MadCap Software

DotNet Help 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

CONTENTS

CHAPTER 1 About DotNet Help Output 5

CHAPTER 2 Merging Output from DNH Projects 7

CHAPTER 3 CSH Calls for DotNet Help 15

- Available DotNet Help Features in Flare 16
- What You Need to Do 19
- CSH Calls for DotNet Help—Developers 20
- HelpViewerClient API 23
- HelpViewerEmbeddedClient API 25
- IEmbeddedHelpSystem API 27
- ICSHIDProvider API 30
- Command Line 32

APPENDIX A PDF Guides 35

APPENDIX B Output Type Comparison Tables 37

- General 38
- Accessibility 40
- Context-Sensitive Help 41
- Feedback Statistics and Reporting Features 42
- Generated Content 43
- Glossaries 45
- Horizontal Rules 46
- Images 47
- Indexes 48
- Language Support 49
- Master Pages and Page Layouts 51
- Merging Output 52
- Miscellaneous Features 53

Multimedia 55
Navigation Links 56
Pulse (Socially Enabled Output) 57
Search 58
Skins 65
Tables of Contents and Mini-TOCs 70
Target Settings 72
Variables 75

CHAPTER 1

About DotNet Help Output

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

The output consists of a collection of files that you will distribute to users with the freely downloadable MadCap Help Viewer. The main entry file has an .mchelp extension.

The online Help attached to the Flare application was created with DotNet Help.

Following are some of the key features of the DotNet Help format:

- **.NET Application Support**
- **Accessible** This is a good choice if you do not want users to be burdened by the security warnings and limitations that are often encountered with Microsoft HTML Help and WebHelp.
- **Customizable/Modern Interface** You can produce a customizable interface that is much more modern looking than the aging Microsoft HTML Help. It will blend easily into a modern environment.
- **Embedded Context-Sensitive Help (CSH)** You can create embedded CSH in a .NET application. This includes the ability to produce Dynamic Help.
- **Multiple Languages** Users can select between multiple languages for the interface when viewing your output. This is possible because the freely distributable MadCap Help Viewer lets users select English, French, German, or Japanese for the interface.
- **Boolean Operator and Asterisk (*) Wildcard Character Searches** These are supported on Microsoft Windows XP (and newer operating systems) when SQL Compact is installed on the local computer building the Help.

CHAPTER 2

Merging Output from DNH Projects

Use this merging method if you are developing DotNet Help and you want to merge the output files from your parent project with the DotNet Help output files from an external project. This method is useful, for example, if another author is working on the external DotNet Help project to which you are linking and you only have access to the output files (not the project files).

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.

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

Let's say the master project's output files are being published to the following:

C:\MyHelpSystem\Master\Master.mchelp

In that case, the output folders and files for the child project should be copied to the following folder:

C:\MyHelpSystem\Master\Subsystems

For example: 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.mchelp" is the main output file for the child project that was created. Not only does the main .mchelp output file need to be included in this folder, but also the ancillary output folders and files (e.g., Content, Data).

You can merge DotNet Help outputs using an automatic or manual method.

HOW TO MERGE OUTPUT FROM DOTNET HELP PROJECTS (AUTOMATIC METHOD)

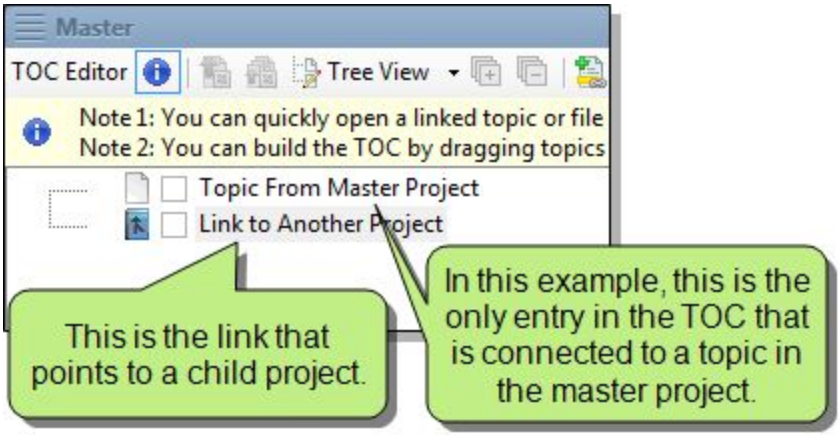
1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.

   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.

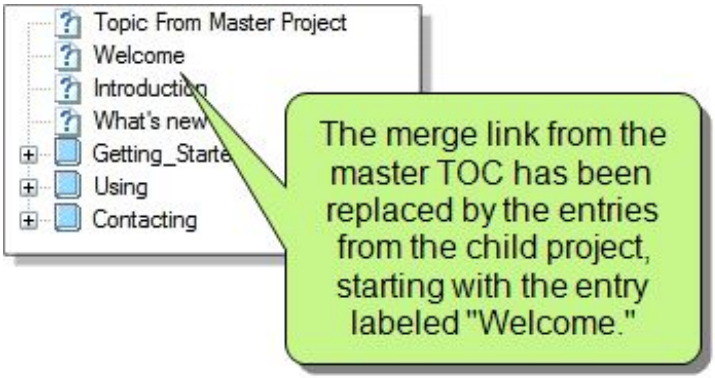
3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child DotNet Help project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External 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 DotNet Help Files (*.mchelp)**. Then find and select the MCHELP 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**.

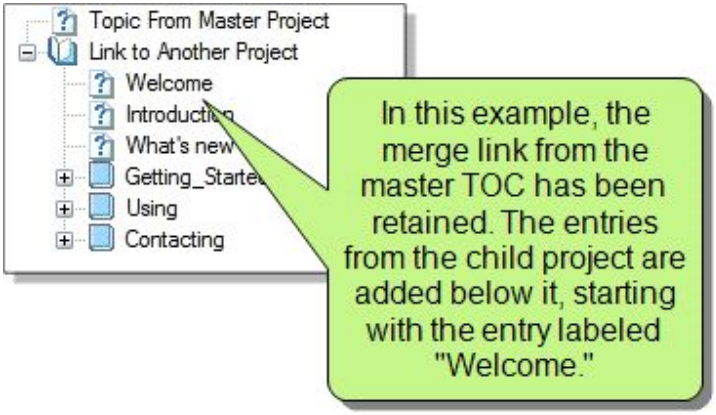
For example, you might have a TOC in your master project that looks like this:



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:



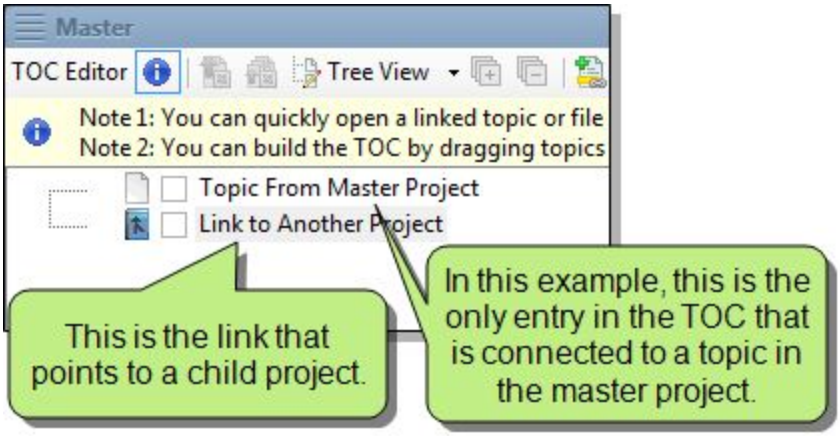
10. In the Properties dialog, click **OK**.
11. Build the output for the DotNet Help target in your "master" project.

    Flare automatically creates the Subsystems folder for you. It also copies the target folder and .mchelp 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.

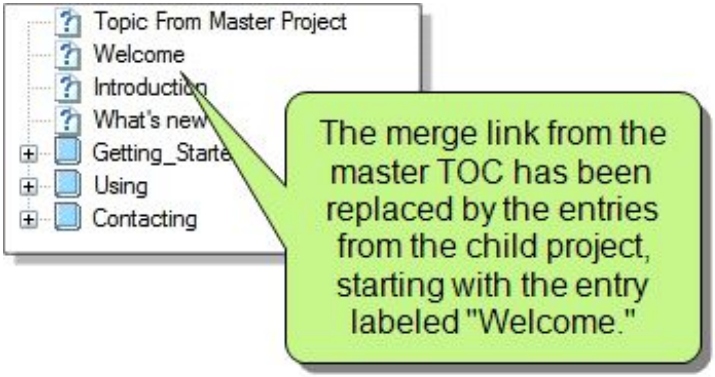
HOW TO MERGE OUTPUT FROM DOTNET HELP PROJECTS (MANUAL METHOD)

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 DotNet Help 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., AnotherHelpSystem/AnotherHelpSystem.mchelp).
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**.

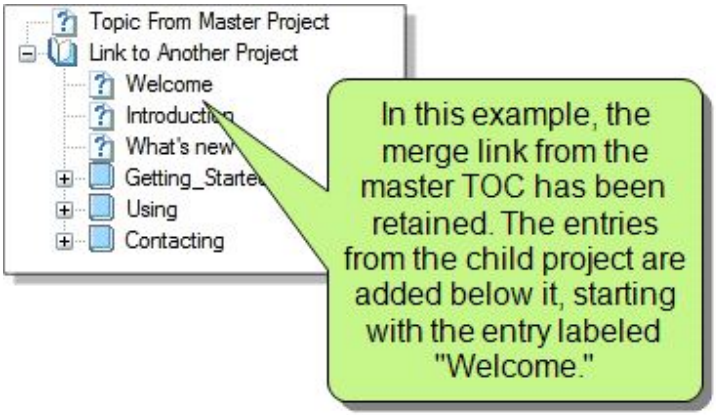
   For example, you might have a TOC in your master project that looks like this:



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 DotNet Help 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: 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 3

CSH Calls for DotNet Help

There are several ways that your Flare DotNet Help system can be "connected" to a software application.

This chapter discusses the following:

Available DotNet Help Features in Flare 16
What You Need to Do 19
CSH Calls for DotNet Help—Developers 20
HelpViewerClient API 23
HelpViewerEmbeddedClient API 25
IEmbeddedHelpSystem API 27
ICSHIDProvider API 30
Command Line 32

Available DotNet Help Features in Flare

Features that are available include the following.

- **Detached or Integrated Help** The Help system can be connected so that the Help opens in a window separate from your software application (detached Help).

  E X A M P L E

  To see an example of detached Help in Flare's online Help, select **Tools>Manage Templates** and then press **F1**. The topic for that dialog opens in a separate window.

  Alternatively, the Help can be connected so that the Help opens within the software application (integrated Help).

  E X A M P L E

  To see an example of integrated Help in Flare's online Help, select **Help>Contents**. The TOC for the Help opens within the application. If you click a topic page in the TOC, the topic opens in the application to the right.

- **Same or Different Process** The Help system can be connected so that it uses the same process as the software application (embedded). If the Help uses the same process, the Help will close when the application is closed. The online Help in Flare is connected so that it uses the same process as Flare. Alternatively, the Help system can be connected so that it uses a different process than the software application. If the Help uses a different process, the Help can remain open even if the application is closed.

» **Basic Help or Context-Sensitive Help (CSH)** The developer can connect the application to your basic DotNet Help output, rather than to a specific topic. For example, you might want a standalone version of the Help Viewer to open separately from the software application, displaying your startup topic and showing panes for the navigation elements (e.g., TOC, Index, Search, Browse Sequences) that you have created. Or maybe you want users to be able to open the different navigation elements within the application and access topics from each one.

E X A M P L E

To see an example of this in Flare's online Help, select **Help>Contents**. The TOC for the Help opens within the application. If you click a topic page in the TOC, the topic opens in the application to the right.

Alternatively, the developer can use CSH to connect the application to specific topics in the DotNet Help output (as long as you have created CSH in your Flare project and share the header file information with the developer).

E X A M P L E

To see an example of CSH in Flare's online Help, select **Tools>Manage Templates** and then press **F1**. The topic for that specific dialog opens.

» **Dynamic Help** The developer can also incorporate your Help into the application using a unique feature called "Dynamic Help." This is a type of CSH where a Help window automatically displays topics from a Help system as an individual uses the application. It does not require any other action from the user (e.g., clicking a button or pressing a shortcut key). The Help system simply follows the actions of the user, automatically providing the appropriate Help content based on the area of the interface that is being clicked.

E X A M P L E

To see an example of Dynamic Help in Flare's online Help, select **Help>Dynamic Help**. Then start clicking anywhere in the Flare interface. As you do so, the Help topic changes accordingly.

» **F1 Help** The developer can connect the application to the Help in such a way that the Help opens when a user presses the F1 key.

> **E X A M P L E**
>
> To see an example of this in Flare's online Help, simply click in any area of the interface or open any dialog. Then press **F1**. The appropriate Help topic for that area of the interface displays.

» **Dialog's Help Button** The developer can connect the application to the Help in such a way that the Help opens when a user clicks the dialog's Help button (i.e., the "question mark" button).

> **E X A M P L E**
>
> To see an example of this in Flare's online Help, select **Tools>Manage Templates**. Then click the **?** button in the upper-right corner of the dialog.

What You Need to Do

1. Work with your developer to determine which of the above features you want to use. You may want to use several of the features, as is the case with the Flare's online Help.
2. Create and build your DotNet Help project in Flare.
3. Provide the developer with the output files, as well as the CSH header file (if you plan to use CSH).
4. Provide the developer with the information in the following topics. The developer should begin with the "CSH Calls for DotNet Help—Developers" topic.
   - "CSH Calls for DotNet Help—Developers" on the next page
   - "HelpViewerClient API" on page 23
   - "HelpViewerEmbeddedClient API" on page 25
   - "IEmbeddedHelpSystem API" on page 27
   - "ICSHIDProvider API" on page 30
   - "Command Line" on page 32

CSH Calls for DotNet Help—Developers

Information for Developers

There are three ways to make DotNet Help calls from your Windows application. The most basic way is to use the command line functionality of the DotNet Help Viewer (see "Command Line" on page 32). The other two ways are to use the HelpViewerClient (HVC) (see "HelpViewerClient API" on page 23) and HelpViewerEmbeddedClient (HVEC) (see "HelpViewerEmbeddedClient API" on page 25) classes. Use the command line version only if your application doesn't have access to the latest .NET framework. Otherwise, you should use one of the other two methods.

Features of the DotNet Help CSH Engine

- Open a topic of your choice in the MadCap Help Viewer.
- Launch a Help system with the search results from a search string that you specify.
- Embed Help topics or search results directly into your application.
- Dynamic Help—A Help window automatically shows relevant information based on the user's focus in your application.

Key Differences between the HVC and HVEC Classes

- HVC only supports basic, non-embedded CSH calls. It also does not provide Dynamic Help functionality.
- HVEC contains built-in functionality for F1 help. F1 help is still possible with HVC, however the developer must manually handle the user pressing the F1 key.
- HVC opens the Help Viewer in a separate process from your application, whereas HVEC opens the Help Viewer in the same process as your application. This means that with HVC, if your application closes, the Help window will remain open. On the other hand, with HVEC, if your application closes, the Help window will also close.

Using the API

You can download the Help Viewer SDK at:

http://www.madcapsoftware.com/downloads/redistributables.aspx

You must add a reference to the appropriate DLL in your project to use the API.

In Visual Studio 2005, do the following.

1. In the Solution Explorer, right-click the project.
2. Click **Add Reference**.
3. Select the **Browse** tab.
4. Navigate to the appropriate .dll file and select it:
   - Select **MadCap.HelpViewerClient.dll** to use the HelpViewerClient API.

     OR
   - Select the **MadCap.HelpViewerEmbeddedClient.dll** to use the HelpViewerEmbeddClient API.
5. Click **OK**.

Sample Applications

There are two sample applications available in the SDK that demonstrate using the HelpViewerClient and HelpViewerEmbeddedClient classes.

Distribution

Help Viewer must be installed on the user's machine that is using your application. The SDK contains a merge module that you can incorporate into an MSI-based installer to distribute the Help Viewer.

The following are instructions on how to include the merge module using Visual Studio 2005.

1. In the Solution Explorer, right-click the MSI installer project.
2. Click **Add**.
3. Click **Merge Module**.
4. Browse to the **InstallMadCapHelpViewer.msm** file and select it.
5. Click **Open**.

Search Performance and Wildcards

When SQL Server Compact 3.5 SP1 is present, Help Viewer can use it to improve search performance and also allow wildcard searching. To take advantage of these capabilities, you can include SQL Server Compact 3.5 SP1 as part of your application's installation. For more information regarding SQL Server Compact 3.5 SP1 including redistribution rights, please visit:

http://www.microsoft.com/Sqlserver/2008/en/us/compact.aspx

HelpViewerClient API

Information for Developers

The HelpViewerClient API is used to make DotNet Help calls, including context-sensitive help (CSH) calls. HelpViewerClient only supports basic, non-embedded calls. It also does not provide Dynamic Help functionality. The HelpViewerClient API will open the Help Viewer in a separate process from your application.

If instead you require integrated Help, Dynamic Help, or opening the Help Viewer in the same process as your application, see "HelpViewerEmbeddedClient API" on page 25.

HOW TO USE THE HELP VIEWER CLIENT API

1. Create an instance of the class using the constructor.
2. Call **Load()** or **TryLoad()** to load the Help system of your choice.
3. Call **Search()** or **ShowTopic()**. The first time either of these two methods are called, a new Help Viewer window will be launched. Every call after that will open the desired topic or search results in the existing Help Viewer window.

Name	Description
`public void HelpViewerClient ()`	Creates an instance of the HelpViewerClient.
`public void Load ( string helpsystem )`	Loads the Help system at the path specified in the **helpsystem** parameter. This method must be called once before calling **Search()** or **ShowTopic()**. If the Help system does not exist, an **ArgumentException** is thrown. See also **TryLoad()**.
`public void Search ( string searchString )`	Performs a search for the string specified in the **searchString** parameter.
`public void Search ( string searchString, string cshid )`	Performs a search for the string specified in the **searchString** parameter. The **cshid** parameter is used to override the default starting topic of the Help system. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system.

Name	Description
`public void ShowTopic ( string cshid )`	Opens the Help system with the topic specified in **cshid**. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system.
`public bool TryLoad( string helpsystem )`	See **Load()**. This method is equivalent to the Load() method. However, instead of throwing an exception, it returns true if the Help system was loaded successfully; it returns false otherwise.

EXAMPLES

Refer to the sample application called "HelpViewerClientTester" for a fully functional application that uses the HelpViewerClient API. This sample application is available in the SDK, which you can download from:

http://www.madcapsoftware.com/downloads/redistributables.aspx

```
  HelpViewerClient client = new HelpViewerClient();

  if ( client.TryLoad( @"C:\My Project\Manual.mchelp" ) )

{

  client.ShowTopic( "MyID1" );

}
```

HelpViewerEmbeddedClient API

Information for Developers

The HelpViewerEmbeddedClient API is used to make calls to the DotNet Help system, including context-sensitive help (CSH) calls. This API supports embedded CSH calls and can also provide Dynamic Help functionality. When you use this API, the MadCap Help Viewer opens in the same process as your application. If instead you require detached Help, opening the Help Viewer in a separate process from your application, see "HelpViewerClient API" on page 23.

HOW TO USE THE HELP VIEWER EMBEDDED CLIENT API

1. Create an instance of the class using the constructor.
2. Call **Load()** or **TryLoad()** to load the Help system of your choice.
3. Use the functionality of the Help System object. For more information see "IEmbeddedHelpSystem API" on page 27.

Name	Description
`public void HelpViewerEmbeddedClient()`	Creates an instance of the HelpViewerEmbeddedClient.
`public void HelpViewerEmbeddedClient( string title )`	Creates an instance of the HelpViewerEmbeddedClient. Sets the title of the Help system to the string specified in **title**.
`public void HelpViewerEmbeddedClient( string title, Icon icon )`	Creates an instance of the HelpViewerEmbeddedClient. Sets the title of the Help system to the string specified in **title**. Sets the icon of the help system to the icon specified in **icon**.
`public void Load( string helpsystem )`	Loads the Help system at the path specified in the **helpsystem** parameter. This method must be called once before calling **Search()** or **ShowTopic()**. If the Help system does not exist, an **ArgumentException** is thrown. See also **TryLoad()**.
`public bool TryLoad ( string helpsystem )`	See **Load()**. This method is equivalent to the Load() method. However, instead of throwing an exception, it returns true if the Help system was loaded successfully; it returns false otherwise.
`public IEmbeddedHelpSystem HelpSystem`	Gets an interface to the Help system. See "IEmbeddedHelpSystem API" on page 27 for more information.

Name	Description
`public virtual string Title`	Gets the title of the Help system.

EXAMPLES

Refer to the sample application called "HelpViewerEmbeddedClientTester" for a fully functional application that uses the HelpViewerEmbeddedClient API. This sample application is available in the SDK, which you can download from:

http://www.madcapsoftware.com/downloads/redistributables.aspx

```
  // Create a group box where the embedded help topic will be displayed

  GroupBox mTopicPanel = new GroupBox();

  mTopicPanel.Location = new System.Drawing.Point( 32, 204 );

  mTopicPanel.Name = "mTopicPanel";

  mTopicPanel.Text = "Help Topic";

  // Display a topic embedded in the group box

  HelpViewerEmbeddedClient client = new HelpViewerEmbeddedClient();

  if ( client.TryLoad( @"C:\My Project\Manual.mchelp" ) )

{

  client.HelpSystem.DynamicHelpPanel = mTopicPanel;

  client.HelpSystem.LoadTopic( "MyID1" );

}
```

IEmbeddedHelpSystem API

Information for Developers

The IEmbeddedHelpSystem API is an interface that exposes functionality for context-sensitive Help (CSH), embedded Help, and Dynamic Help.

Detached CSH

To launch the Help system in a separate Help Viewer window, use the **ShowHelp()** method.

To make a CSH call and have the results displayed in a separate Help Viewer window, use the **ShowTopic()** method.

To display search results from a Help system in a separate Help Viewer window, use the **Search()** method.

These methods will launch an instance of the Help Viewer application. It will be launched in the same process as your application so when the user closes your application, the Help Viewer will also be closed.

Embedded CSH

To embed a topic from your Help system into your application, use the **LoadTopic()** method.

To embed search results from your Help system into your application, use the **EmbeddedSearch()** method.

You must set the **DynamicHelpPanel** property prior to calling **LoadTopic()**. Similarly, you must set the **SearchPanel** property prior to calling **EmbeddedSearch()**.

E X A M P L E S

Refer to the sample application called "HelpViewerEmbeddedClientTester." This sample application is available in the SDK, which you can download from:

http://www.madcapsoftware.com/downloads/utilities/redistributables.aspx

Dynamic Help

Dynamic Help is a feature where a Help window automatically displays topics from a Help system that are relevant to the context as an individual uses an application. For example, if the user clicks on an area of your UI called "Project Organizer," the topic pertinent to the Project Organizer opens. You can see an example of Dynamic Help in Flare's online Help (select **Help>Dynamic Help**).

To use this functionality, you must set the **EnableDynamicHelp** property to true. The CSH engine needs a way to determine the appropriate topic to display. Your application can supply this information in either of two ways.

- One way is to implement the ICSHIDProvider interface. For more information see "ICSHIDProvider API" on page 30.
- The other way is to set the control's **Tag** property to the desired CSH ID string. In this case, you must prefix the string with "CSH:"

  For example, if your CSH ID is "MyHelpTopic," you would set the Tag object to the string "CSH:MyHelpTopic."

Name	Description
`public void EmbeddedSearch ( string searchString )`	Performs a search for the string specified in the **searchString** parameter. Displays the results embedded in the control that is set as the **SearchPanel** property.
`public void EnableHelpButton ( System.Windows.Forms.Form form )`	Enables the Help button in the form specified in **form**. See "ICSHIDProvider API" on page 30 if you are enabling this feature.
`public void LoadTopic( string cshid)`	Loads the topic specified in **cshid**. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system. Displays the topic embedded in the control that is set as the **DynamicHelpPanel** property.
`public void Search( string searchString )`	Performs a search for the string specified in the **searchString** parameter. Displays the results in a new Help Viewer window.
`public void Search( string searchString, string cshid )`	Performs a search for the string specified in the **searchString** parameter. The **cshid** parameter is used to override the default starting topic of the Help system. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system. Displays the results in a new Help Viewer window.
`public void ShowHelp()`	Opens the Help system in a new Help Viewer window.

Name	Description
`public void ShowTopic ( string cshid )`	Opens the Help system with the topic specified in **cshid**. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system.
`System.Windows.Forms.Control DynamicHelpPanel`	Gets or sets the control where the embedded Help topic will be loaded. This must be set before calling **LoadTopic()**.
`bool EnableDynamicHelp`	Gets or sets whether Dynamic Help is enabled for the form. See "ICSHIDProvider API" on the next page if you are enabling this feature.
`bool EnableF1Help`	Gets or sets whether F1 Help is enabled for the form. See "ICSHIDProvider API" on the next page if you are enabling this feature.
`System.Windows.Forms.Control SearchPanel`	Gets or sets the control where the embedded Help search results will be loaded. This must be set before calling **EmbeddedSearch()**.

ICSHIDProvider API

Information for Developers

The ICSHIDProvider API is used to enable a System.Windows.Forms.Control to be a context-sensitive Help (CSH) ID provider. This is required to incorporate any of the following.

- **Dynamic Help** This is a feature where a Help window automatically displays topics from a Help system that are relevant to the context as an individual uses an application. For example, if the user clicks on an area of your UI called "Project Organizer," the topic pertinent to the Project Organizer opens. You can see an example of Dynamic Help in Flare's online Help (select **Help>Dynamic Help**).
- **F1 Help** This is a feature where the Help opens when a user presses the F1 key.
- **Dialog's Help Button** This is a feature where the Help opens when a user clicks the dialog's Help button (i.e., the "question mark" button).

If your application uses any of these features, the DotNet CSH engine uses this interface to determine the desired topic to display. As an alternative to implementing this interface, you may also set the control's **Tag** property to the desired CSH ID string. In this case, you must prefix the string with "CSH:"

E X A M P L E

For example, if your CSH ID is "MyHelpTopic," you would set the Tag object to the string "CSH:MyHelpTopic"

Name	Description
`public string HelpSystemCSHID`	Gets the CSH ID of the topic that should be displayed in the Help system.

The following shows how to implement the ICSHIDProvider interface.

```
public class MyTextBox : TextBox, ICSHIDProvider
  {
  private string mHelpSystemCSHID = "TestID1";
  public string HelpSystemCSHID
  {
  get { return mHelpSystemCSHID; } }
  ...
  }
```

The following shows how to set the Tag object.

```
TextBox myTextBox = new TextBox();
myTextBox.Tag = "CSH:TestID1";
```

Command Line

Information for Developers

Following is an explanation of how to use the command line options of the MadCap Help Viewer application. Use this method only if your application does not have access to the latest .NET framework. Otherwise, use either the HelpViewerClient API or HelpViewerEmbeddedClient API. See "HelpViewerClient API" on page 23 and "HelpViewerEmbeddedClient API" on page 25.

Command Line Syntax

```
HelpViewer.exe -file <path to help system> [-cshid <identifier>] [-search <search_
string>]
```

Parameter	Description
file	Path to the DotNet Help output file (.mchelp extension)
cshid	CSH ID of the desired topic to display. This can be either the identifier name or value. Alternatively, the ID may contain a topic path. When using a topic path, it must be relative to the Content folder of the Help system.
search	Search string that the Help system will automatically search for when it is opened

The following opens the specified Help system.

```
>HelpViewer.exe -file "C:\My Help System\Manual.mchelp"
```

The following opens the specified Help system and opens the topic associated with "MyID."

```
>HelpViewer.exe -file "C:\My Help System\Manual.mchelp" -cshid MyID
```

The following opens the specified Help system and opens the topic located at MyTopic.htm.

```
>HelpViewer.exe -file "C:\My Help System\Manual.mchelp" -cshid MyTopic.htm
```

The following opens the specified Help system and displays search results for "bikes and trikes."

```
>HelpViewer.exe -file "C:\My Help System\Manual.mchelp" -search "bikes and trikes"
```

The following opens the specified Help system, opens the topic associated with "MyID," and displays search results for "bikes and trikes."

```
>HelpViewer.exe -file "C:\My Help System\Manual.mchelp" -cshid MyID -search "bikes and
trikes"
```

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 Bulleted)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
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 5	AIR	WebHelp Mobile	WebHelp Plus	HTML 5
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 Standalone 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 5	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	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓