Indexing 1, 9, 11, 32, 41
 Microsoft Windows 7 1-2
 Microsoft Windows Server 2003 1-2, 29, 53
 Microsoft Windows Server 2008 1-2
 Microsoft Windows Vista 1-2, 16, 26-27, 33, 44
 Microsoft Windows XP 1-2, 15, 29, 45

O

Outline TOC *See Tables of contents*
 Output
 main entry file 3
 WebHelp Plus 1, 9, 11, 25, 45

P

Projects
merging 9, 11, 31-32, 34, 36, 38, 41

S

Search
IFilter 41
non-XHTML files 9, 11, 32, 41
WebHelp Plus 9, 11, 32, 41, 55
Server
WebHelp Plus 1

T

Tables of contents
auto-merge 34
WebHelp Plus 52-53
TOCs *See Tables of contents*

W

WebHelp 3
WebHelp AIR 3
WebHelp Mobile 3
WebHelp Plus 1, 3
browse sequences 52
catalog 15
enabling 9, 11, 45
FAQs 9, 51
favorites 52
glossaries 52
IFilter 41
indexes 52
merging 9, 11, 31-32, 34, 36, 38, 41
search 9, 11, 32, 41, 55
tables of contents 52-53
viewing 25, 45
Windows Search 1