



DoctoHelp® 2006

Copyright © 1987-2007 ComponentOne LLC. All rights reserved.

Corporate Headquarters
ComponentOne LLC
201 South Highland Avenue
3rd Floor
Pittsburgh, PA 15206 USA

Internet: info@ComponentOne.com
Web site: <http://www.componentone.com>

Sales

E-mail: sales@componentone.com
Telephone: 1.800.858.2739 or 1.412.681.4343 (Pittsburgh, PA USA Office)

Technical Support

See *Technical Support* in this manual for information on obtaining technical support.

Trademarks

ComponentOne Doc-To-Help 2006 and the ComponentOne Doc-To-Help 2006 logo are trademarks, and ComponentOne is a registered trademark of ComponentOne LLC. All other trademarks used herein are the properties of their respective owners.

Warranty

ComponentOne warrants that the original CD (or diskettes) are free from defects in material and workmanship, assuming normal use, for a period of 90 days from the date of purchase. If a defect occurs during this time, you may return the defective CD (or disk) to ComponentOne, along with a dated proof of purchase, and ComponentOne will replace it at no charge. After 90 days, you can obtain a replacement for a defective CD (or disk) by sending it and a check for \$25 (to cover postage and handling) to ComponentOne.

Except for the express warranty of the original CD (or disks) set forth here, ComponentOne makes no other warranties, express or implied. Every attempt has been made to ensure that the information contained in this manual is correct as of the time it was written. We are not responsible for any errors or omissions. ComponentOne's liability is limited to the amount you paid for the product. ComponentOne is not liable for any special, consequential, or other damages for any reason.

Copying and Distribution

While you are welcome to make backup copies of the software for your own use and protection, you are not permitted to make copies for the use of anyone else. We put a lot of time and effort into creating this product, and we appreciate your support in seeing that it is used by licensed users only. Please read the *END-USER LICENSE AGREEMENT FOR COMPONENTONE SOFTWARE* and *License and Redistributable Files* sections in this manual before copying and redistributing any ComponentOne Doc-To-Help 2006 files.



Table of Contents

Table of Contents	iii
ComponentOne Doc-To-Help 2006	1
About ComponentOne Doc-To-Help 2006.....	1
What's Available in Doc-To-Help.....	1
What's New in Doc-To-Help.....	3
END-USER LICENSE AGREEMENT FOR COMPONENTONE DOC-TO-HELP ...	7
License and Redistributable Files.....	12
Technical Support	12
System Requirements	13
Using Anti-virus Software with Doc-To-Help.....	13
Installation.....	13
Authoring Templates for Microsoft Word	13
Global Templates.....	15
Doc-To-Help Cascading Style Sheets	15
Help Files	15
Microsoft Help and HTML Help Workshop	16
About This Document.....	16
Help Authoring Basics	19
What is a Help System?.....	19
Building Help Systems with Doc-To-Help	22
Help Targets in Doc-To-Help	27
The Doc-To-Help Project Editor	41
Navigating the Project Editor.....	41
The Project Icon.....	42
The Views Icon	46
The Topics Icon	46
The Contents Icon.....	48
The Index Icon	48
The Build Icon	49
A Guided Tour of Doc-To-Help	51
Converting Projects to Doc-To-Help 2006.....	51
Using Microsoft Word	61
Using HTML Source Documents	92
Using the Doc-To-Help Project.....	116
Working with Projects	129
The Doc-To-Help Start Page.....	129
Using Live Update	161
Compacting a Project File	162
Project Settings Properties	163
Changing the Current Help Target	164
Building the Current Help Target.....	164
Viewing the Current Help Target	166
Adding a New Help Target.....	166
Renaming a Help Target.....	167
Changing the Help Target Directory	167
Help Target Properties.....	168
Doc-To-Help 2006 Source Documents	181
Project Types	181
Document Tree	183
Single and Multiple Topic Documents	183

Word Source Documents.....	185
HTML Source Documents.....	187
Organizing Your Project Files.....	190
Adding a Document to a Project.....	191
Removing a File From a Project.....	196
Compiling Source Documents.....	196
Doc-To-Help Markup in Source Documents.....	197
Templates and Cascading Style Sheets.....	199
Adding Templates to a Project.....	199
Opening a Template from a Project.....	200
Using Cascading Style Sheets.....	201
Opening a Cascading Style Sheet From a Project.....	209
Using Styles in Doc-To-Help.....	211
Using Paragraph Styles.....	211
Using Character Styles.....	216
Importing Styles from a Word or HTML Document, Template, or Cascading Style Sheet.....	220
Importing Styles From Another Project.....	221
Using Topic Types.....	222
Working With Styles in Word.....	227
Using HTML Styles in a Doc-To-Help Project.....	230
Working with Cascading Style Sheets and Word Documents.....	231
Doc-To-Help Markup Language (D2HML).....	233
D2HML Styles.....	233
D2HML Formatting Rules.....	241
D2HML Quick Reference.....	246
Creating a Hot Spot.....	249
Hot Spot Types.....	252
Showing Hidden Hot Spots.....	275
Defining and Organizing Topics.....	279
Topic Properties.....	280
Making a Paragraph Style Active.....	281
Specifying the Help Contents.....	282
Specifying the Navigation Sequence.....	284
Providing Links to Subtopics.....	285
Providing Links to Specific Topics.....	285
Customizing the Related Topic Jumps.....	286
Labeling Subtopic Links.....	287
Providing Links to the Next Topic.....	288
Editing Topic Properties.....	289
Adding a Link Tag to a Topic.....	289
Modifying the Help Contents.....	291
Context Sensitive Help.....	297
Sorting Topics For Easy Viewing With Outlook Style Grouping.....	306
Filtering Topics in Doc-To-Help.....	308
Links and Hot Spots.....	309
Creating a Jump to a Topic in Another Document.....	309
Creating Popup Topics.....	310
Creating Keyword Links (KLinks).....	310
Creating Group Links (ALinks).....	311
Creating Mid-Topic Jumps Using D2HML.....	312
Expanding, Dropdown and Popup Text.....	313
Creating Active Character Styles.....	318
Alternative Options for Creating Links and Hot Spots in Word.....	319
Using Image Maps.....	329

Creating a Glossary	333
Specifying a Glossary Document	333
Opening the Glossary	334
Editing the Glossary in Doc-To-Help	334
Adding and Formatting Glossary Entries	334
Filtering Topics from the Glossary	335
Viewing Glossary Terms	336
Adding and Sorting Glossary Terms in Word	336
Creating Glossary Links	338
Building an Index	341
Project Editor Index View	341
Using the Index Toolbar	342
Adding Index Keywords Manually	343
Creating Index Keywords Automatically Using Styles	343
Adding Secondary Index Keywords Manually	344
Renaming an Index Keyword	345
Assigning Topics to an Index Keyword	345
Removing Unused Index Keywords	347
Adding an Index Keyword to a Topic	348
Creating Topic Groups	349
Renaming a Group	349
Assigning Topics to a Group	349
Adding a Group to a Topic	349
Removing Topics from Index Keywords or Topic Groups	350
Using the Index Elements Auto-Completion Feature	350
Customizing Help Windows	353
HTML Help Window Properties	353
NetHelp Window Properties	354
WinHelp 4.0 Window Properties	355
Creating a New Help Window	355
Assigning a Window to a Style	356
Editing Help Window Properties	356
Using Themes to Modify the Help Window	357
Modifying the Background of the Window Pane	359
Modifying the Window Size Using the Size Tool	362
Tips and Techniques for Word	365
Working with Graphics in Word	365
Inserting Standard Tables with Doc-To-Help	368
Linking to Internet Sites	369
Generating Project Reports	373
Managing Project Files	376
Deleting Help Topics	378
Using HTML Help Object Tags in Word	378
Using WinHelp Macros in Word	379
Doc-To-Help Shortcut Keys for Word	397
Conditional Text and Attributes	401
Specifying Conditional Text	401
Conditional Text Target Options	402
Working with Conditional Text in Word Documents	403
Working with Conditional Text in HTML Documents	416
Applying Multiple Conditions to Text	417
Applying Conditional Text to a Topic	419
Applying Conditional Text to a Document	421
Using Attributes	422

Using Modular Help	427
Managing Your Modular Help System	427
Creating a Modular Help Project for WinHelp.....	428
Creating a Modular Help Project for HTML Help	431
Creating a Modular Help Project for NetHelp.....	435
Scripting Techniques	439
What are Scripts?.....	439
Working with Scripts	441
Common Script Operations	444
DocuCenter for .NET	449
Generation Speed and Supported Word Versions.....	449
DocuCenter for .NET Guided Tour	449
DocuCenter Controls and Toolbars	470
Changing the Project Options	473
Examining the Automatically Generated Document and Project	473
Adding Narrative to a DocuCenter Project.....	475
Adding Formatting and Links to XML Comment Text	477
Adding Topic Links to a DocuCenter Project	479
Multilanguage Support in DocuCenter for .NET	481
Customization and Localization of DocuCenter for .NET	482
Formatter Versions in DocuCenter for .NET	484
Using DocuCenter Styles to Create Links.....	484
Doc-To-Help 2006 Features	495
Doc-To-Help Express	495
Team Authoring.....	514
Using Natural Search for Doc-To-Help	552
Using the Image Map Editor in Microsoft Word.....	560
The Modular TOC Utility.....	563
Using the Context String Editor	569
Using the Theme Designer.....	572
Using the Topic Tools in Word	604
Object Model Reference	611
Index	685

ComponentOne Doc-To-Help 2006

About ComponentOne Doc-To-Help 2006

Thank you for purchasing ComponentOne Doc-To-Help 2006! Doc-To-Help is a Single-Source Help Authoring Tool. It allows you to take one or a set of source documents and convert them to as many Help systems or printed manuals as you like.

Doc-To-Help 2006 is available in two versions: Doc-To-Help Enterprise 2006 and Doc-To-Help for Word 2006. Doc-To-Help Enterprise 2006 empowers you to author content in Microsoft Word, HTML or both. Doc-To-Help Word 2006 supports Word source documents only. Both versions allow you to create professional manuals and various online Help formats, including HTML Help, NetHelp for Web deployment, MS Help 2.0, JavaHelp 2.0 and WinHelp 4.0.

Whether you're creating an online Help system or a book, your goal is to communicate clearly with your audience and provide them with the information they need, when they need it. You do that by developing content on one or more subjects (*topics*), often pointing your audience to places where they can find additional information or images (*links*). You give them a hierarchical view, or outline, of the information they'll find (*contents*); a searchable list of words that correspond to one or more topics (*index*); and ways to access (*navigate*) the information.

Doc-To-Help helps you to do all of that by analyzing the styles, templates, and cascading style sheets you apply to Word and HTML documents.

Our easy-to-use [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) offers a wide variety of predefined styles and makes formatting your source documents as simple as clicking a button. You can define Help elements such as topic links, tables, conditional text, glossary terms, expanding text and much more.

Doc-To-Help also provides templates and cascading style sheets for your convenience, or you may choose to use your own.

If you have existing RoboHelp or other HTML Help projects, you can begin converting them to Doc-To-Help with one click. The resulting Doc-To-Help project contains clean source files and is immediately ready for output to the Help target of your choice.

Let Doc-To-Help free you from complex Help system mechanics so you can focus on content, not production. The upcoming sections will take you through the process step-by-step. In no time, you'll be up and running!

What's Available in Doc-To-Help

Doc-To-Help 2006 is available in two versions: Doc-To-Help Enterprise 2006 and Doc-To-Help for Word 2006. Both versions contain great features, including Doc-To-Help Markup Language, Documenter enhancements, NetHelp Output and enhancements, expanding text, drop-down text and text-only pop-ups and much more!

Doc-To-Help Enterprise 2006 allows you to use HTML source documents and cascading style sheets, and you can even import and convert existing RoboHelp and other HTML Help projects into Doc-To-Help. If you want the full functionality of Doc-To-Help, Enterprise 2006 is the right choice. If you do not plan to use HTML source documents in your help projects, Doc-To-Help for Word 2006 may be the better option. The following table compares the key features available in both versions.

Doc-To-Help 7.5, 2005 and 2006 Feature Comparison						
	7.5	7.5 Pro	for Word 2005	Enterprise 2005	for Word 2006	Enterprise 2006
New Doc-To-Help Features						
Team Authoring Support						★
PDF Output					★	★
Document Conditions					★	★
Favorites Tab in NetHelp					★	★
Email Topic in NetHelp					★	★
Print Topic in NetHelp					★	★
Convert RoboHelp Single Source Layouts					★	★
Merge Multiple NetHelp Projects					★	★
Live Links in Printed Manuals (for better PDF conversion)					★	★
Support for Images Maps in D2HML					★	★
Generating Links to .NET Framework Types in Help 2.0					★	★
Macromedia Dreamweaver Integration				★		★
Microsoft FrontPage Integration				★		★
Import RoboHelp HTML Projects				★		★
Import Any HTML Help Project				★		★
Import Individual HTML Files				★		★
Use Any HTML Editor				★		★
Import RoboHelp Word Projects			★	★	★	★
Import Any WinHelp Project			★	★	★	★
Doc-To-Help Express			★	★	★	★
Context Sensitive Help for Web Applications			★	★	★	★
Live Update Utility			★	★	★	★
Unicode Support			★	★	★	★
NetHelp Output (for Web deployment)			★	★	★	★
Text-only Pop-ups			★	★	★	★
Expanding Text			★	★	★	★
Drop-down Text			★	★	★	★

Classic Doc-To-Help Features						
Natural Search		★		★		★
Documenter for .NET		★	★	★	★	★
Import Individual Word Files	★	★	★	★	★	★
Use Microsoft Word as Editor	★	★	★	★	★	★
Microsoft Word Integration	★	★	★	★	★	★
Printed Documentation Output	★	★	★	★	★	★
HTML Help Output	★	★	★	★	★	★
WinHelp Output	★	★	★	★	★	★
JavaHelp Output	★	★	★	★	★	★
Visual Studio .NET-style Output	★	★	★	★	★	★
Image Map Editor	★	★	★	★	★	★
Hyperlinks/Topic Links	★	★	★	★	★	★
Topic Pop-ups	★	★	★	★	★	★
Table of Contents and Index	★	★	★	★	★	★
Glossary	★	★	★	★	★	★
Keyword Search	★	★	★	★	★	★
Context Sensitive Help	★	★	★	★	★	★

What's New in Doc-To-Help 2006

Team Authoring Support

Doc-To-Help allows you and your team to collaborate on projects with "check in/check out" synchronization and other team support features. This feature is available in Doc-To-Help Enterprise 2006 only.

You start with a regular, single-user, Doc-To-Help project. You then make it available to other members of your team by uploading it to a central repository, either on a machine on your organization's network or on a Web server. Team members can connect to the team project and create their own working copies of it, where they will make their changes. Once changes are made, the project and document files can be checked in, or uploaded, to the repository and then be retrieved by other team members using special Doc-To-Help commands. Only one author at a time can check out a project or document file, preventing team members from overwriting each other's work.

For more information on using Doc-To-Help's team authoring features, see [Team Authoring](#) (page 514).

PDF Output

Doc-To-Help will now create a PDF file from the Manual target. You can create live, or working, links in the PDF using the [LiveLinks](#) property, as well as create an outline, or bookmarks, in the PDF by setting the [OutlineInPDF](#) property. For more information, see [Creating a PDF](#) (page 127) in the Doc-To-Help Guided Tour.

Document Conditions

Conditions can now be specified in the Doc-To-Help project editor on the document level, in addition to the topic and character style level. You can now designate entire source documents for individual platforms, Help targets, or attributes. See [Conditional Text and Attributes](#) (page 401) for more information.

NetHelp Enhancements (2006 v3)

NetHelp has a brand new look and feel, as well as new features to improve your users' experience. Some of the enhancements include:

- Automatic table of contents (TOC) synchronization: By default, NetHelp automatically synchronizes the TOC with the current topic; therefore, when a user jumps to a topic, that topic automatically becomes the current topic in the TOC. If you prefer not to synchronize the TOC automatically, you can add a **Sync TOC** button to your theme using the Theme Designer. See the [NetHelp Target](#) (page 32) topic for more information.
- Print and email buttons: **Print** and **Email** commands have been added to the Theme Designer. The **Print** command, when clicked, prints the current topic. The **Email** command, when clicked, creates an email with a link to the current topic and opens it in your default email application so you can edit and send the email. You can specify the image and/or text that appears in the theme for each of these commands, as well as determine on which toolbar in the target they appear. For more information, see the [Commands Node \(NetHelp Only\)](#) (page 584) topic.
- Favorites tab and Add to Favorites button: A Favorites pane is now available in the NetHelp target when the [FavoritesTab](#) property is set to **True**. You can customize the **Add to Favorites** button and **Delete** link that allow users to add and remove favorites, respectively, from the new pane. For more information, see [Exploring the Favorites Node \(NetHelp Only\)](#) (page 595).
- Show/hide navigation panes: The new Hide Navigation Pane command allows users to hide the navigation pane, or left pane used for **Contents**, **Index**, **Search** and **Favorites**, in the NetHelp target. For more information, see the [Commands Node \(NetHelp Only\)](#) (page 584) topic.
- Topic URL opens the topic complete with NetHelp panes: Previously, if you opened a topic file directly in a browser, for example, "NetHelp/WordDocuments/introduction.htm", the HTML file was opened without the NetHelp navigation pane and toolbars. Now when you open a topic file, the NetHelp navigation pane and toolbars are visible.
- Other new features in the Theme Designer include:
 - a. It is now possible to configure three toolbars: Main, Left and Topic. Previously, there was only one toolbar, called the Navigation Bar, but this name is no longer used for NetHelp.
 - b. Commands, or buttons, can be placed on any of the three toolbars, rearranged, and grouped. Each button group on a toolbar can be left, center or right aligned. You can specify custom separators between groups.
 - c. The user can now add custom commands, or buttons, to the toolbars. There are two custom commands used in the built-in themes: **Logo**, for the Doc-To-Help logo, and **Caption**, which shows the name of the navigation tab currently selected by the user.
 - d. There is a choice between two looks for every toolbar: **Modern** or **Classic**.
 - e. You can specify the **Body background color**, or the color of the <BODY> tag. The specified color appears in places not covered by the NetHelp panes, which include toolbars and the navigation and topic panes, such as margins.
 - f. Added a new **Email** property to the **Message Files** commands in the **Messages** node.
- Features no longer supported for the NetHelp target, but that are still supported for other targets, include:
 - a. Previously, you could specify an arbitrary background picture for the toolbar, or Navigation Bar. Now, only a gradient picture, which is one pixel wide and stretched the width of the toolbar, when applied, can be specified.
 - b. It is no longer possible to add custom HTML in the areas adjoining the toolbar. This was done in the previous version of Doc-To-Help using the **Navigation Bar | Customize** node. Now, custom buttons, or commands, can be used.

See [Using the Theme Designer](#) (page 572) for more detailed information on all of the NetHelp enhancements.

Section 508 Compliance

You can now create Section 508 compliant NetHelp targets, making the information in your NetHelp readily available for people with visual or hearing impairments or other disabilities. Doc-To-Help allows you to do this using a new [Accessibility](#) property that can enable special compliance features. For example, you can now specify strings shown as tooltip text for links, popups, expanding text and drop-down text. You can also specify the titles, or prompts, to be shown for open or closed books and topics in the table of contents. This text can be easily read by accessibility devices. Additionally, popups, margin notes and glossary term links are shown as jump links rather than pop-up links for easier readability by the accessibility devices. See [Section 508 Compliant NetHelp](#) (page 35) for a complete list and explanation of Section 508 compliance features.

D2HML Enhancements

D2HML now supports links to bookmarks inside topics. This allows you to specify a link to any location inside a topic. In previous versions of Doc-To-Help, such links had to be specified using non-D2HML features, such as cross-references in Word and <A> links in HTML.

D2HML now also supports links to a topic specified by its document and a link tag or a bookmark inside the document. Previously, D2HML only supported links specified by a link tag, which made it difficult to link to topics without unique names in the project. For more information on linking to specific topics, see the [Topic Link Hot Spot](#) (page 252) topic.

Batch Mode

It is now possible to build Help targets in batch mode from a command line. All error and log messages are output to the console standard output. In this case, Doc-To-Help does not open a window; it works as a console application. Batch mode can be used to automate the process of building Help targets. See [Building Help Targets in Batch Mode](#) (page 165) for additional information.

Output Folder Specification

Now you can specify any folder as the output folder of a Help target using the HelpTarget's [Folder](#) property. Previously, only folders within the project directory were supported. Specifying output folders outside of the project directory is useful for backup purposes and for performing automated builds.

If you select a folder outside of the project directory, Doc-To-Help creates a **<target name>** folder within a **DocToHelpOutput** folder in the directory. For example, the directory structure might look something like the following: Project1\DocToHelpOutput\HTMLHelp. This is done to prevent deleting valuable files you may have in that folder, as all files in the output folder are deleted when Doc-To-Help rebuilds a target.

Highlight Search Results in NetHelp

When a search is performed in a NetHelp Help file, the search phrase is now highlighted in the topic text. The user can control whether the search phrase is highlighted by checking or unchecking the **Highlight search hits** checkbox on the **Search** tab of the NetHelp target. See [Exploring the Search Node \(NetHelp Only\)](#) (page 591) for more information.

Documenter: Support for user-defined tags and attributes in source code XML comments

Documenter now supports any custom user-defined tags and attributes in XML comments. This is done through an optional XSLT transformation file that can now be specified in a Documenter project. For more information on creating and specifying an XSLT file, see [Adding User-defined Tags and Attributes to XML Comments](#) (page 477).

Manual Glossary Links

When you build a Help target, Doc-To-Help, by default, examines the project's source documents for text that matches the glossary entries in your glossary document, converting these matches into hyperlinks. Whenever the user clicks one of the hyperlinks, a DHTML pop-up opens containing the corresponding glossary entry description. You now have the option to create glossary links manually rather than having Doc-To-Help automatically create them. Format your glossary headings with the new **Glossary Heading (no auto links)** paragraph style, and use the **Glossary Links** button on the **D2HML Styles** toolbar to format specific text you want to link to a glossary term. For more information, see [Creating a Glossary](#) (page 333).

Converting RoboHelp Layouts

RoboHelp layouts are now converted to Doc-To-Help targets. Most properties specified in a RoboHelp layout are converted to Doc-To-Help properties. For a list of Doc-To-Help supported properties, see [Doc-To-Help RoboHelp, HTML Help and WinHelp Converters](#) (page 136).

Conditional build expressions in RoboHelp layouts are now converted to the [AttributeExpression](#) property in Doc-To-Help. Additionally, advanced conditional build tag expressions containing AND, OR and NOT logical operators are supported by Doc-To-Help. See [Using Attributes](#) (page 422) for additional information.

NetHelp Enhancements (2006 v1)

- Doc-To-Help now supports modular Help projects for NetHelp, in addition to WinHelp and HTML Help. Multiple module projects can be created and merged with a hub project. For more information, see [Creating a Modular Help Project for NetHelp](#) (page 435).
- When you click a link in a NetHelp target that has multiple destinations, the Topics Found HTML page no longer appears. The NetHelp target now provides a new pop-up menu for you to choose which topic you want to view. Use the **Menu** bar in the **Theme Designer** to customize the menu colors and border. See [Using the Theme Designer](#) (page 572) for more information.
- You can now create a noise word list for NetHelp searches. The words provided in the list are ignored when a NetHelp search is performed. See [Using a Noise Word List for NetHelp](#) (page 35) for more information.

Support for Image Maps in D2HML

You can create image maps in HTML source documents by drawing a hot spot and clicking one of the D2HML link buttons on the **D2HML Styles** toolbar. The *D2HML* attribute of the <AREA> tag is used to specify the topic to which you are linking. For more information, see [Using Image Maps](#) (page 329).

Live Links in Printed Manual

Doc-To-Help creates links in the Word document created when you build a Manual target. If you convert the resulting document to a PDF, these links become live, or working, links in the PDF. For more information, see [Creating Live Links](#) (page 126) in the Doc-To-Help Guided Tour.

New Features in Documenter for .NET

- In addition to the standard tags of XML documentation described in the [Microsoft documentation](#), Documenter now supports all tags supported by the free tool NDOC. See [Supported XML Comment Tags](#) (page 477) for more information.
- Documenter now generates links to system classes belonging to the .NET framework, such as the **System.String** class. See [Changing the Project Options](#) (page 473) for more information.
- Documenter can now generate Japanese documentation in addition to English, French, German, Spanish, Italian and Russian. See [Multilanguage Support in Documenter for .NET](#) (page 481) for more information.

Support for D2HML Properties as Attributes in HTML Source Documents

The secondary part of any hot spot in an HTML source document is now specified using the new *d2hml* attribute. This is useful if you do not want the secondary part to appear as text in the design view of your HTML editor. The *d2hml* attribute can also be used to specify the D2HML hot spot style without changing the *class* attribute of an HTML element. See [Using the D2HML Attribute in HTML Source Documents](#) (page 244) for more information.

D2HML Comments in Word

A D2HML comment can be created for any type of D2HML hot spot. Formatting text with a D2HML comment rather than a D2HML style is helpful when you have large ranges of text containing style formatting. Conditional,

expanding and drop-down text is now applied as a comment in your Word documents by default. See [Specifying Conditional Text](#) (page 401) and [Expanding, Dropdown and Popup Text](#) (page 313) for more information.

Collections Tab Icons

The Collections tab of the property browser now shows an icon for each of the following collection items: keyword, group, link tag, and context id. The icon indicates whether the item was added manually or added automatically from the document and what type of element it is: character style, paragraph style, script or Word {XE} field. For more information, see [The Topic Properties Toolbar](#) (page 47).

END-USER LICENSE AGREEMENT FOR COMPONENTONE DOC-TO-HELP

IMPORTANT-READ CAREFULLY: This End-User License Agreement (this "EULA") contains the terms and conditions that govern your use of the SOFTWARE (as defined below) and imposes material limitations to your rights. You should read this EULA carefully and treat it as valuable property.

I. THIS EULA.

- 1. Software Covered by this EULA.** This EULA governs your use of the ComponentOne, LLC ("C1") software product(s) enclosed or otherwise accompanied herewith (individually and collectively, the "SOFTWARE"). The term "SOFTWARE" includes, to the extent provided by C1: 1) any revisions, updates and/or upgrades thereto; 2) any data, image or executable files, databases, data engines, computer software, or similar items customarily used or distributed with computer software products; 3) anything in any form whatsoever intended to be used with or in conjunction with the SOFTWARE; and 4) any associated media, documentation (including physical, electronic and online) and printed materials (the "Documentation").
- 2. This EULA is a Legally Binding Agreement Between You and C1.** If you are acting as an agent of a company or another legal person, such as an officer or other employee acting for your employer, then "you" and "your" mean your principal, the entity or other legal person for whom you are acting. However, importantly, even if you are acting as an agent for another, you may still be personally liable for violation of federal and State laws, such as copyright infringement.

By signifying your acceptance of the terms of this EULA, you intend to be, and hereby are, legally bound to this EULA to the same extent as if C1 and you physically signed this EULA. By installing, copying, or otherwise using the SOFTWARE, you agree to be bound by all the terms and conditions of this EULA. If you do not agree to all of such terms and conditions, **you may not install or use the SOFTWARE.** If you do not agree with any of the terms herewith and, for whatever reason, installation has begun or has been completed, you should cancel installation or un-install the SOFTWARE, as the case may be. Furthermore, you should promptly return the SOFTWARE to the place of business from which you obtained it in accordance with any return policies of such place of business. Return policies may vary among resellers; therefore you must comply with the return policies of your supplier as you agreed at the point of purchase. If the place of business from which you purchased the SOFTWARE does not honor a full refund for a period of thirty (30) days from the date of purchase, you may then return the SOFTWARE directly to C1 for a refund provided that such returns is authorized within the same thirty (30) days time period. To return the product directly to C1, you must first obtain a Return Authorization Number by contacting C1, and you must forward to C1 all items purchased, including the proof of purchase. The return must be postage-prepaid, and post-marked within thirty (30) days from the proof of purchase, time being of the essence. The return option to C1 is only available to the original purchaser of an unopened factory packaged item.

II. YOUR LICENSE TO USE AND TO DISTRIBUTE.

As provided in more detail below, this EULA grants you two licenses: 1) a license to use the SOFTWARE (the "User License") to create online help, manuals or other documentation in electronic or printed format (the "Output Documents"); and 2) a license to distribute the Output Documents and to incorporate in such Output Documents those files identified in the Documentation as Redistributable Files.(the "Distribution License"). These licenses (individually and collectively, the "Licenses") are explained and defined in more detail below. Except for those specific Redistributable Files, you MAY NOT distribute the SOFTWARE, in any format, to others.

1. Definitions. The following terms have the respective meanings as used in this EULA:

"Network Server" means a computer with one or more computer central processing units (CPU's) that operates for the purpose of serving other computers logically or physically connected to it, including, but not limited to, other computers connected to it on an internal network, intranet or the Internet.

"Web Server" means a type of Network Server that serves other computers which, are specifically connected to it through either an intranet or the Internet.

"Redistributable Files" means the SOFTWARE files or other portions of the SOFTWARE that are provided by C1 and are identified as such in the Documentation for distribution by you with the Output Documents.

"User" means a human being or any other automated device using the SOFTWARE in accordance with the terms and conditions of this EULA.

"User Seat License" means that each User using or otherwise accessing the SOFTWARE must obtain the right to do so by purchasing a separate End User License.

2. Your User License.

License to Use the Software. You are hereby granted a limited, royalty-free, non-exclusive right to use the SOFTWARE to create Output Documents such as online help, manuals or other documentation in electronic or printed format, on the express condition that, and only for so long as, you fully comply with all terms and conditions of this EULA.

The SOFTWARE is licensed to you on a User Seat License basis.

User Seat License basis means that you may perform an installation of the SOFTWARE for use in creating Output Documents by a single User on one or more computers, each with a single set of input devices, so long as 1) such computer/computers is/are used only by one single User at any given time and not concurrently and, 2) the user is the primary User to whom the license has been granted. Conversely, you may not install or use the SOFTWARE on a computer that is a network server or a computer at which the SOFTWARE is used by more than one User. You may not network the SOFTWARE or any component part of it, where it is or may be used by more than one User unless you purchase an additional User License for each User. You must purchase another separate license to the SOFTWARE in order to add additional User seats, whether the additional Users are accessing the SOFTWARE in a stand-alone environment or on a computer network.

The license rights granted under this Agreement may be limited to a specified number of days after you first install the SOFTWARE unless you supply information required to license or activate your licensed copy, as the case may be, within the time and the manner described during the SOFTWARE setup sequence and/or in the dialog boxes appearing during use of the SOFTWARE. You may need to activate the SOFTWARE through the use of the Internet, email or telephone; toll charges may apply. You may need to re-activate the SOFTWARE if you modify your computer hardware or if you have installed it on a different computer; in some cases the number of activations allowed may be limited and you will have to contact C1 for clearance. Product activation is based on the exchange of information between your computer and C1. None of this information contains personally identifiable information nor can they be used to identify any personal information about you or any information you store in your computer. YOU ACKNOWLEDGE AND UNDERSTAND THAT THERE ARE TECHNOLOGICAL MEASURES IN THE SOFTWARE THAT ARE DESIGNED TO PREVENT UNLICENSED OR ILLEGAL USE OF THE SOFTWARE. YOU AGREE THAT C1 MAY USE SUCH MEASURES AND YOU AGREE TO FOLLOW ANY REQUIREMENTS REGARDING SUCH TECHNOLOGICAL MEASURES. YOU ACKNOWLEDGE AND AGREE THAT THE SOFTWARE WILL CEASE TO FUNCTION UNLESS AND UNTIL YOU ACTIVATE THE APPLICABLE SOFTWARE SERIAL NUMBER.

You agree that C1 may audit your use of the SOFTWARE for compliance with these terms at any time, upon reasonable notice. In the event that such audit reveals any use of the SOFTWARE other than in full compliance with the terms of this EULA, you shall reimburse C1 for all reasonable expenses related to such audit in addition to any other liabilities you may incur as a result of such non-compliance.

In all cases, (a) you may not use C1's name, logo, or trademarks to market your Output Documents without the express written consent of C1 except in the default themes legend that may appear on the Output Documents which includes the C1 logo and a line of text reading "Powered by ComponentOne Doc-To-Help"; (b) you agree to indemnify, hold harmless, and defend C1, its suppliers and resellers, from and against any claims or lawsuits, including attorney's fees that may arise from the use or distribution of your Output Documents.

3. Your Distribution License.

License to Distribute Output Documents. Subject to the terms and conditions in this EULA, you are granted the license to use and to distribute Output Documents on a royalty-free basis, provided that if the Output Documents are distributed in electronic format and incorporate portions of the SOFTWARE they do so as an integral part of the Output Documents (customarily a ".chm" file, etc.). You may distribute, on a royalty-free basis, Redistributable Files with Output Documents only.

4. **Updates/Upgrades; Subscription.** Subject to the terms and conditions of this EULA, the Licenses are perpetual. Updates and upgrades to the SOFTWARE may be provided by C1 from time-to-time, and, if so provided by C1, are provided upon the terms and conditions offered at that time by C1 in its sole discretion. C1 may provide updates and upgrades to the SOFTWARE for free or for any charge, at any time or never, and through its chosen manner of access and distribution, all in C1's sole discretion.

The SOFTWARE is revised from time-to-time (meaning, for example, revised with updates and/or upgrades). To receive any such revisions to the SOFTWARE, you must have a valid SOFTWARE subscription. Together with the Licenses, the original purchaser is granted a one-year subscription from the date of purchase. Upon expiration, you must renew your license subscription to continue to be entitled to receive SOFTWARE revisions.

5. **Serial Number.** With your license, you will be issued a unique serial number (the "**Serial Number**") used for the activation of the SOFTWARE. The Serial Number is subject to the restrictions set forth in this EULA and may not be disclosed or distributed either with your Output Documents or in any other way. The disclosure or distribution of the Serial Number constitutes a breach of this EULA, the effect of which shall be the immediate termination and revocation of all the rights granted herein.
6. **Evaluation Copy.** If you are using an "evaluation copy", specifically designated as such by C1 on its website or elsewhere, then the Licenses are limited as follows: a) you are granted a license to use the SOFTWARE for a period of thirty (30) days counted from the day of installation (the "**Evaluation Period**"); b) upon completion of the Evaluation Period, you shall either i) delete the SOFTWARE from the computer containing the installation, or you may ii) obtain a paid license of the SOFTWARE from C1 or any of its resellers; and c) any Output Documents created with the Evaluation Copy may not be distributed or used for any commercial purpose.

III. INTELLECTUAL PROPERTY.

1. **Copyright.** You agree that all right, title, and interest in and to the SOFTWARE (including, but not limited to, any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the SOFTWARE), and any copies of the SOFTWARE, and any copyrights and other intellectual properties therein or related thereto are owned exclusively by C1, except to the limited extent that C1 may be the rightful license holder of certain third-party technologies incorporated into the SOFTWARE. The SOFTWARE is protected by copyright laws and international treaty provisions. **The SOFTWARE is licensed to you, not sold to you.** C1 reserves all rights not otherwise expressly and specifically granted to you in this EULA.
2. **Backups.** You may make a copy of the SOFTWARE solely for backup or archival purposes. Notwithstanding the foregoing, you may not copy the printed Documentation.
3. **General Limitations.** You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that applicable law expressly permits such activity notwithstanding this limitation.
4. **Software Transfers.** You may not rent or lease the SOFTWARE. You may permanently transfer all of your rights under the EULA, provided that you retain no copies, that you transfer all the SOFTWARE (including all component parts, the media and printed materials, any updates, upgrades, this EULA and, if applicable,

the Certificate of Authenticity), and that the transferee agrees to be bound by the terms of this EULA. If the SOFTWARE is an update or upgrade, any transfer must include all prior versions of the SOFTWARE.

5. **Termination.** Without prejudice to any other rights it may have, C1 may terminate this EULA and the Licenses if you fail to comply with the terms and conditions contained herein. In such an event, you must destroy all copies of the SOFTWARE and all of its component parts.
6. **Export Restrictions.** You acknowledge that the SOFTWARE is of U.S. origin. You acknowledge that the license and distribution of the SOFTWARE is subject to the export control laws and regulations of the United States of America, and any amendments thereof, which restrict exports and re-exports of software, technical data, and direct products of technical data, including services and Developed Software. You agree that you will not export or re-export the SOFTWARE or any Developed Software, or any information, documentation and/or printed materials related thereto, directly or indirectly, without first obtaining permission to do so as required from the United States of America Department of Commerce's Bureau of Industry and Security ("**BIS**"), or other appropriate governmental agencies, to any countries, end-users, or for any end-uses that are restricted by U.S. export laws and regulations, and any amendments thereof, which include, but are not limited to: Restricted Countries, Restricted End-Users, and Restricted End-Uses.

These restrictions change from time to time. You represent and warrant that neither the BIS nor any other United States federal agency has suspended, revoked or denied your export privileges. C1 acknowledges that it shall use reasonable efforts to supply you with all reasonably necessary information regarding the SOFTWARE and its business to enable you to fully comply with the provisions of this Section. If you have any questions regarding your obligations under United States of America export regulations, you should contact the Bureau of Industry and Security, United States Department of Commerce, Exporter Counseling Division, Washington DC. U.S.A. (202) 482-4811, <http://www.bis.doc.gov>.

7. **U.S. Government Restricted Rights.** This Software and the documentation are provided with "RESTRICTED RIGHTS" applicable to private and public licenses alike. The Software is a "commercial item," as that term is defined at 48 C.F.R. 2.101 (Oct 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sep 1995) and is provided to the U.S. Government only as a commercial end item. Any technical data provided with such Software is commercial technical data as defined in 48 C.F.R. 12.211 (Sep 1995). Consistent with 48 C.F.R. 12.211 through 12.212, 48 C.F.R. 227.7202-1 through 227.7202-4 (Jun 1995), and 48 C.F.R. 252.227-7015 (Nov 1995), all U.S. Government End Users acquire the Software with only those rights expressly set forth in this EULA.

Without limiting the foregoing, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in this EULA and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013 (c)(1)(ii)(OCT 1988), FAR 12.212(a)(1995), FAR 52.227-19, or FAR 52.227-14, as applicable. For purposes of these regulations the

Manufacturer is ComponentOne, LLC, 201 South Highland Avenue , 3rd Floor, Pittsburgh, Pennsylvania 15206 USA.

IV. WARRANTIES AND REMEDIES.

1. **Limited Warranty.** C1 warrants that the original media, if any, are free from defects for ninety (90) days from the date of delivery of the SOFTWARE. C1 also warrants that: (i) it has the full power to enter into this Agreement and grant the license rights set forth herein; (ii) it has not granted and will not grant any rights in the Software to any third party which grant is inconsistent with the rights granted to you in this Agreement; and (iii) the Software does not and will not infringe any trade secret, copyright, trademark or other proprietary right held by any third party and does not infringe any patent held by any third party. **EXCEPT AS OTHERWISE PROVIDED IN THE PRECEDING SENTENCE, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, C1 EXPRESSLY DISCLAIMS ANY WARRANTY FOR THE SOFTWARE, DOCUMENTATION AND ANYTHING ELSE PROVIDED BY C1 HEREBY AND C1 PROVIDES THE SAME IN "AS IS" CONDITION WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION**

**REMAINS WITH YOU. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS.
YOU MAY HAVE OTHERS WHICH VARY FROM STATE TO STATE.**

- Limited Remedy.** C1 PROVIDES NO REMEDIES OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, FOR ANY SAMPLE APPLICATION CODE, REDISTRIBUTABLE FILES, TRIAL VERSION AND THE NOT FOR RESALE VERSION OF THE SOFTWARE. ANY SAMPLE APPLICATION CODE, TRIAL VERSION AND THE NOT FOR RESALE VERSION OF THE SOFTWARE ARE PROVIDED "AS IS".

C1's entire liability and your exclusive remedy under this EULA shall be, at C1's sole option, either (a) return of the price paid for the SOFTWARE; (b) repair the SOFTWARE through updates distributed online or otherwise in C1's discretion; or (c) replace the SOFTWARE with SOFTWARE that substantially performs as described in the SOFTWARE documentation, provided that you return the SOFTWARE in the same manner as provided in Section I.2 for return of the SOFTWARE for non-acceptance of this EULA. Any media for any repaired or replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. THESE REMEDIES ARE NOT AVAILABLE OUTSIDE OF THE UNITED STATES OF AMERICA. **TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL C1 BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFIT, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF C1 HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES IN CERTAIN CASES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.**

V. MISCELLANEOUS.

- This is the Entire Agreement.** This EULA (including any addendum to this EULA included with the SOFTWARE) is the final, complete and exclusive statement of the entire agreement between you and C1 relating to the SOFTWARE. This EULA supersedes any prior and contemporaneous proposals, purchase orders, advertisements, and all other communications in relation to the subject matter of this EULA, whether oral or written. No terms or conditions, other than those contained herein, and no other understanding or agreement which in any way modifies these terms and conditions, shall be binding upon the parties unless entered into in writing executed between the parties, or by other non-oral manner of agreement whereby the parties objectively and definitively act in a manner to be bound (such as by continuing with an installation of the SOFTWARE, etc.). Employees, agents and other representatives of C1 are not permitted to orally modify this EULA.
- You Indemnify C1.** You agree to indemnify, hold harmless, and defend C1 and its suppliers and resellers from and against any and all claims or lawsuits, including attorney's fees, which arise out of or result from your distribution of Output Documents or from your breach of any of the terms and conditions of this EULA.
- Interpretation of this EULA.** If for any reason a court of competent jurisdiction finds any provision of this EULA, or any portion thereof, to be unenforceable, that provision of this EULA will be enforced to the maximum extent permissible so as to effect the intent of the parties, and the remainder of this EULA will continue in full force and effect. Formatives of defined terms shall have the same meaning of the defined term. Failure by either party to enforce any provision of this EULA will not be deemed a waiver of future enforcement of that or any other provision. Except as otherwise required or superseded by law, this EULA is governed by the laws of the Commonwealth of Pennsylvania, without regard to its conflict of laws principles. The parties consent to the personal jurisdiction and venue of the Commonwealth of Pennsylvania, in the County of Allegheny, and agree that any legal proceedings arising out of this EULA shall be conducted solely in such Commonwealth. If the SOFTWARE was acquired outside the United States, then local law may apply.

License and Redistributable Files

Doc-To-Help 2006 is developed and published by ComponentOne LLC. You may use it for Help authoring in conjunction with Microsoft Word. You may also distribute the following files, royalty free, with any online Help system or printed manual you develop:

- Any file generated by Doc-To-Help within an output subdirectory relative to a Doc-To-Help project file.

End-users of your online Help systems and print manuals are **not** licensed to use Doc-To-Help for authoring, and may **not** redistribute any file not cited above.

You are **not** licensed to distribute any Doc-To-Help file to users for development purposes. You are **not** allowed to add or transfer the Doc-To-Help serial number to the registry of your users' computer(s).

You are **not** licensed to redistribute any version of Microsoft Internet Explorer provided with Doc-To-Help.

It is your responsibility to make such restrictions clear to your users.

WARNING: Doc-To-Help must be licensed within 30 days of installation in order to continue using the product.

Technical Support

Doc-To-Help 2006 is developed and supported by ComponentOne LLC, a company formed by the merger of APEX Software Corporation and VideoSoft.

ComponentOne offers various support options. For a complete list and a description of each, visit the Doc-To-Help Web site at <http://www.doctohelp.com/Support>.

Some methods for obtaining technical support include:

- **Online Support via [HelpCentral](#)**
ComponentOne HelpCentral provides customers with a comprehensive set of technical resources in the form of [FAQs](#), [samples](#), [TechTips](#), [Version Release History](#), [Articles](#), searchable [Knowledge Base](#), searchable [Online Help](#) and more. We recommend this as the first place to look for answers to your Doc-To-Help technical questions.
- **Online Support via our Incident Submission Form**
This online support service provides you with direct access to our Technical Support staff via an online incident submission form. When you submit an incident, you'll immediately receive a response via e-mail confirming that you've successfully created an incident. This email will provide you with an Issue Reference ID and will provide you with a set of possible answers to your question from our Knowledgebase. You will receive a response from one of the ComponentOne staff members via e-mail in 2 business days or less.
- **Telephone Support**
Standard support also includes thirty (30) days of telephone support for Doc-To-Help customers. This service starts from the date of your first call to the technical support department.
- **Peer-to-Peer Product Forums and Newsgroups**
Doc-To-Help peer-to-peer product [forums and newsgroups](#) are available to exchange information, tips, and techniques regarding Doc-To-Help. ComponentOne sponsors these areas as a forum for users to share information. While ComponentOne does not provide direct support in the forums and newsgroups, we periodically monitor them to ensure accuracy of information and provide comments when appropriate. Please note that a ComponentOne User Account is required to participate in the Product Forums.
- **Installation Issues**
Registered users can obtain help with problems installing Doc-To-Help. Contact technical support by using the online incident submission form or by phone (412.681.4738).
- **Documentation**
Doc-To-Help documentation is available in HTML Help and PDF format. All of the PDFs are also available on [HelpCentral](#). If you have suggestions on how we can improve our documentation, please

email the [Documentation team](#). Please note that e-mail sent to the [Documentation team](#) is for documentation feedback only. [Technical Support](#) and [Sales](#) issues should be sent directly to their respective departments.

Note: You must create a ComponentOne Account and register your product with a valid serial number to obtain support using some of the above methods.

System Requirements

Computer/Processor	PC with Pentium II 500 MHz or greater processor
Memory	128 MB of RAM Minimum
Hard Disk Space	25 MB of hard disk space for the Doc-To-Help application and related files 13 MB of hard disk space for MDAC 2.5
Operating System	Microsoft Windows 98, ME, XP, 2000 or NT 4.0 (with Service Pack 3 or later)
Microsoft Word	Microsoft Word 2000 (version 9.0) or greater
.NET Framework	Microsoft .NET framework version 1.1 Microsoft .NET framework version 2.0 (required for Team Authoring only)
Microsoft Data Access Components (MDAC)	MDAC version 2.6 or later

Using Anti-virus Software with Doc-To-Help

Note that some anti-virus software can limit the functionality of a range of software applications, including Doc-To-Help.

For example, anti-virus software may:

- Interfere with the integration between Doc-To-Help and Microsoft Word.
- Limit or prohibit the use of scripting technology.

If your anti-virus software alerts you with a warning while using Doc-To-Help, it may be necessary to modify the properties of your anti-virus software.

For known issues relating to anti-virus software, contact [Technical Support](#).

Installation

Insert the Doc-To-Help 2006 product CD in your CD-ROM drive and follow the instructions presented.

If your Doc-To-Help 2006 product CD does not auto start, click **Start** from the Windows Task Bar, choose **Run**, then type SETUP.EXE in the **Open** textbox and click **OK**.

Authoring Templates for Microsoft Word

Doc-To-Help 2006 includes the Microsoft Word document templates used to produce its own documentation.

Although you are not required to use them, you may find these templates helpful as a starting point.

C1H_DOTNET_HLP.DOT	This template is used to format WinHelp for Documenter for .NET projects.
C1H_DOTNET_HTML.DOT	This is the template used to format both standard HTML and Microsoft HTML Help for Documenter for .NET projects.

C1H_DOTNET_PRN.DOT	This is the template used to format the printed manual target for Documenter for .NET projects.
C1H_DOTNET_SRC.DOT	This is the template used to author the source documents for Documenter for .NET.
C1H_HELP.DOT	This is the template used to format WinHelp.
C1H_HTML.DOT	This is the template used to format both standard HTML and Microsoft HTML Help.
C1H_IN.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process.
C1H_IN_A4.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process (A4 size paper).
C1H_INSD.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process.
C1H_INSD_A4.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process (A4 size paper).
C1H_INSM.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process.
C1H_INSM_A4.DOT	This template is used during the Doc-To-Help 2000 to Doc-To-Help 2006 conversion process (A4 size paper).
C1H_NOMARGIN.DOT	This is the template used to author the source documents. It differs from the default template C1H_NORM.DOT in that it does not have the wide two inch left margin.
C1H_NORM.DOT	This is the template used to author the source documents.
C1H_NORM_A4.DOT	This is the template used to author the source documents (A4 size paper).
C1H_PRNOMARGIN.DOT	This is the template used to format the printed manual target. It differs from the default template C1H_PRNORM.DOT in that it does not have the wide two inch left margin.
C1H_PRNORM.DOT	This is the template used to format the printed manual target.
C1H_PRNORM_A4.DOT	This is the template used to format the printed manual target (A4 size paper).
C1H_PRSIDE.DOT	This is the template used to format the standard sidehead printed manual.
C1H_PRSIDE_A4.DOT	This is the template used to format the standard sidehead printed manual (A4 size paper).
C1H_PRSMAL.DOT	This is the template used to format the standard small printed manual. Note that the actual fonts used may not be present on your system.
C1H_PRSMAL_A4.DOT	This is the template used to format the standard small printed manual (A4 size paper). Note that the actual fonts used may not be present on your system.
C1H_SIDE.DOT	This is the template used to format the standard sidehead source documents.
C1H_SIDE_A4.DOT	This is the template used to format the standard sidehead source documents (A4 size paper).
C1H_SMAL.DOT	This is the template used to format the standard small source documents. Note that the actual fonts used may not be present on your system.
C1H_SMAL_A4.DOT	This is the template used to format the standard small source documents (A4 size paper). Note that the actual fonts used may not be present on your system.

Doc-To-Help setup installs the templates in Microsoft Word's default template folder and also places backup copies in the *Templates* subdirectory of the Doc-To-Help installation directory.

For more information, see [Adding Templates to a Project](#) (page 199).

Global Templates

Doc-To-Help also installs two global templates named `C1D2HAuthor.dot` and `C1D2HEngine.dot` into the Microsoft Office startup directory.

Doc-To-Help Cascading Style Sheets

Doc-To-Help style sheets are installed in the `DefaultCSSFiles` subdirectory of the Doc-To-Help installation directory; the default location is `C:\Program Files\DocToHelp\DefaultCSSFiles`. They are duplicated in the `DefaultCSSFiles` subdirectory in each project, and Doc-To-Help automatically keeps those copies in the project directory in sync with the original style sheets in the installation directory.

Here is the list of style sheets provided with Doc-To-Help:

<code>C1H_Source_full.css</code>	Source style sheet, full-style
<code>C1H_Source_short.css</code>	Source style sheet, minimal-style
<code>C1H_HTML_full.css</code>	Target style sheet for all HTML-based targets, full-style
<code>C1H_HTML_short.css</code>	Target style sheet for all HTML-based targets, minimal-style
<code>C1H_Help_full.css</code>	Target style sheet for WinHelp target, full-style
<code>C1H_Help_short.css</code>	Target style sheet for WinHelp target, minimal-style
<code>C1H_Print_full.css</code>	Target style sheet for Manual target, full-style
<code>C1H_Print_short.css</code>	Target style sheet for Manual target, minimal-style
<code>C1H_Print_nomargin.css</code>	Target style sheet for Manual target, full-style, without the wide two inch left margin
<code>C1H_dotnet_src.css</code>	Source style sheet for Documenter projects, full-style
<code>C1H_dotnet_src_short.css</code>	Source style sheet for Documenter projects, minimal-style
<code>C1H_dotnet_html.css</code>	Target style sheet for Documenter projects for HTML-based targets, full-style
<code>C1H_dotnet_html_short.css</code>	Target style sheet for Documenter projects for HTML-based targets, minimal-style
<code>C1H_dotnet_prn.css</code>	Target style sheet for Documenter projects for Manual target, full-style
<code>C1H_dotnet_prn_short.css</code>	Target style sheet for Documenter projects for Manual target, minimal-style

See [Using Cascading Style Sheets](#) (page 201) for additional information.

Help Files

Doc-To-Help places five HTML Help files in the `DocToHelp` directory.

<code>C1D2H.CHM</code>	This is the primary help file, providing you with online documentation for Doc-To-Help.
<code>C1D2HAPI.CHM</code>	This subsidiary help file is used to populate the Help Pane of the Doc-To-Help project editor and is not a “stand alone” file. This file should not be renamed or moved from the directory where the Doc-To-Help application resides.

FPDreamD2HML.CHM	This help file contains a tutorial for using Doc-To-Help Markup Language (D2HML) with Microsoft FrontPage and Macromedia Dreamweaver.
D2HMLOther.CHM	This help file contains a tutorial for using D2HML with HTML editors that are not integrated with Doc-To-Help.
WordD2HML.CHM	This help file contains a tutorial for using D2HML with Microsoft Word.

Microsoft Help and HTML Help Workshop

Microsoft Help Workshop

The Microsoft Help Workshop (for WinHelp) is included on the Doc-To-Help 2006 CD. To install it, run **hcwsetup.exe** from the **WinHelpWorkshop** folder on the CD.

For more information on the Microsoft Help Workshop, [search the Microsoft Web site](#) for **hcwsetup.exe**.

Microsoft HTML Help Workshop

The HTML Help Workshop is included on the Doc-To-Help 2006 CD. To install it, run **htmlhelp.exe** from the **HTMLHelpWorkshop** folder on the CD.

For more information on the Microsoft HTML Help Workshop, [search the Microsoft Web site](#) for **HTML Help Workshop** or visit the HTML Help Start Page at <http://msdn.microsoft.com/library/en-us/htmlhelp/html/vsconhh1start.asp>.

About This Document

[Help Authoring Basics](#) (page 19) describes the key elements of a Help system and provides a brief overview of how they are implemented using Doc-To-Help.

[The Doc-To-Help Project Editor](#) (page 41) describes the main features of the user interface as well as a brief description of the properties for each feature.

[A Guided Tour of Doc-To-Help](#) (page 51) contains a detailed set of instructions for transforming a set of ordinary Microsoft Word and HTML documents (included with Doc-To-Help) into online Help for multiple platforms.

Note: Throughout this documentation we refer to Microsoft FrontPage when editing HTML source documents. You may edit your HTML source documents with any HTML editor you choose.

[Working with Projects](#) (page 129) contains instructions on creating, opening and working with the Help project within the project editor.

[Doc-To-Help 2006 Source Documents](#) (page 181) explains the types of source documents Doc-To-Help 2006 supports, how documents are displayed within the project and how to specify the types documents used in the project.

[Templates and Cascading Style Sheets](#) (page 199) provides instructions on using templates and cascading style sheets within the Doc-To-Help project.

[Using Styles in Doc-To-Help](#) (page 211) provides information on how to define and modify paragraph and character styles within the project editor.

[Doc-To-Help Markup Language \(D2HML\)](#) (page 233) provides information on formatting your source documents with D2HML styles to create links, keywords, conditional text, and much more.

[Defining and Organizing Topics](#) (page 279) provides information on how to use the project editor to develop your topic structure, organize the navigation sequence and modify the contents of your Help project.

[Links and Hot Spots](#) (page 309) contains instructions on developing your links, cross-references, margin notes and mid topic jumps.

[Creating a Glossary](#) (page 333) provides information on how to specify a glossary document and then add, format and link to glossary terms.

[Building an Index](#) (page 341) contains instructions on how to define you index, add primary and secondary keywords and create topic groups.

[Customizing Help Windows](#) (page 353) provides information on how to define and modify your Help windows.

[Tips and Techniques for Word](#) (page 365) illustrates the advanced features of Doc-to-Help: graphics and Internet links, printing, reporting, modifying the glossary and project file management.

[Conditional Text and Attributes](#) (page 401) provides information on how to specify conditional text and how to use attributes to refine your Help files for specific audiences.

[Using Modular Help](#) (page 427) explains how to create and manage a modular Help project for WinHelp or HTML Help.

[Scripting Techniques](#) (page 439) explains the use of Microsoft VBScript code in Doc-to-Help projects, with examples of common operations.

[Documenter for .NET](#) (page 449) contains instructions that allow you to use Documenter to automatically create reference documentation by simply selecting a .NET assembly.

[Doc-To-Help Express](#) (page 495) demonstrates how to use Doc-To-Help's simplified, wizard-like interface.

[Using Natural Search for Doc-To-Help](#) (page 548) describes how to implement our natural language search component.

[Using the Image Map Editor in Microsoft Word](#) (page 560) demonstrates how to define topic link hot spots by using the image map control.

[The Modular TOC Utility](#) (page 563) contains instructions on how to modify your modular Help system so that both parent and child Help files have a full table of contents.

[Using the Context String Editor](#) (page 569) demonstrates how to add Microsoft Help 2.0 context strings to your Help project.

[Using the Theme Designer](#) (page 572) provides information on how to define and save custom themes for HTML Help, HTML 4.0 and Help 2.0 targets.

[Using the Topic Tools in Word](#) (page 604) describes how to add, rename and remove fully functional topics without rebuilding your project.

[Object Model Reference](#) (page 611) describes the individual properties and methods exposed by the Doc-to-Help object model.

Help Authoring Basics

If you are new to Help authoring, please read this chapter to learn about the basic elements of a Help system and how you would implement them with ComponentOne Doc-To-Help 2006.

If you are an experienced Help author, you can skip the introductory section, but the remainder of this chapter covers key information that you need to understand in order to get the most out of Doc-To-Help.

What is a Help System?

From the reader's perspective, a Help system incorporates some or all of the following five elements:

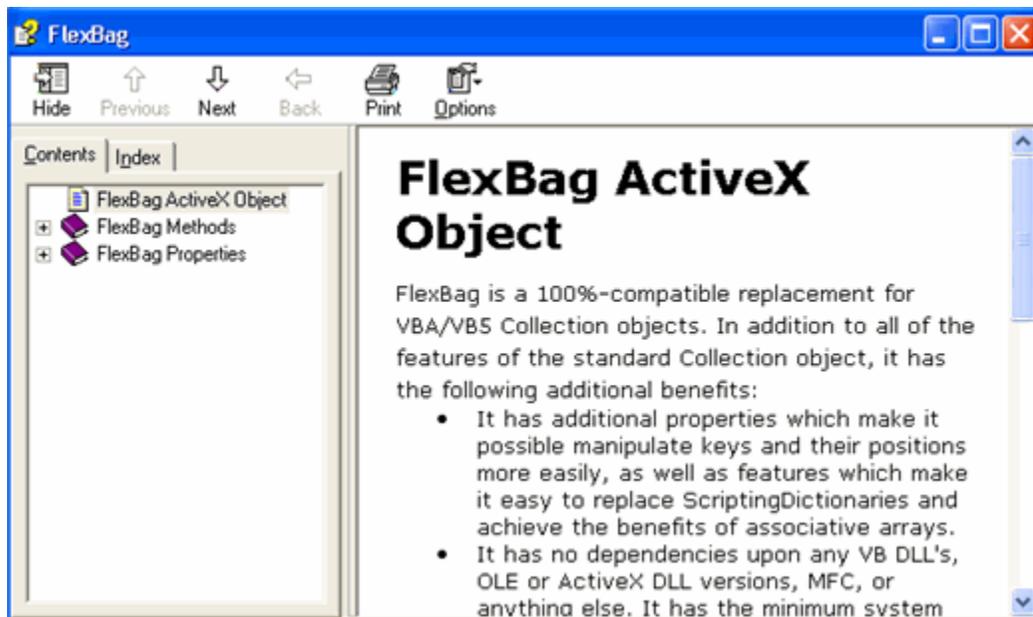
- Topics** A set of "pages" containing static text and graphics.
- Contents** A hierarchical view, or outline, of the entire Help system.
- Index** A searchable list of keywords, each of which corresponds to one or more topics.
- Navigation** A mechanism for accessing topics sequentially or by subtopic.
- Hypertext** Interactive elements that link individual words and phrases with other topics or topic groups.

Note that this definition of a Help system doesn't presume a particular platform or even a computer! A printed manual is a Help system of sorts, as it has a set of topics, a table of contents, an index, and navigation (via your thumb and forefinger). A typical home page on the World Wide Web doesn't have a site map or a searchable index, but may include hyperlinks to other "Help topics" or a navigation sequence that steps through a set of vacation photos or baby pictures.

Topics Defined

A topic comprises static text and graphics that represents a "page" of information. In a well-organized Help system, topics are concise and serve a specific purpose, such as explaining how to carry out a procedure, defining a glossary term, or providing a list of related topics.

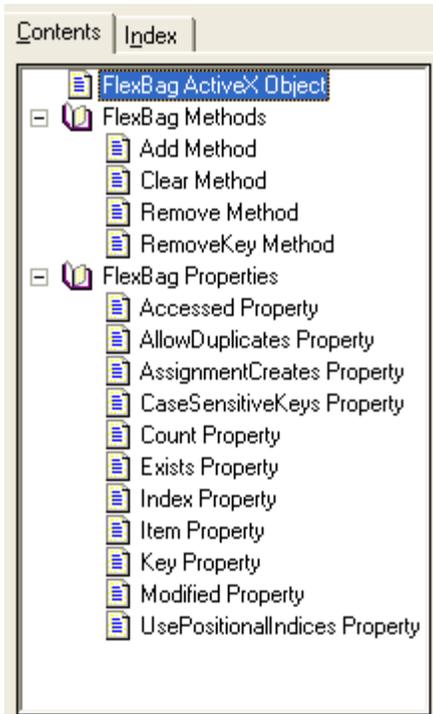
The following figure shows a topic in HTML Help.



Contents Defined

The term contents denotes a hierarchical view, or outline, of the entire Help system that lists topic titles in a way that shows parent-child relationships. In an online Help system, selecting a title displays the corresponding topic. In a printed manual, the table of contents shows the corresponding page number.

The following figure shows the Contents tab in HTML Help.



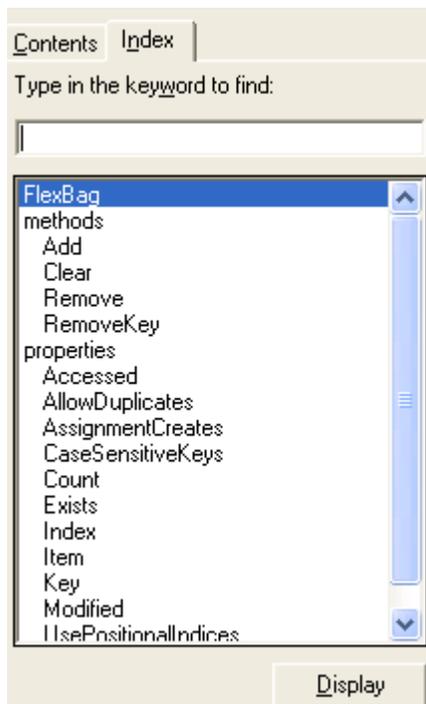
Glossary Defined

A glossary is a list of specialized words with their definitions, often placed at the end of a book. In an online Help system, glossary terms can be formatted to appear as hyperlinks which, when clicked, open or jump to a window containing the definition of the term. Doc-To-Help automatically creates an empty glossary document whenever a new project is created.

Index Defined

An index is a searchable list of keywords, each of which corresponds to one or more topics. The author of a well-designed index anticipates which words and phrases the reader is likely to look for and associates them with matching Help topics. In an online Help system, selecting a keyword displays a list of the associated topics (or the topic itself if there is only one). In a printed manual, the index at the back of the book shows the page numbers that correspond to each keyword.

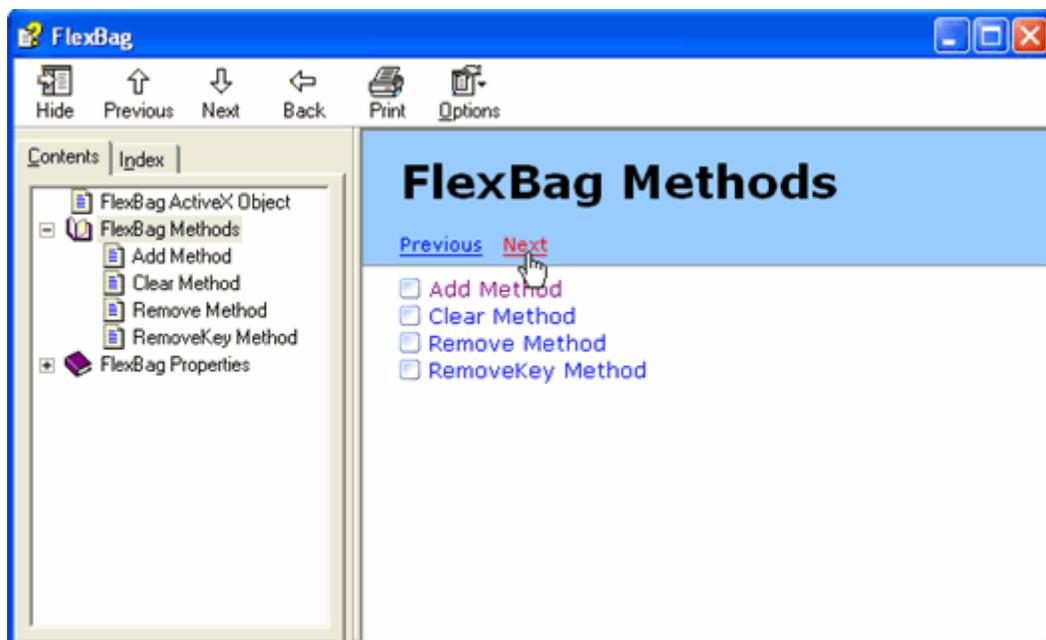
The following figure shows the Index tab in HTML Help.



Navigation Defined

The term navigation refers to a means whereby the reader can access topics sequentially in an order defined by the author. Individual topics can also contain navigational aids that provide quick access to a list of subtopics.

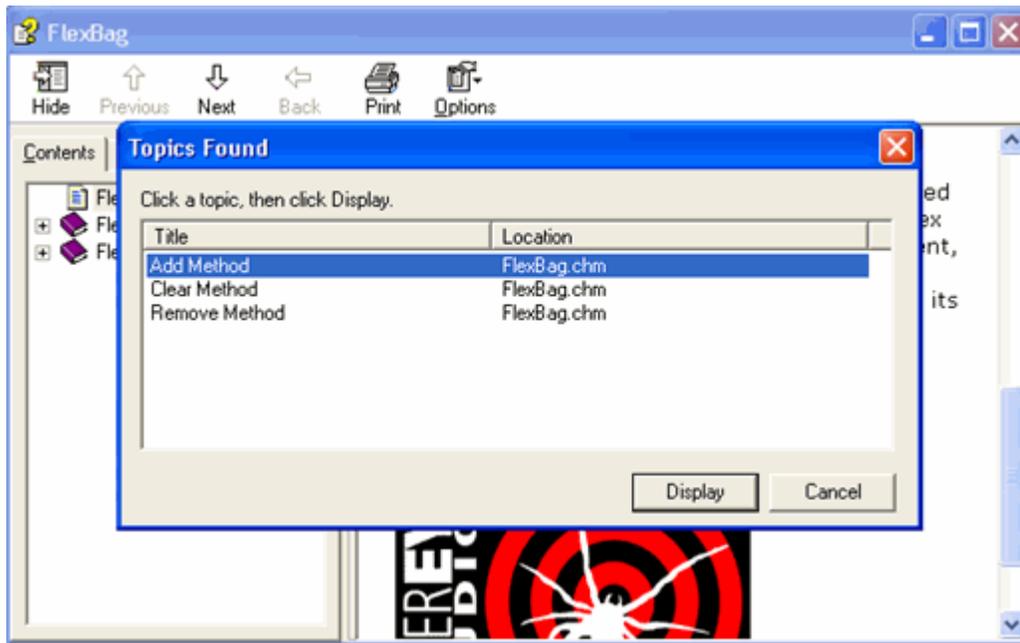
The following figure shows an HTML Help topic that provides two kinds of navigation. The Previous, Next and Back buttons on the HTML Help toolbar let the reader browse the topics. The small buttons within the topic itself let the reader jump directly to one of its subtopics.



Hypertext Defined

The term hypertext refers to interactive elements that link individual words and phrases with other topics or topic groups. A good interactive Help system does more than present concise information in a logical order; it incorporates hypertext elements such as jumps and pop-ups that guide the reader to related topics without imposing a particular sequence. Authors can also use hypertext to keep related topics accessible without presenting an overwhelming amount of information to the reader in a single topic.

The following figure shows a list of related topics displayed by a hypertext link in HTML Help.



Building Help Systems with Doc-To-Help

As an author, you implement the five fundamental elements of a Help system by correlating your source documents with objects you define in a Doc-To-Help project. This sounds more complicated than it is; if you start with a good outline in Microsoft Word or your HTML files, you're well on your way to producing a full-fledged Help system.

Doc-To-Help defines the elements of a Help system as follows:

Topics	Contiguous source document regions delimited by active paragraph styles.
Contents	A hierarchy derived from paragraph style outline levels, similar to Word's document map.
Index	A set of associations between keywords and topics. Keywords can be generated from styles automatically, manually entered by the author, or scripted.
Navigation	Sequential navigation is defined by the order of topics within a document. Subtopic navigation is derived from paragraph style outline levels.
Hypertext	Words or phrases, formatted with an active character style that textually match a single topic, or a set of topics via an index element.

Understanding the Doc-To-Help Build Options

When working with Doc-To-Help 2006, you will encounter three terms that, while somewhat similar in function, are distinct in how they are used in the context of our product.

Compile

In general, the term **Compile** is used to describe the internal process by which Doc-To-Help translates your Microsoft Word or HTML source documents into help target files.

By right-clicking on any document, you can use the **Compile** command to update an individual source document, without updating the help target itself. For example, if a build fails because a topic title is too long, you can edit the document containing the error, and then compile it by itself to determine if the error has been fixed.

Make Target and Rebuild Target

The **Make Target** and **Rebuild Target** commands are similar in functionality, but different in scope. They both govern compilation – the difference lies in *what* each command actually compiles.

The **Rebuild Target** command removes all files from the output directory and recompiles your entire project, placing all new files in your output directory. This is most commonly used when you are making some sort of “global” change that affects the entire help target. For example, when adding a new term to the glossary, it is necessary for Doc-To-Help to search all source documents for instances of the new term. In this case, a **Rebuild Target** is required.

Alternately, the **Make Target** command modifies only those items (source documents, properties) that have changed since the last build. **Make Target** can be a time saver when working with a very large help project that includes many source documents because it compiles only those documents that you have modified recently.

Defining Help Topics

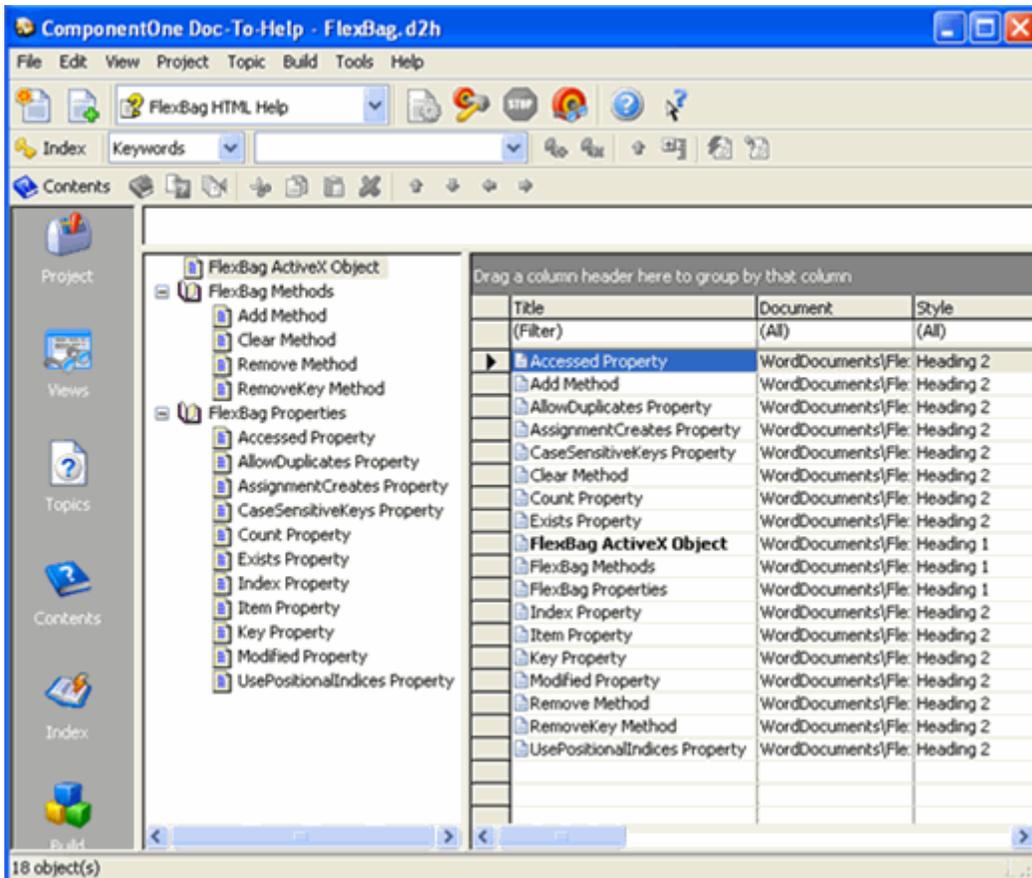
With Doc-To-Help 2006, there are two types of source documents: single topic and multiple topic. Single topic documents allow you have only one topic, while multiple topic documents allow you to have as many topics as necessary. In a multiple topic document, you define your topics using paragraph styles. A topic consists of all text and graphics beginning with the topic title and ending just before the next active paragraph style. The end of file marker terminates the last topic in a source document.

In a single topic document, the topic is defined by the document's [Title](#) and [Style](#) properties. This tells Doc-To-Help the title and position of the topic in the table of contents hierarchy.

For additional information on the types of source documents, see [Doc-To-Help 2006 Source Documents](#) (page 181).

Organizing the Help Contents

Doc-To-Help automatically generates a contents file (or HTML equivalent) based upon the outline level (1 through 9) of the paragraph styles that define topic boundaries for a Doc-To-Help project. The result is similar to the document map in Microsoft Word. The following figure shows an equivalent Help contents view within the Doc-To-Help authoring environment.



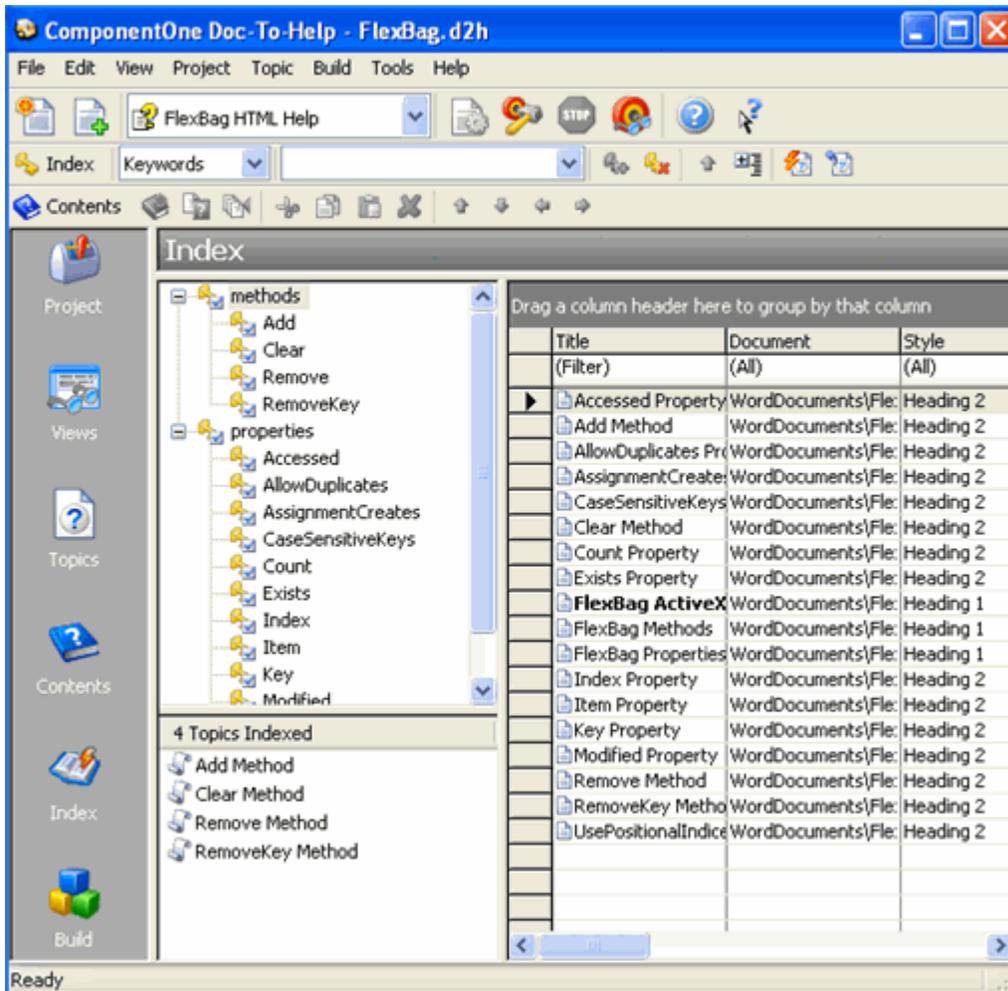
For more information, see [Modifying the Help Contents](#) (page 291).

Constructing the Index

In Windows Help, the Index tab displays an alphabetized list of keywords, each of which is associated with one or more Help topics. The author makes these associations by assigning to each topic one or more K-footnotes containing the appropriate keywords. Apart from writing the actual text, this process is widely regarded as the most tedious and painstaking aspect of Help authoring, particularly when done well.

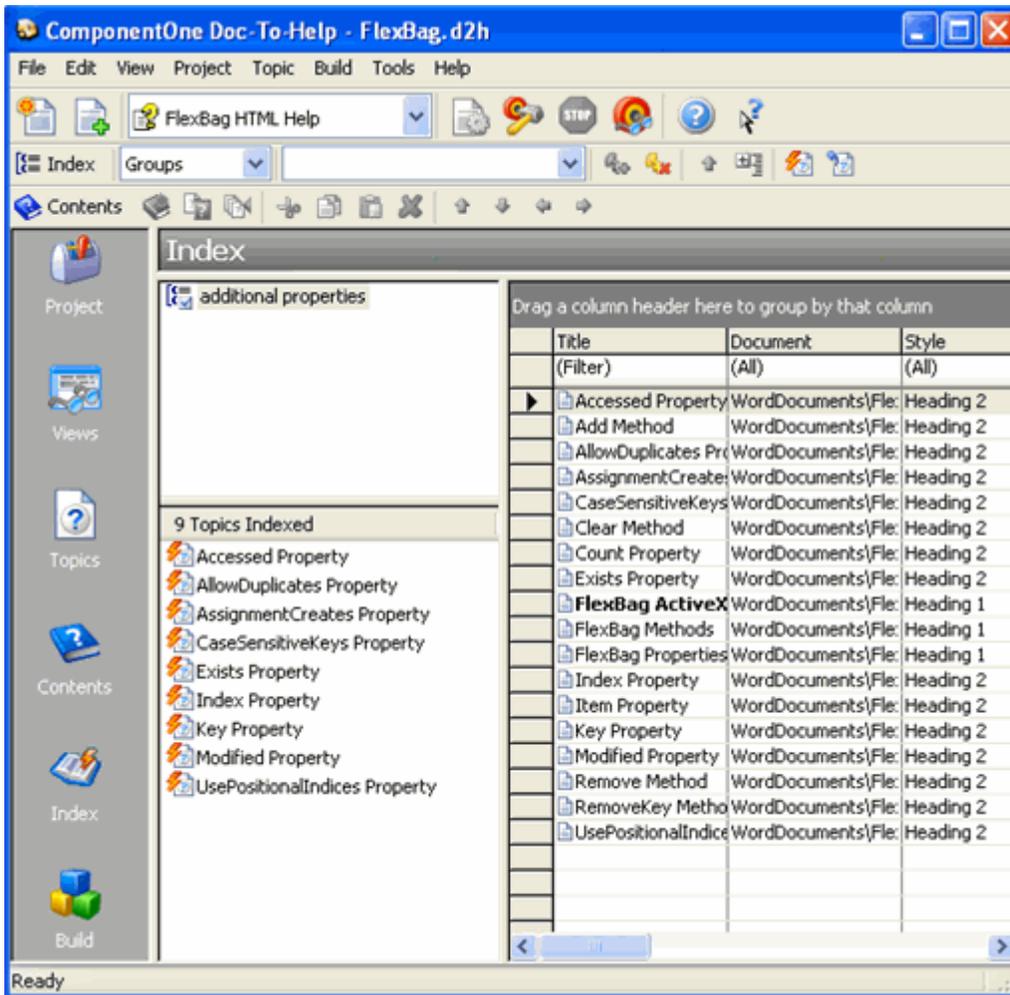
Doc-To-Help does more than just wrap a fancy interface around a tedious process. It *automates* the process by adding indexing behavior to styles, and gives you even more control by exposing an intuitive object model for scripting. For example, you can write small modules to analyze the titles of reference topics and generate several index entries for each automatically. As new topics are added, they are indexed by your script code automatically at compile time!

You can also add index keywords and associate them with topics manually using a Windows Explorer-like interface, as shown in the following figure.



In addition to index keywords that are visible to the reader, Doc-To-Help supports associative indexing via named topic groups that are hidden from the reader. If you are an experienced Windows Help author, you will recognize these as ALinks. For example, you can use groups to categorize topics according to the reader's experience level, then implement hot spots for novice, intermediate, and advanced topics.

The interface for creating and assigning groups is virtually identical to the one used to build the visible keyword index.



For more information, see [Building an Index](#) (page 341).

Specifying Navigational Elements

In multiple topic documents, Doc-To-Help derives the topic navigation sequence from the physical order of topics within the documents. In a single topic document, the topic navigation sequence is derived from the document's [Style](#) property.

With both document types, you can exclude topics defined with a particular paragraph style or [Style](#) property from the navigation sequence. For example, suppose that a multiple topic source document contains a set of function descriptions at the *Heading 2* level, each of which is immediately followed by a code example at the *Heading 3* level. If each function description provides an explicit hyperlink to its code sample, and *Heading 3* is excluded from the navigation sequence, then the reader can step through the function descriptions and bypass code samples that are not of interest. If you have a single topic source document in the same project with a [Style](#) property set to *Heading 3*, and *Heading 3* is excluded from the navigation sequence, this topic is skipped.

Doc-To-Help provides another form of automatic topic navigation that takes advantage of subtopic relationships defined by paragraph styles. For styles having an outline level of 2 or greater, you can opt to display a list of subtopic buttons at the end of each parent topic. In a single topic document, the [Style](#) property must be set to *Heading 2* or greater.

For more information, see [Providing Links to Subtopics](#) (page 285).

Linking Related Topics

In Windows Help, the author implements inter-topic links by manually assigning a unique string known as a topic ID to each destination topic and then citing the topic ID within specially formatted text that defines a hot spot. Topic IDs are specified as #-footnotes but are referenced as hidden text. The reader never sees the topic ID; it is used only by the Help compiler as a destination for a jump (or pop-up) hot spot, which consists of double-underlined (or single-underlined) text followed by a topic ID formatted as hidden text. If you have ever authored a Windows Help file, you are well aware that linking topics in this manner is a time consuming and error prone process.

Doc-To-Help offers several methods for implementing hot spots. The preferred method of defining links is through D2HML. D2HML allows you to create topic links, in addition to many other types of links, quickly and easily in both Word and HTML documents. For more information, see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233). Another method you can use in Word source documents is using the **Add Topic Link** and **Add Dynamic Link** commands on the **Doc-To-Help** toolbar. Using these buttons, you can define links by simply highlighting the text where you want a hot spot and selecting the target from a convenient dialog box. For more information, see [Links and Hot Spots](#) (page 309).

Note: Doc-To-Help uses the term *link tag* instead of the WinHelp term *topic ID* to describe a unique string that identifies a topic. This term was introduced to distinguish these strings from the unique numeric topic ID assigned by Doc-To-Help during compilation.

Help Targets in Doc-To-Help

ComponentOne Doc-To-Help 2006 allows you to create a wide range of outputs, called “targets” from a single project.

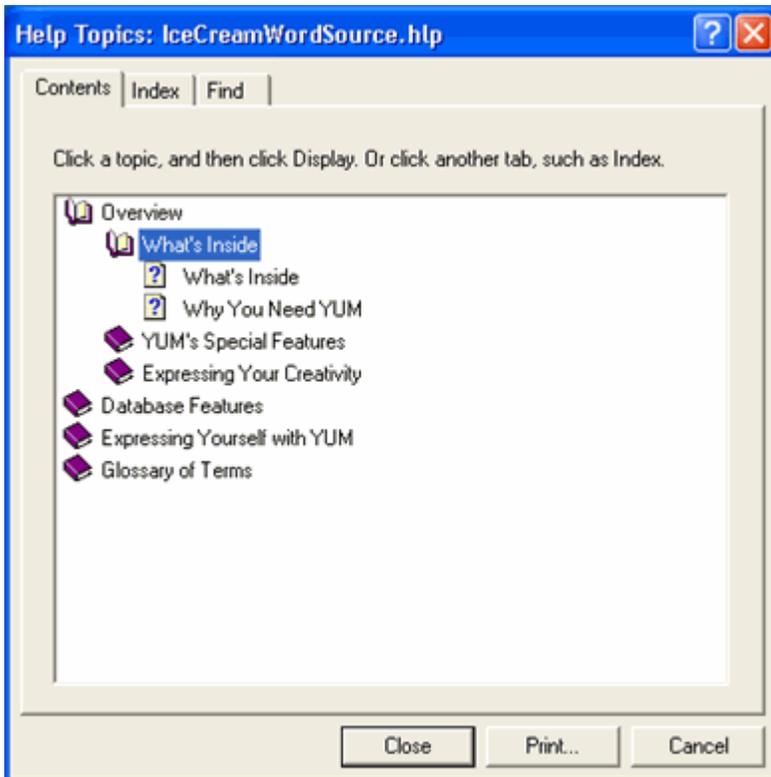
WinHelp Target

In 1995, Microsoft released WinHelp 4.0 as the online Help standard for desktop applications for the Windows operating system. It offered a sleek new look (for its day), as well as a host of navigation and access features. While WinHelp’s grip on the Help market is diminishing, its feature set provides a baseline for comparison among all the available Help targets.

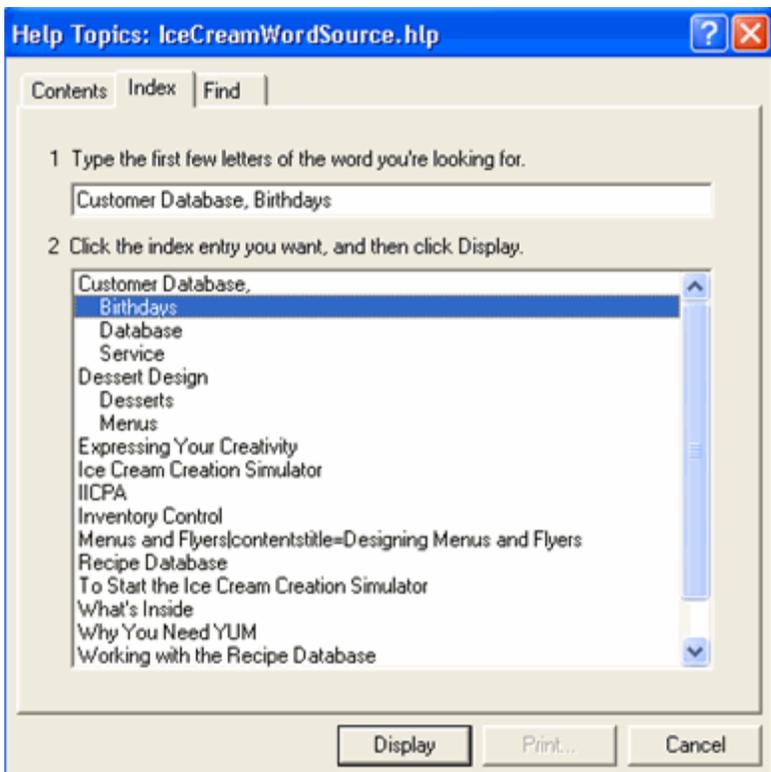
WinHelp 4.0 introduced a model where major navigational components reside separately from content. The contents, index, and search functions display in a separate, somewhat annoying modal dialog box, the Help Topics dialog box. The major weakness inherent in this display model is the lack of ease in switching back and forth between the Help Topics dialog box and the WinHelp viewer—between navigation and content.

The Help Topics dialog box supports three standard tabs: Contents, Index, and Find (full-text search).

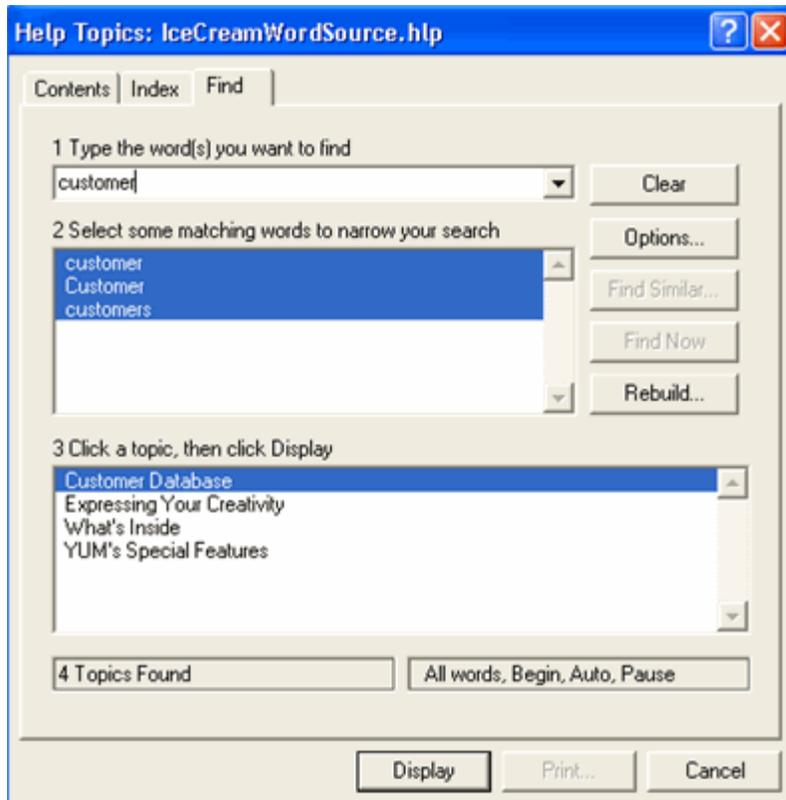
The Contents tab replaces the Contents topic in earlier WinHelp systems, and it displays the Help contents in an expandable tree format, containing books and pages. Books divide the content into a logical structure, while pages link to specific topics.



WinHelp 4.0 also introduced true second-level keyword entries. Help authors gained the functionality required to create a proper index two levels deep—an index that more closely resembles a back-of-the-book index.

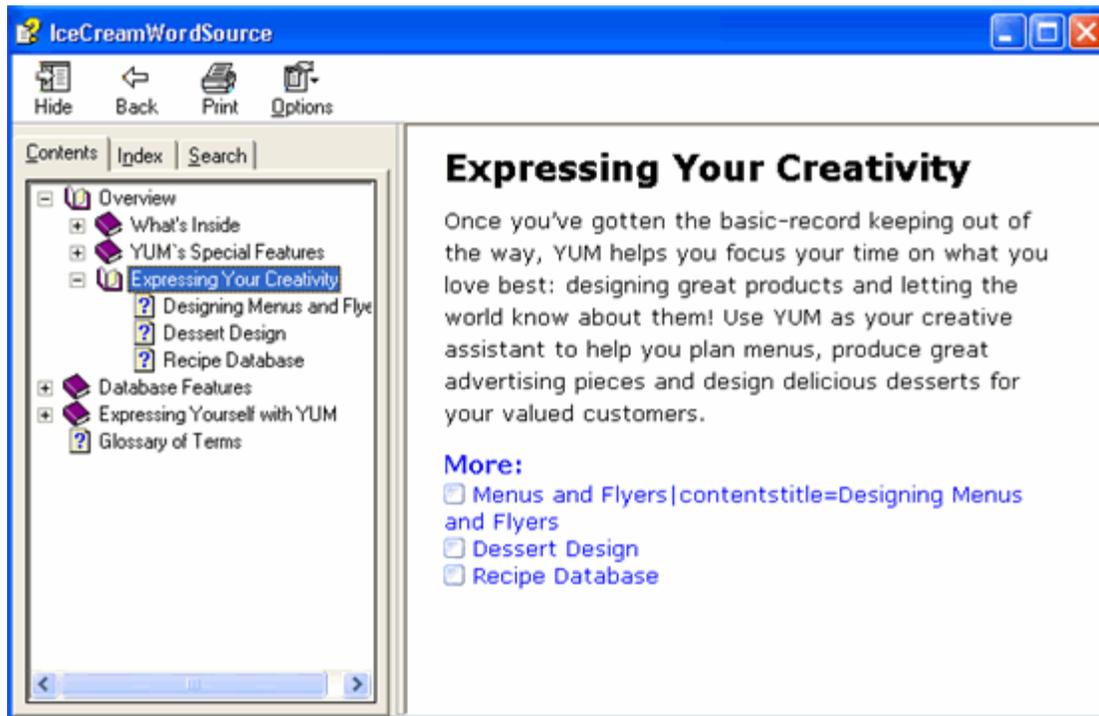


The Find tab supports full-text search capability. With no stop list to filter noise, the full-text search is a less powerful search function than a well-constructed Index. In addition, the full-text search shows every occurrence of a word, it is not selective; and it is literal. It only shows the words that exist. In order to find something, you must know what words the author used – not what you might use.



Microsoft HTML Help 1.x Target

In 1996, Microsoft announced that HTML Help would replace WinHelp as the standard desktop online Help delivery mechanism, a radical departure from the WinHelp .rtf format. With its debut in 1997, HTML Help became the first widely accepted online Help system to use HTML as its underlying language.



HTML Help 1.x builds on the popular feature set in WinHelp 4.0, and improves upon it, providing the author greater flexibility and additional opportunities for customization.

HTML Help introduced the tripane viewer, which combines navigation and content in one location. This tripane model has since become the standard for most HTML-based Help technologies. In addition to the familiar Contents, Index, and Search tabs, HTML Help supports a Favorites tab, allowing users to add topics for ready reference.

The second-generation, auto-synchronizing Help contents, the multi-level Index, the advanced full-text search function, and the Favorites options improve on the WinHelp model.

JavaHelp Target

JavaHelp software was developed to provide a standard Help solution for pure Java applications. JavaHelp software was released in April 1999, by Sun Microsystems, and is currently in release 2.0. While there is no “standard” viewer, JavaHelp uses components from the HotJava browser for its display. To view a JavaHelp HelpSet, the Java Run time Environment (JRE) or the Java Development Kit (JDK) must reside on the computer in addition to JavaHelp.

Doc-To-Help 2006 has improved support for the JavaHelp target. JavaHelp versions starting from 1.1.3 are supported, including JavaHelp 2.0.

- JavaHelp secondary and pop-up windows are now supported. Topics designated for display in a secondary or pop-up window are now shown in a JavaHelp secondary or pop-up window. Previously, all topics were shown in the main window.
- This version fully supports fonts in JavaHelp. JavaHelp has known problems with font support. Therefore, help authors who used other tools or authored JavaHelp manually were usually limited to using just one default font for the entire text. Now Doc-To-Help authors can make JavaHelp display fonts properly.

Font support is controlled by three new properties in Help Target (JavaHelp): **IgnoreFontNames**, **IgnoreFontSizes** and **ScaleFontSizes**. By default, IgnoreFontNames = True and IgnoreFontSizes = True. It means that fonts specified in the source document are ignored, default JavaHelp fonts are used instead (default font for text, special default fonts for headings). This is the same behavior as in previous versions. If both **IgnoreFontNames** and **IgnoreFontSizes** properties are set to False, JavaHelp will use fonts exactly as they are defined in the source document. This is normally what authors want, but the matter is slightly complicated because of a known JavaHelp bug that makes all fonts appear smaller (approximately 1.3 times smaller) than they should be. Doc-To-Help provides two alternative ways of rectifying this. Setting **IgnoreFontSizes** to True will allow you to have consistent default JavaHelp font sizes while retaining proper font families (font names). Alternatively, you can set the **ScaleFontSizes** property to a real number other than the default 1.0 that will scale all their fonts to adjust their sizes. For example, if you see that JavaHelp displays fonts 1.3 times smaller than they should be, you can set ScaleFontSizes to 1.3 to fix that.

- All colors are now supported in JavaHelp themes. Previously, only basic colors were supported.
- Help window caption can now be specified using the new Caption property in Help Target (JavaHelp).
- Default topic, the topic that is shown when opening help, is now properly supported.
- The size of the main and secondary windows can now be specified in the Windows node of the Project tab using the Left, Top, Width, Height properties or the Size Tool invoked from the context menu.
- In JavaHelp themes in Theme Designer, it is now possible to hide the navigation bar (Next and Previous commands) by selecting Position = None in Navigation Bar\Layout.
- JavaHelp tabs Contents, Index, Search, Favorites can be shown/hidden by setting new properties in the Help Target (JavaHelp): ContentsTab, IndexTab, SearchTab and FavoritesTab. Tooltip strings for these tabs can be customized\localized using the properties StringContents, StringIndex, StringSearch and StringFavorites. Please note that Favorites tab is available only in JavaHelp 2.0; it is not supported in JavaHelp 1.1.3.

To build and view JavaHelp, you must first install the necessary files from Sun. The JavaHelp folder on the Doc-To-Help CD contains the JRE (Java Runtime Environment) and the JavaHelp 1.1.3 distribution. To install JavaHelp, do the following:

1. Run j2re-1_4_1_02-windows-i586.exe to install the JRE.

2. Unzip javahelp-1_1_3.zip into any directory. Root folders such as C:\ are acceptable.
3. Set the environment variable JAVAHELP_HOME based on the folder used in the previous step. For example, if you unzipped into the root folder on the C: drive, then the variable value should be:

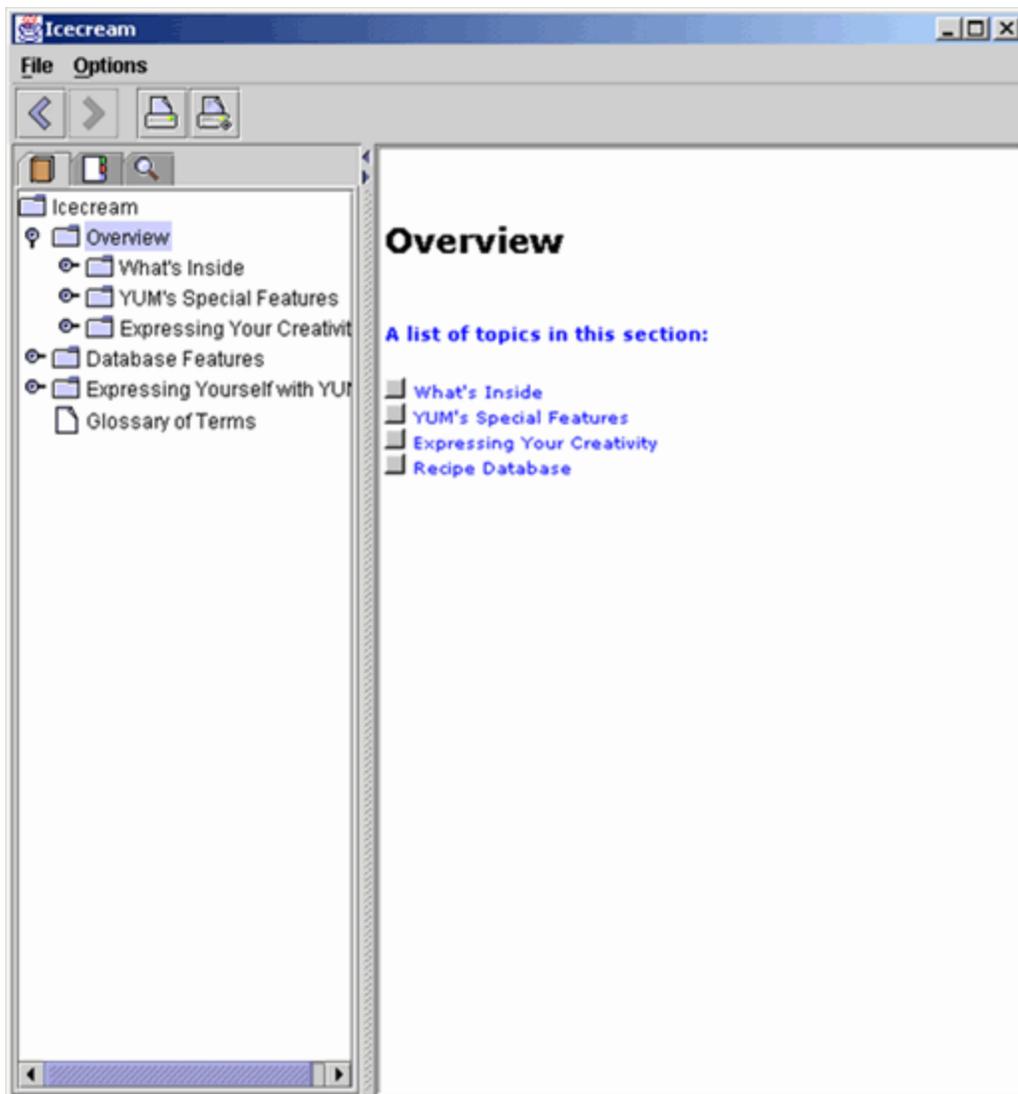
C:\jh1.1.3

Note: Do not include a trailing backslash.

4. Make sure to include java.exe in your path. If you are unable to launch the JavaHelp viewer from Doc-To-Help, this is most likely the reason.

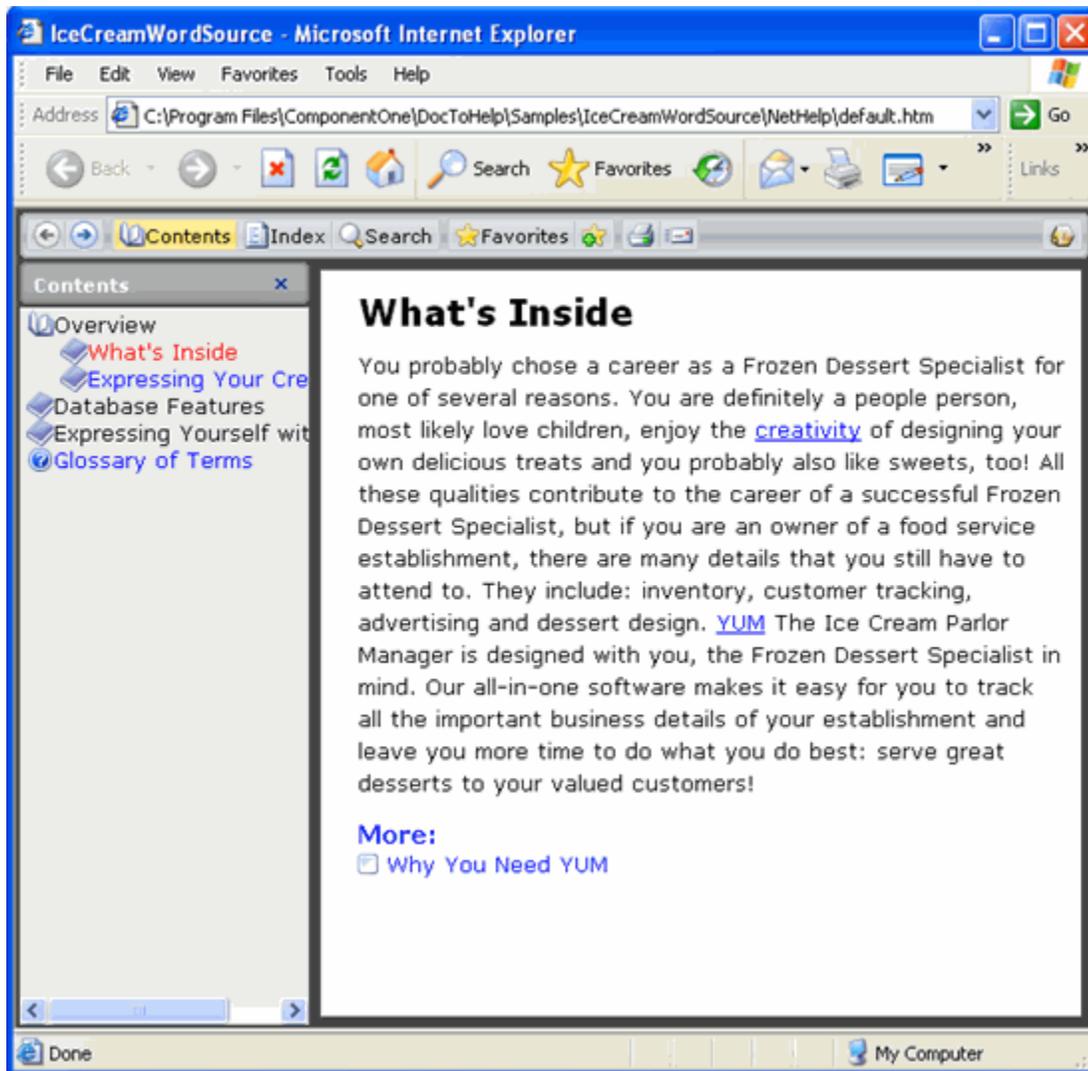
For more information, see Sun's JavaHelp Web site at: <http://java.sun.com/products/javahelp/>

JavaHelp uses the three-pane viewer model, combining navigational components, viewer pane, and buttons in the viewer. Navigational components include contents, index, and search. Unlike the HTML Help standard viewer, the HelpSet viewer is a demo viewer, not a standard viewer. JavaHelp is fully extensible and customizable, should a programmer wish to take advantage of this flexibility. In practice, however, the HelpSet viewer has become the de facto standard viewer.



NetHelp Target

NetHelp, known in previous versions of Doc-To-Help as HTML 4.0, has been greatly enhanced in Doc-To-Help 2006. The NetHelp target is a one-size-fits-all solution for providing online Help for Web-based applications. NetHelp Help systems work across all modern browsers and operating systems.



The NetHelp target can be viewed with all browsers supporting Dynamic HTML. The navigation conforms to the three-pane Help standard, simultaneously displaying basic navigational components in the left frame and content in the right frame. The browser buttons provide additional basic navigation.

NetHelp has full cross-browser support. Previously, HTML 4.0 supported Dynamic HTML, including expanding and collapsing table of contents (TOC) nodes and pop-ups, only in Internet Explorer. The other popular browsers that are now supported are Netscape, Mozilla, Firefox, and Opera.

By default, NetHelp automatically synchronizes the TOC with the current topic; therefore, when a user jumps to a topic, that topic automatically becomes the current topic in the TOC. If you prefer not to synchronize the TOC automatically, you can add a Sync TOC button to your theme using the Theme Designer. The **sync toc** button will appear in the TOC, which can be clicked to synchronize the topic appearing in the right pane with the TOC, and automatic TOC synchronization is disabled. This synchronization was not supported by HTML 4.0 in previous versions of Doc-To-Help.

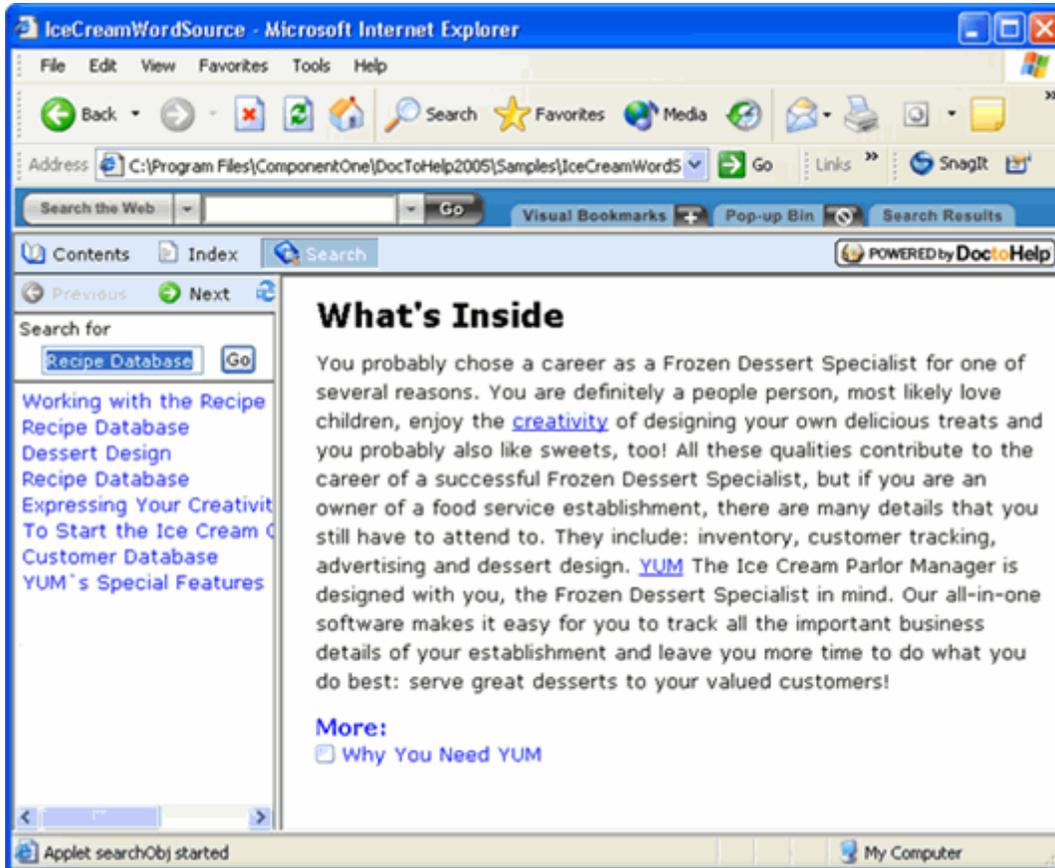
NetHelp themes have a **Favorites** button, by default, which, when clicked, shows the **Favorites** pane where you can organize all of your favorite topics. Once in the **Favorites** pane, you can use the default **Add to Favorites** button to

add the current topic to the list, or click the **Delete** link to remove the topic. The appearance of the **Favorites** pane and buttons can be customized or disabled through the [Theme Designer](#) (page 572).

You can disable the **Favorites** pane by setting the HelpTarget.[FavoritesTab](#) property to **False**.

Note: Between browser sessions, **Favorites** items are stored as cookies; therefore, cookies should be enabled in the browser in order for **Favorites** to persist between browser sessions.

NetHelp also has search capability. By default, a **Search** link (or button, depending on the theme) appears at the top of the right pane, along with the **Contents**, **Index**, **Previous** and **Next** links. Clicking the **Search** link opens a search panel in the left-hand pane where the TOC and index are normally displayed. The user can simply enter text in the textbox and click the **Go** button to begin a search.



NetHelp search uses a Java servlet or a Java applet, depending on whether the help is deployed on a server or locally in the client file system. If NetHelp is deployed on a server, it must be deployed on a Web server that supports Java servlets or IIS. If it is deployed on a client machine, the client browser must be Java-enabled.

Additions have also been made to the Theme Designer to allow customization of the new Search and Synchronize TOC features, along with a Message area used for displaying error messages in the NetHelp target. For additional information on these features, see [Using the Theme Designer](#) (page 572).

Deploying and Using NetHelp

NetHelp can be deployed either locally on a single machine or on a server for use in any client browsers over the Internet or Intranet.

Local NetHelp

NetHelp used locally on a single machine (or on multiple computers accessing a share network drive) is just a folder containing files that you open in a browser. You don't need special deployment to make it accessible, just copy it to a folder where you want to keep it. However, there are two requirements that you should be aware of:

Windows XP SP2 security block

If you view NetHelp in Internet Explorer on Windows XP SP2 and higher, you may encounter the problem that your active content (JavaScript) is disabled. In that case, NetHelp will not have the proper layout and won't respond properly to user interaction. This situation is indicated by a light-yellow bar under the Address box. This is a general problem for viewing all HTML contents in local file, not Doc-To-Help-specific. Blocking local file content has caused many user complaints. And indeed, local content is usually considered more safe than content in the Internet zone. To allow active contents (JavaScript) to run, click the info bar (the light-yellow strip below the Address box) and select **Allow Blocked Content** from the menu. To disable this security block for all local content, check the **Allow local content to run in files on My Computer** check box in Tools | Internet options | Advanced | Security in Internet Explorer.

Java must be enabled for NetHelp Search

Local NetHelp Search requires Java to be enabled in the browser. Both Sun and Microsoft Java VM are supported, so it does not matter what Java is installed on your computer, what matters is only that Java is enabled in your browser. If, for some reason, you do not want to enable Java, you can still use NetHelp without Search functionality. If you need to remove the Search button from NetHelp, build NetHelp with `HelpTarget(NetHelp).SearchTab` property set to `False`. Note that Java is required for NetHelp Search only when NetHelp is deployed locally in files on your machine. Java is not used in NetHelp at all on client machines if NetHelp is deployed in a Web server.

Server-based NetHelp

NetHelp can be deployed in any Web server and viewed in client browsers over the Internet or Intranet. There are no additional requirements for NetHelp on the server except for NetHelp Search functionality. NetHelp Search uses Java, so it requires Sun JRE installed on the server machine (Microsoft JVM is not supported, because Microsoft is ending support of MSJVM). For the same reason, to enable Search, NetHelp must be deployed either on a Web server supporting Java servlets (Tomcat, Resin, etc) or in Microsoft IIS (these two categories of Web servers cover all popular general use Web servers).

Note that Java is not needed on user machines to view server-based NetHelp in user browsers.

Also note that Windows XP SP2 security block described in [Local NetHelp](#) (page 34) does not apply to server-based NetHelp.

Deploying NetHelp in a Java servlet-enabled Web server

Deploying NetHelp in any popular Java-enabled Web server (such as Tomcat, Resin, etc) does not require any special steps except the standard procedure of publishing Web pages on the server. Just follow your Web server instructions for publishing content. For example, suppose generated NetHelp is in a `... \MyProject \NetHelp` folder and you want to publish it in Jakarta Tomcat. Suppose your Tomcat folder is `... \jakarta-tomcat-5`. Copy the NetHelp folder to `... \jakarta-tomcat-5 \webapps`. Start Tomcat, run `... \jakarta-tomcat-5 \startup.bat`. You can immediately see NetHelp in your browser. For example, on the same server machine you can navigate your browser to `http://localhost:8080/NetHelp/default.htm`

Deploying NetHelp in Microsoft IIS

To deploy NetHelp in Microsoft IIS you need to create an IIS virtual directory for your NetHelp (the standard procedure for publishing any content in IIS) and perform a few simple additional steps (these steps are not needed if you don't need Search functionality in NetHelp):

1. As with any server-based NetHelp deployment, Sun Java Runtime Environment (JRE) must be installed on your server machine.

2. Copy C1D2HASPHandler.dll found in the NetHelp subdirectory of the Doc-To-Help installation directory to the server machine (does not matter to what directory, the directory where NetHelp is deployed is a good choice) and register it with

```
regsvr32 C1D2HASPHandler.dll
```

3. Modify the platform.js file in the NetHelp directory. By default this file contains

```
var d2hServerPlatform = "jsp";
```

which means that NetHelp Search uses a Java servlet. Change it to

```
var d2hServerPlatform = "asp";
```

to use ASP instead.

Using a Noise Word List for NetHelp

You can create your own text file containing the noise words to be ignored when a NetHelp Help file is searched. The [SearchStopList](#) property allows you to specify the list file to be used.

1. Create a text file (.txt) with your noise words; each word must appear on a separate line in the file. If you specify a noise word list using the [SearchStopList](#) property, the default list installed with Doc-To-Help is overridden. Therefore, you must add the standard noise words to your list if you want them to be ignored in the search. The standard noise words are:

```
a, all, am, an, and, any, are, as, at,
be, but, by, can, could, did, do, does,
etc, for, from, goes, got, had, has, have,
he, her, him, his, how, if, in, is, it,
let, me, more, much, must, my, nor, not,
now, of, off, on, or, our, own, see, set,
shall, she, should, so, some, than, that,
the, them, then, there, these, this, those,
though, to, too, us, was, way, we, what,
when, where, which, who, why, will, would,
yes, yet, you
```

2. In the Doc-To-Help project, select **NetHelp** from the Help Target drop-down list.
3. In the property pane, click the **ellipsis** button next to the [SearchStopList](#) property and locate the noise word text file you created.
4. Select the noise word file and click **OK**.
5. Click the **Make Target** button  to build the NetHelp Help file. If the word(s) entered in the **Search for** text box matches a word specified in your noise word list, it is ignored during the search process.

Section 508 Compliant NetHelp

Section 508 of the Rehabilitation Act of 1973 was amended by Congress in 1998 to require Federal agencies to make electronic and information technology accessible to Federal employees and members of the public who are seeking information from a Federal agency and have visual, auditory, motor, or other disabilities. Section 508 states that people with disabilities should have the same access to and use of information as someone without disabilities.

Doc-To-Help provides a special [Accessibility](#) property to help you create Section 508 compliant NetHelp. Setting the [Accessibility](#) property to **Section 508 Compliance** enables the following features:

- All links generated by Doc-To-Help have title strings, which indicate the link type, and appear as tooltips that are read by accessibility devices. The default title strings are: *link*, *popup*, *expanding text*, and *dropdown*

text. These strings can be changed in the [Theme Designer](#) (page 572) using the **Hot spot titles** commands in the **Accessibility** node.

- Icons in the table of contents have titles, their text equivalents, indicating whether the item is a book or a topic. If the [DynamicTOC](#) property of the NetHelp target is set to **True** and, therefore, the icon is a book, this title indicates whether it is open or closed. The strings can be changed in the [Theme Designer](#) (page 572) using the **Text equivalents for images** commands under the **Accessibility** node.
- Pop-up links become jump links to allow easier accessibility. For the same reason, margin notes and glossary term links, which usually appear as pop-up windows in normal mode, are not shown as pop-ups but as normal HTML pages in the main frame.
- Inline pop-up text is shown as inline, or expanding, text rather appearing in a pop-up box.
- The menu normally displayed when a user clicks a group or keyword link or a keyword in the index that has multiple destinations, or target topics, does not appear. Instead, the destinations are shown in the main frame as a normal HTML page. By default, the heading of this page is "N topics found", where N is the number of topics having the specified group or keyword associated with it. The heading can be changed in the [Theme Designer](#) (page 572) using the **Other strings** commands under the **Accessibility** node. Specify the text under **Keyword/group link found multiple topics**.

When the [Accessibility](#) property is set to **Normal**, the section 508 compliance features are disabled.

Regardless of the value of the **Accessibility** property, all links and buttons in NetHelp themes are accessible from the keyboard using the **Tab** key. Additionally, when the [DynamicTOC](#) property is set to **True** and a target is built, the user can expand and collapse books in the table of contents using the **Num +/-** buttons on the keyboard.

Each frame of a NetHelp theme also has a title string that can be read by accessibility devices, even when [Accessibility](#) is set to **Normal**. By default, the title strings match the frame titles. These strings can be changed in the [Theme Designer](#) (page 572) using the **Other strings** commands in the **Accessibility** node. The default strings are: *Topic navigation*, *Top topic navigation*, *Bottom topic navigation*, *Navigation panes*, *Topic text*, *Index lookup pane*, *Index list*, *Search lookup pane* and *Search result list*.

For additional information on Section 508, visit <http://www.section508.gov>.

Accessibility in HTML Source Documents

While accessibility for items generated by Doc-To-Help, such as pop-up links, pop-up text and table of contents icons, is automatically enabled when [Accessibility](#) is set to **Section 508 compliance**, you must make sure your own content, or any content that is not generated by Doc-To-Help, is also Section 508 compliant. We recommend you read from the wide variety of guidelines and informative materials available about how to make sure your HTML content is accessible. There are also many tools available, such as Macromedia Dreamweaver's accessibility validation feature, to check accessibility in your HTML source documents. There are various third-party tools for checking accessibility, as well.

Accessibility in Word Source Documents

When building a NetHelp target, Doc-To-Help uses your Microsoft Word source documents to produce HTML output. Most of the generated HTML is accessible by accessibility devices, but there are a few exceptions; images without alternative text and tables without captions and summaries are not accessible.

There are no third-party tools to check accessibility in Word source documents as there are for HTML source documents; however, Doc-To-Help helps to ensure your content is accessible by providing warnings when the NetHelp target is built and images without alternative text or tables that are missing a caption and summary are found.

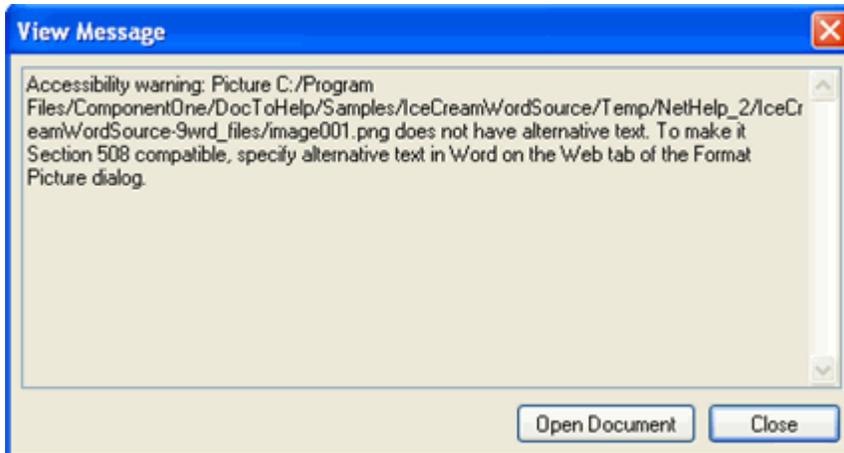
Alternative Text for Images

Images within your Word source documents must have alternative text, or a description that can be read by accessibility devices for visually impaired users. To specify this text:

1. Right-click an image and select **Format Picture**.

2. Click the **Web** tab in the **Format Picture** dialog box.
3. Enter the text in the **Alternative text** textbox and click **OK**.

If the [Accessibility](#) property is set to **Section 508 Compliance**, Doc-To-Help provides warnings in the build log when images without alternative text are encountered.



These warnings can be ignored if you know that certain images do not need to have alternative text; this will not interfere with accessibility. Doc-To-Help inserts an empty alternative text tag in the NetHelp target if text is not provided.

Table Captions and Summaries

Tables must have a caption and summary specified in HTML in order to be Section 508 compliant. The caption is specified using a special **<caption>** tag which must appear within the **<table>** tag. It is displayed in the target and read by assistive devices as information pertaining to the table.

To specify the caption in your Word source documents:

1. Select and right-clicking an entire table.
2. Choose **Caption** from the context menu.
3. Enter the text for the table caption in the **Caption** textbox and click **OK**.

Word-generated HTML presents the caption as normal text with special formatting, not as a **<caption>** tag inside of a **<table>** tag. When [Accessibility](#) is set to **Section 508 Compliance**, Doc-To-Help corrects this problem, placing the **<caption>** tag inside the **<table>** tag, so that assistive devices can see this caption belongs to the table.

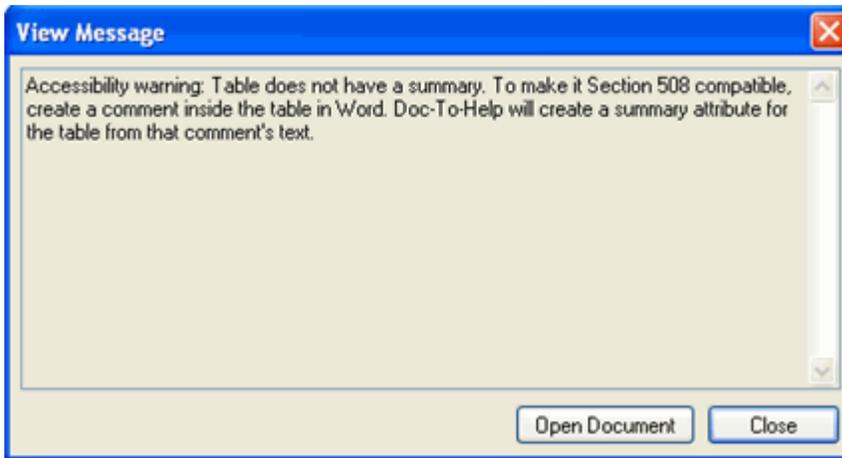
The summary describes the table's purpose and usually provides more detail than the caption. It is specified in the *summary* attribute of the **<table>** tag. It does not appear in the target, but it is read by assistive devices.

Although there is no native Word feature for specifying a table summary, Doc-To-Help makes it possible to do so. When **Section 508 Compliance** is enabled and the NetHelp target is built, Doc-To-Help interprets any comment inside the table as the table's summary. So, to specify a table summary, simply insert a comment inside the table.

To insert a Word comment into a table:

1. Select the table or place your cursor within a cell in the table.
2. Choose **Insert | Comment**.
3. Enter the desired summary.

Doc-To-Help provides warnings in the build log for every table that does not have a caption or a summary.

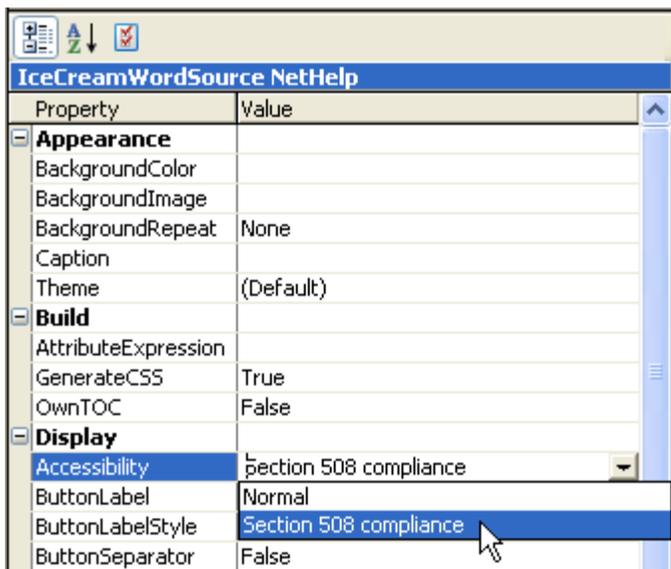


These warnings can be ignored if you know a certain table does not need to have a caption and summary; this will not interfere with accessibility. Doc-To-Help always inserts an empty caption and summary in the NetHelp target if they are not provided.

Creating Section 508 Compliant NetHelp

Once you have created accessible content in your Word or HTML source documents, use the following steps to build a Section 508 compliant NetHelp target:

1. On the **Icon** bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Select a **NetHelp** target in the right pane.
4. In the **Property** pane, click the drop-down arrow next to the [Accessibility](#) property and select **Section 508 compliance**.



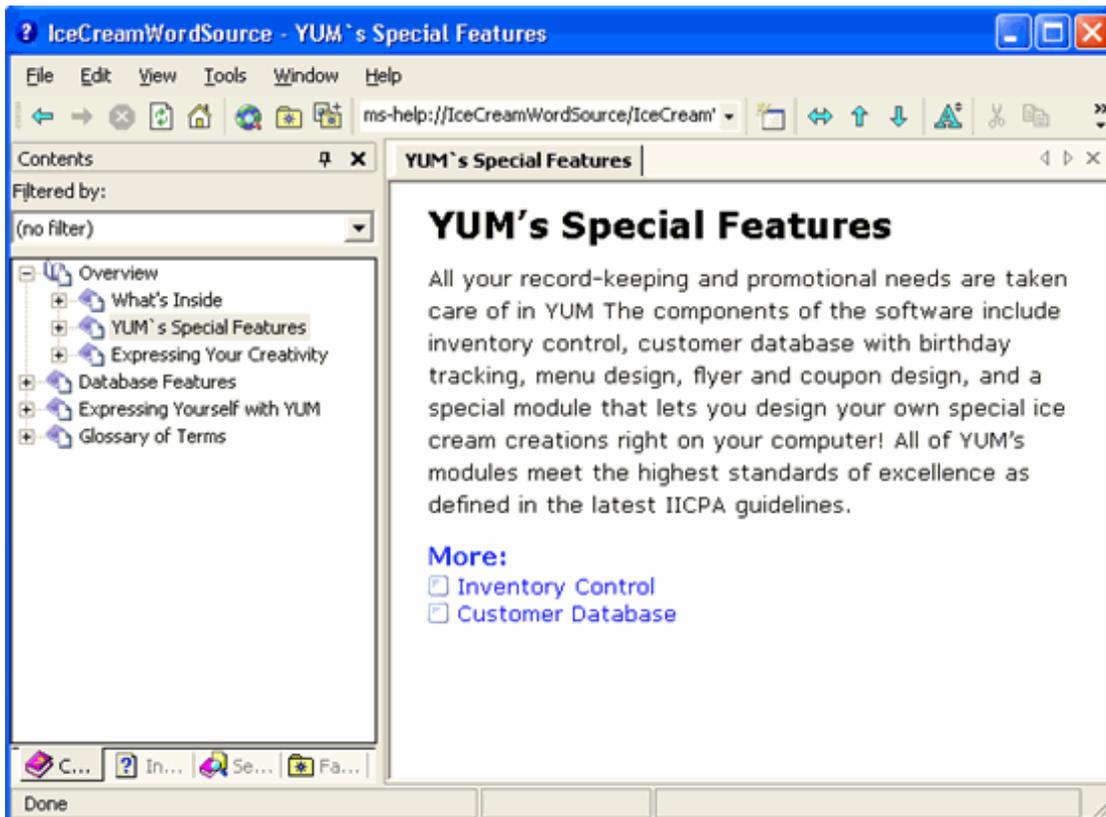
5. Click the **Make Target**  button.

Note: See [Accessibility in Word Source Documents](#) (page 36) and [Accessibility in HTML Source Documents](#) (page 36) for more information on authoring accessible content in your source documents.

MS Help 2.0 Target

Microsoft Help 2.0 closely resembles the Microsoft MSDN collection, and uses some of the same underlying technology. Programmers who distribute .NET components can create Help 2.0, because the .NET tools include the run-time components necessary to view Help 2.0 Help systems.

Note: The development of Microsoft Help 2.0 has been discontinued by Microsoft, though it is still the Help authoring format for .NET developers.



In order to create and view .HxS, or Microsoft Help 2.0, files, you must install Visual Studio.NET and VSHIK (Visual Studio Help Integration Kit). To build Help 2.0, Doc-To-Help also needs to know the locations of the Help 2.0 executable files. These locations can be specified in Doc-To-Help in the **Tools | Options** dialog box. By default, Doc-To-Help assumes the following locations:

- C:\Program Files\Microsoft Help 2.0 SDK\HxComp.exe
- C:\Program Files\Microsoft Help 2.0 SDK\HxReg.exe
- C:\Program Files\Common Files\Microsoft Shared\Help\dexplore.exe

If these paths are not correct, Doc-To-Help will prompt you to specify the correct paths in the **Options** dialog box when you build Help 2.0.

Printed Manual Target

Doc-To-Help 2006 allows you to produce a Microsoft Word master document as a printed manual target. If unmodified, the master document includes:

- The content from your source documents.
- A default table of contents based on your heading titles.
- A default title page.

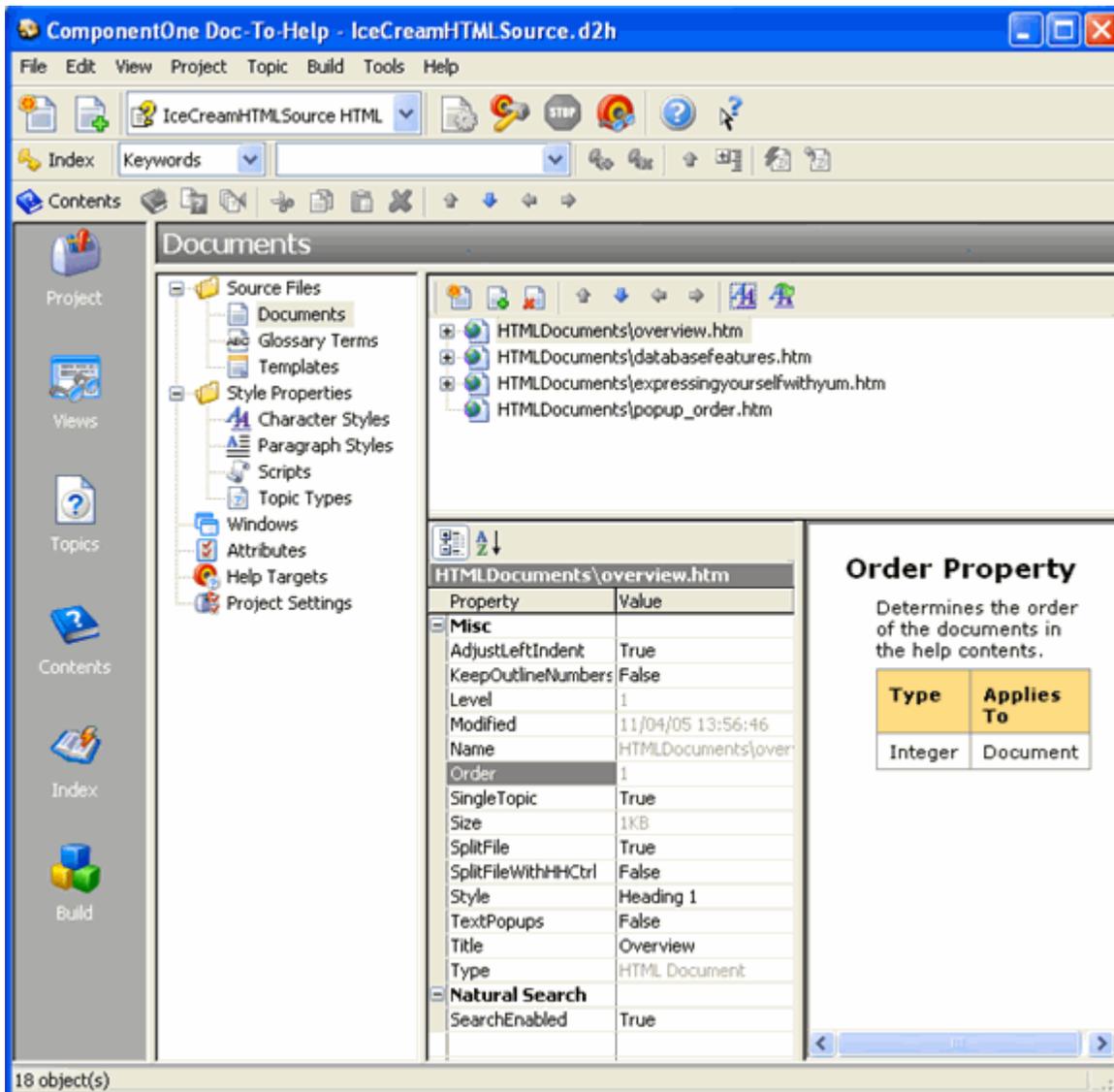
- A default index.

The printed manual target may also be used to produce a PDF, not only to be printed, but to be posted on Web sites or distributed for viewing. See [Creating a PDF](#) (page 127) for more information on producing a PDF file.

The Live Links feature of the manual target allows you to create live links, or links that, when the master document is converted to a PDF, become working links in the PDF. All links and cross-references can become live links in the master document. Use the LiveLinks property of the printed manual target to enable live links. For more information, see [Creating Live Links](#) (page 126).

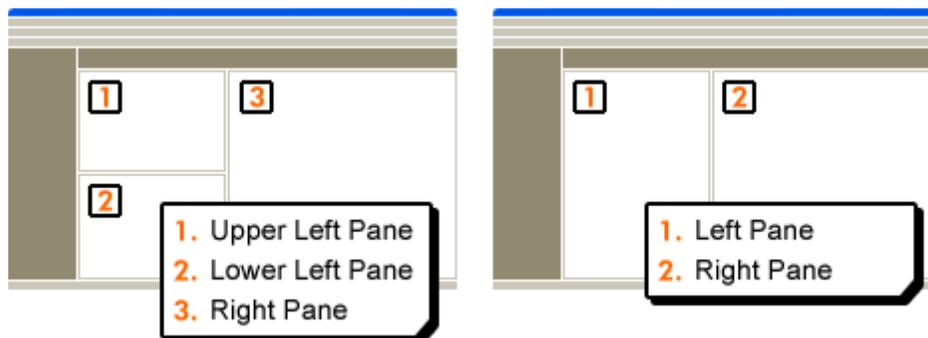
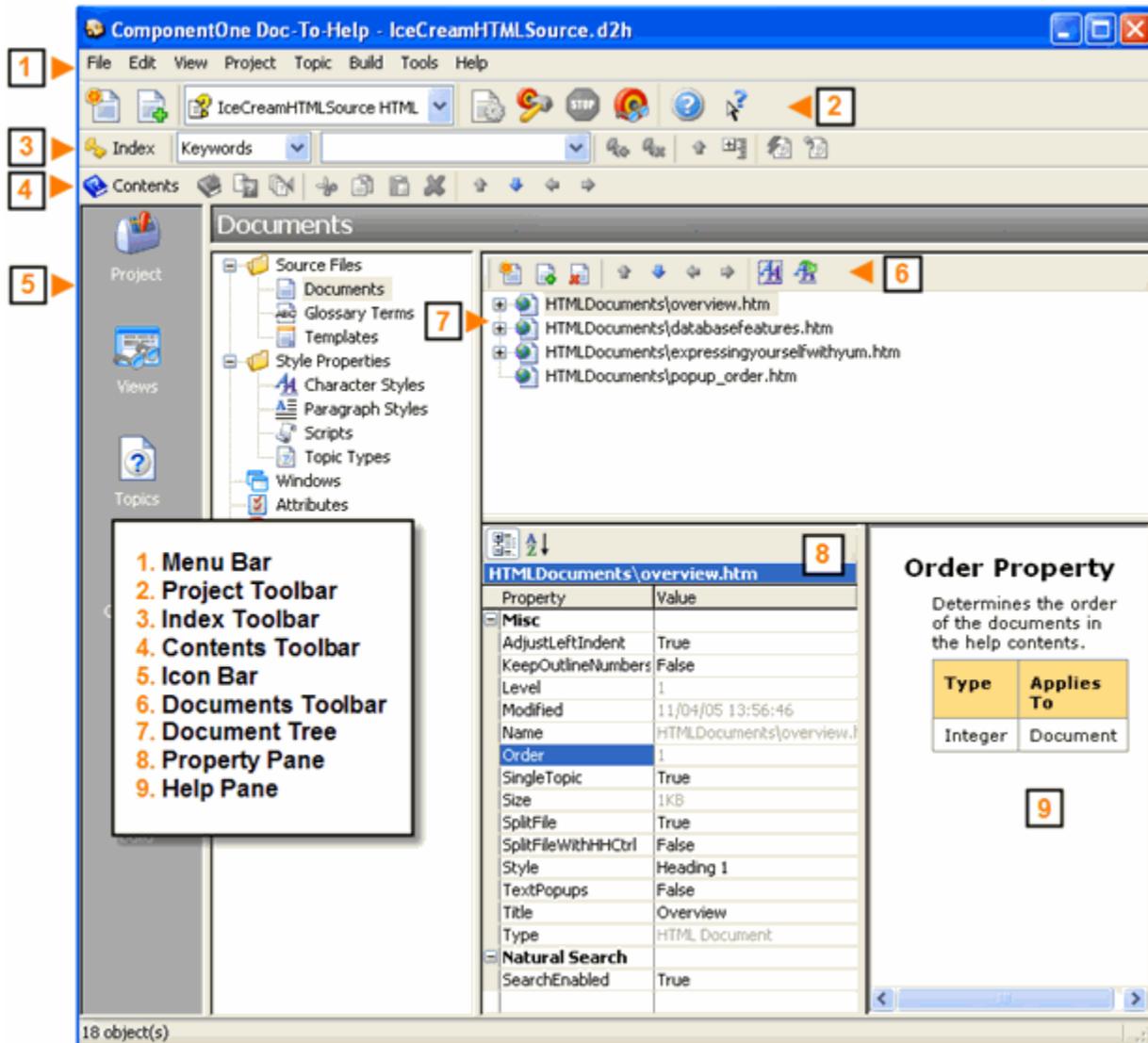
The Doc-To-Help Project Editor

This chapter provides a description of the common operations performed with the ComponentOne Doc-To-Help 2006 Project Editor. The Project Editor allows you the flexibility to perform a wide range of Help authoring tasks from within one simple user interface. Through the Project Editor, you can access and edit your source documents in Microsoft Word, FrontPage, Dreamweaver or another HTML editor and you can produce and modify the properties of a wide range of Help target types. Additionally, with the click of a mouse you can produce a finished paper manual from your Help documentation. Along with the menu items and control buttons located at the top of the Project Editor, each of the icons located in the Outlook style Icon Bar on the left of the Project Editor allows you access to various options contained within the editor.



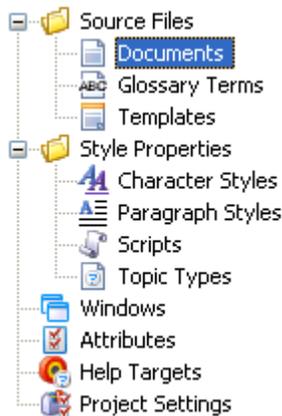
Navigating the Project Editor

The Icon Bar, located on the left side of the project editor provides the user with a simple way to navigate through the various "views" provided with Doc-To-Help 2006. As you select the icons, the window "panes" to the right of the icon bar configure to display the desired controls and information. Each configuration is designed to clearly display the information provided by that icon. The naming conventions used to describe each pane in the project editor are defined in the illustrations below.



The Project Icon

The Project Icon, as its name suggests, provides the user with an interface used to define and modify most of the features of the Help project. The left hand window pane, provides a selectable tree list of the items related to Help features, while, in most cases, the lower right hand pane provides you with information on the properties for each item.



The Project Toolbar

On the **View** menu, choose **Toolbars**, then click **Project**. A check mark next to the command indicates that the toolbar is displayed.

After a Doc-To-Help project has been compiled, you can use the **Project** toolbar to open or edit a document, choose a Help target, compile files, build or view a target and get online Help.

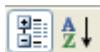


From left to right, the **Project** toolbar contains the following commands:

New Document	Opens the New Document dialog box
Add Documents	Opens the Add Documents dialog box
Help Target List	Specifies which Help target to build
Compile File	Compiles a specific Help document
Make Target	Builds the specified Help target
Stop Build	Stops the build in progress
View Target	Opens the currently compiled Help target
Help	Opens the Doc-To-Help Help file
What's This?	Activates context sensitive Help pop-ups

The Project Properties Toolbar

After a Doc-To-Help project has been compiled, you can use the **Project Properties** toolbar to view the properties by property type or alphabetically.



From left to right, the **Project Properties** toolbar contains the following commands:

View By Property Type	Displays the project properties by type
View Alphabetically	Displays the project properties alphabetically

Source Files

All of the source documents, glossary terms, templates and cascading style sheets within the project can be found under the **Source Files** folder in the project editor tree.

Documents

By highlighting the **Documents** item on the project editor tree list, you are presented with a hierarchical list of the source documents currently used by the Help project in the right pane. By choosing any one of the source documents, the lower left pane displays the property settings for that document. You can modify most properties by simply clicking in the box next to the property.

Opening or editing the documents can be achieved by highlighting the **Document** item on the project editor tree and right-clicking a specific document. New documents can be added by simply clicking in the right hand document list pane.

For more information, see [Doc-To-Help 2006 Source Documents](#) (page 181).

The Documents Toolbar

You can use the **Documents** toolbar in the right pane above the document tree to manipulate the documents in the project.



From left to right, the **Documents** toolbar contains the following commands:

New Document	Creates a new document and adds it to the project.
Add Documents	Adds a document to the project.
Remove Document	Removes the selected document from the project.
Move Up	Moves the document up within the tree.
Move Down	Moves the document down within the tree.
Move Left	Moves the document left within the tree.
Move Right	Moves the document right within the tree.
Select Documents with Incorrect Style	Check the entire tree for documents that do not have the same tree and TOC levels.
Change Document Style	Changes the style of a document.

Glossary Terms

Any glossary terms that have been added to the glossary document appear when **Glossary Terms** is clicked after the project has been compiled. The name of the glossary term is listed in the right pane, along with its type, and definition. Double-click (or right-click and select **Open**) either the name, type or definition to open the glossary document.

For more information, see [Creating a Glossary](#) (page 333).

Templates

Click the **Templates** item in the project editor tree to see a list of all templates and cascading style sheets within the project in the right pane.

Opening or editing a templates or style sheet can be achieved by right-clicking it and selecting **Open**. New templates or style sheets can be added by right-clicking anywhere in the right pane where the existing ones are listed and choosing **Add Templates**.

For more information, see [Templates and Cascading Style Sheets](#) (page 199).

Style Properties

All styles, scripts and topic types within the project can be found under the **Style Properties** folder in the project editor tree.

Character Styles

Most likely you will need to apply formatting to specific text within a paragraph. For example, you may want to add topic links, conditional text, glossary terms, or keywords to enhance your help files. Character styles allow you to format text and apply D2HML styles to create these types of hot spots and more.

Just as with paragraph styles, any character style that has special help-authoring behavior should have a matching style in the Doc-To-Help project in order for it to work correctly in your help target. Doc-To-Help provides a large variety of predefined character styles, so you do not usually need to create your own. For more information, see [Using Character Styles](#) (page 216) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233).

Paragraph Styles

In Word and HTML documents, you can use styles to apply paragraph formats (such as indentation and spacing) to give your documents a uniform appearance.

Doc-To-Help extends the concept of styles to include Help authoring behavior. For any paragraph style that has special help-authoring behavior, a like-named paragraph style must exist in the Doc-To-Help project file. Doc-To-Help comes with many predefined paragraph styles, so it is not usually necessary to create your own. You have the option, if necessary, but this is an advanced feature. By customizing the properties of Doc-To-Help styles, you define the behavior of the compiled Help system.

For more information, see [Using Paragraph Styles](#) (page 211).

Scripts

Doc-To-Help scripts are code modules written in the VBScript language that you can use to modify the behavior of paragraph and character styles during compilation. If you are familiar with event-driven languages such as Visual Basic, you can think of a script as an "event handler" for a style. Just as the Click event is fired when the user clicks a button in a Visual Basic application, a script is executed whenever Doc-To-Help encounters a topic or a hot spot defined by a scripted style.

Scripts can be added by simply clicking in the right hand script list pane. Script editing can be performed in the code text editor located to the right of the [Code](#) property.

For more information, see [Scripting Techniques](#) (page 439).

Topic Types

A topic type is a named collection of topic attributes: what window the Help topic appears in, how it's accessed, whether it gets a map number, where the heading paragraph appears, or if it appears at all. Topics are mapped to topic types by the style of their headings. The use of topic types streamlines the customizing process for the Help author, while helping maintain consistency. Adding a topic type can be achieved by right-clicking anywhere in the topic type list pane on the right side of the project editor.

For more information, see [Using Topic Types](#) (page 222).

Windows

The Help targets use the default window, named main, to specify the size, location, appearance, and user interface presented to the reader. You can change the default characteristics by editing the properties of this window.

You can also define secondary windows for displaying certain kinds of information. For example, if a topic lists a series of steps for carrying out a procedure, you may want to provide links to other topics with illustrations or additional instructions. By associating a secondary window with the character style that implements the link, you can display multiple topics simultaneously while giving the reader control over the level of detail and the layout of the Help windows. [Adding a window](#) can be achieved by right-clicking anywhere in the window list pane on the right side of the project editor.

For more information, see [Customizing Help Windows](#) (page 353).

Attributes

Attributes provide the Help author with the ability to create conditional text based upon specific Help files. In some instances, you may wish to include specific text in one group Help files while having alternate text included in a separate group of Help files. By creating attributes and associating the attribute values with Help targets, you can determine which targets contain which text. For example, you might create an attribute grouping named Audience that contains the values Novice, Expert and Administrator. In this way, you can modify the text contained in Help files for each audience by simply attaching the value to a given group of Help targets and formatting the conditional text with those attributes values. Doc-To-Help also allows you to use advanced conditional build expressions using the [AttributeExpression](#) property. You can specify any combination of attributes using the logical operators OR, AND and NOT to determine in which Help targets the text appears.

By highlighting the **Attributes** item on the project editor tree list, you are presented with a list of the default attributes in the right pane. By choosing any one of the default attributes, the lower left pane displays the property settings for that attribute.

For more information, see [Conditional Text and Attributes](#) (page 401).

Help Targets

By highlighting the **Help Targets** item on the project editor tree list, you are presented with a list of the default Help targets in the right pane. Choosing any one of the default Help targets causes the lower left pane to display the property settings for that target. You can modify most properties by simply clicking in the box next to the property. In addition to the five default Help targets, you can [create additional Help targets](#) with the right-click of your mouse.

For more information, see [Modifying the Help Contents](#) (page 291).

Project Settings

By highlighting the **Project Settings** item on the project editor tree list, you are presented with the Make and View Project commands. The lower right property pane contains the default project properties. You can modify most properties by simply clicking in the box next to the property.

For more information, see [Project Settings Properties](#) (page 163).

The Views Icon

The Views Icon provides the Help author with an “at a glance” examination of many of the key attributes related to the Help project. This, read only, informational window contains a left side tree view pane for choosing a topic and a right side display pane for attributes. The tree view consists of:

Attributes	Displays the Help targets by attribute and topics by attribute
Context IDs	Displays the topic context IDs
Groups	Displays the topic groups and topics attached to each topic group
Help Targets	Displays the current Help targets and attributes by each Help target
Keywords	Displays primary and secondary keywords and topics by keyword
LinkTags	Displays the LinkTags for each topic
Topics	Displays the topic attributes, attributes by topic, context IDs by topic, groups by topic and keywords by topic.

The Topics Icon

The Topics Icon displays an Explorer-like view of the Help topics derived from the source documents during compilation. It also provides an interface for setting properties of individual topics, such as index keywords and context-sensitive Help IDs. The left hand pane of the project editor contains a tree list allowing the author to view specific types of topics. You can view all topics, deleted topics or topics with and without Attributes, Context IDs,

Groups and Keywords. Additionally, you can sort the topics in the right project editor pane by using the convenient outlook style grouping and filtering features provided with Doc-To-Help 2006.

Doc-To-Help derives topics from your source documents. In multiple topic documents, there is not an explicit way for you to create a new topic from scratch. Instead, you create topics indirectly by applying a paragraph style to one or more source document paragraphs in Microsoft Word or your HTML editor, defining a like-named active paragraph style in your Doc-To-Help project, if one does not already exist, then compiling the affected documents either individually or by building the Help target. After compilation, you can view the newly created topics and edit their properties in the **Topics** window. With single topic documents, a new topic is created in your Doc-To-Help project when you create a new document. You can view the new topic and its properties immediately once it is created; you do not need to recompile the document.

For more information, see [Defining and Organizing Topics](#) (page 279).

The Topic Properties Toolbar

Use the **Topic Properties** toolbar to view the properties by property type or alphabetically, view topic attributes or insert [context IDs](#), [topic groups](#), [keywords](#) or [link tags](#).



From left to right, the **Project Properties** toolbar contains the following commands:

View By Property Type	Displays the topic properties by type
View Alphabetically	Displays the topic properties alphabetically
View Attributes	Displays the topic attributes , allowing specific topics to be linked to the attribute values
Topic Collections	<p>Displays the Topic Elements and their icons, which indicate how they were added - manually, or automatically from the document - and what type of elements they are. If an item is added manually, one of the following icons appears next to it:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Context ID  Groups  Keywords <input checked="" type="checkbox"/> Link Tags <p>If an item is added automatically, one of the following icons appears next to it, specifying its type:</p> <ul style="list-style-type: none">  character style  paragraph style  script  Word {XE} field
Choose Collection	Allows you to choose the collection to be modified
Element Textbox	Enter the collection element value
Add	Adds the element value
Delete	Deletes the element value

The Contents Icon

The Contents Icon displays the hierarchy of Help topics as determined by the outline level of the defining styles. This hierarchy is used to build the contents for all online Help targets. Note that the left pane contains the same books and topics as the Contents section in the compiled Help file, and the right pane displays a sorted list of topics as in the Topics window. The topics can be sorted and filtered in the same manner as in the Topic window. Doc-To-Help supports multiple topic selection by using the **Shift** and **Ctrl** keys. Simply press and hold the **Shift** or **Ctrl** keys and choose the record selector buttons to the left of the topic titles.

Doc-To-Help's Contents Window is an interface in which the final Help system's contents can be fully customized. The ability to [redefine the default topic hierarchy](#), [create new books](#), [insert and delete topics](#), and perform common editing functions is available through this icon. The interface utilizes toolbars, drag-and drop editing, and right-click shortcut menus to make the editing of contents easier and you more productive.

For more information, see [Specifying the Help Contents](#) (page 282).

The Contents Toolbar

On the **View** menu, point to **Toolbars**, and then click **Contents**. A check mark next to the command indicates that the toolbar is displayed.

After a Doc-To-Help project has been compiled, you can use the **Contents** toolbar to cut, copy, paste, delete, and rearrange contents and the topics.



From left to right, the **Contents** toolbar contains the following commands:

New Book	Adds a new book to the Help contents
Convert Topic	Changes the selected topic into a book
Insert Topics	Inserts the selected topic(s) from the right pane into the Help contents
Cut	Removes the selection and places it on the clipboard
Copy	Copies the selection to the clipboard
Paste	Inserts the contents of the clipboard at the insertion point, replacing any selection
Delete	Deletes the selection
Move Up	Moves the selection up within the outline
Move Down	Moves the selection down within the outline
Move Left	Moves the selection left within the outline
Move Right	Moves the selection right within the outline

The Index Icon

The Index Icon provides an interface for associating index keywords or named groups with specific Help topics. Index keywords are visible to the reader; only the author sees named groups. Note that the left upper pane contains the index keywords that the reader can see while the lower right pane shows which topics are indexed to the keywords. The right pane displays a sorted list of topics. The topics can be sorted and filtered in the same manner as in the Topic window.

For more information, see [Building an Index](#) (page 341).

The Index Toolbar

On the **View** menu, choose **Toolbars**, and then click **Index**. A check mark next to the command indicates that the toolbar is displayed.

After a Doc-To-Help project has been compiled, you can use the **Index** toolbar to view index collections and modify the Help project index.



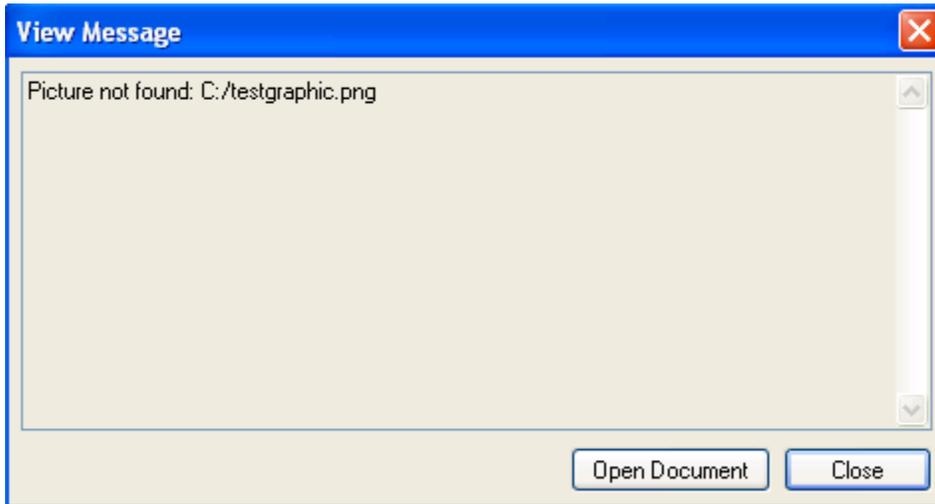
From left to right, the **Index** toolbar contains the following commands:

Index Collections	Allows you to view the groups or keywords collections.
Index Elements	Indicates which index element is being viewed currently
Add	Adds an index element
Delete	Deletes an index element
Up One Level	Moves an index element up one level
Show Secondary	Shows secondary or child index elements
Index Topics	Adds the currently highlighted topic to the current index keyword
Unindex Topics	Removes the currently highlighted topic to the current index keyword

The Build Icon

The Build Icon Window lists the files generated for the active Help target as well as any Help compiler errors or unresolved links that occurred during the previous build.

Select *Errors* or *Unresolved Links* in the left pane, and then double-click the error or unresolved link in the right pane to open the **View Message** dialog box, which lists the error message or unresolved link.



Click the **Open Document** button to open the source document at the location of the topic containing the broken link or error.

Debug output from scripts is also displayed within this window. Select *Script Output* in the left pane.

A Guided Tour of Doc-To-Help

Welcome to the ComponentOne Doc-To-Help 2006 Guided Tour! If you are an experienced Doc-To-Help user, this latest version includes many new design features bundled with the functionality you are familiar with from previous versions of this product. If you are a new user, Doc-To-Help provides an effortless way to create professional looking online Help from your existing Microsoft Word and HTML documents.

The purpose of this guided tour is to give you a "quick start" understanding of the features incorporated into the new version of Doc-To-Help. By working through the tutorial procedures, you will learn the fundamentals of producing printed manuals and online Help systems using Doc-To-Help.

The Doc-To-Help guided tour assumes a level of familiarity with Microsoft Word and FrontPage and Help authoring in general. Before beginning the guided tour, you may wish to read the sections on [Help Authoring Basics](#) (page 19) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233). Additionally, it may be helpful for you to familiarize yourself with the Doc-To-Help project editor and toolbar by reading [The Doc-To-Help Project Editor](#) (page 41).

Tutorial Conventions

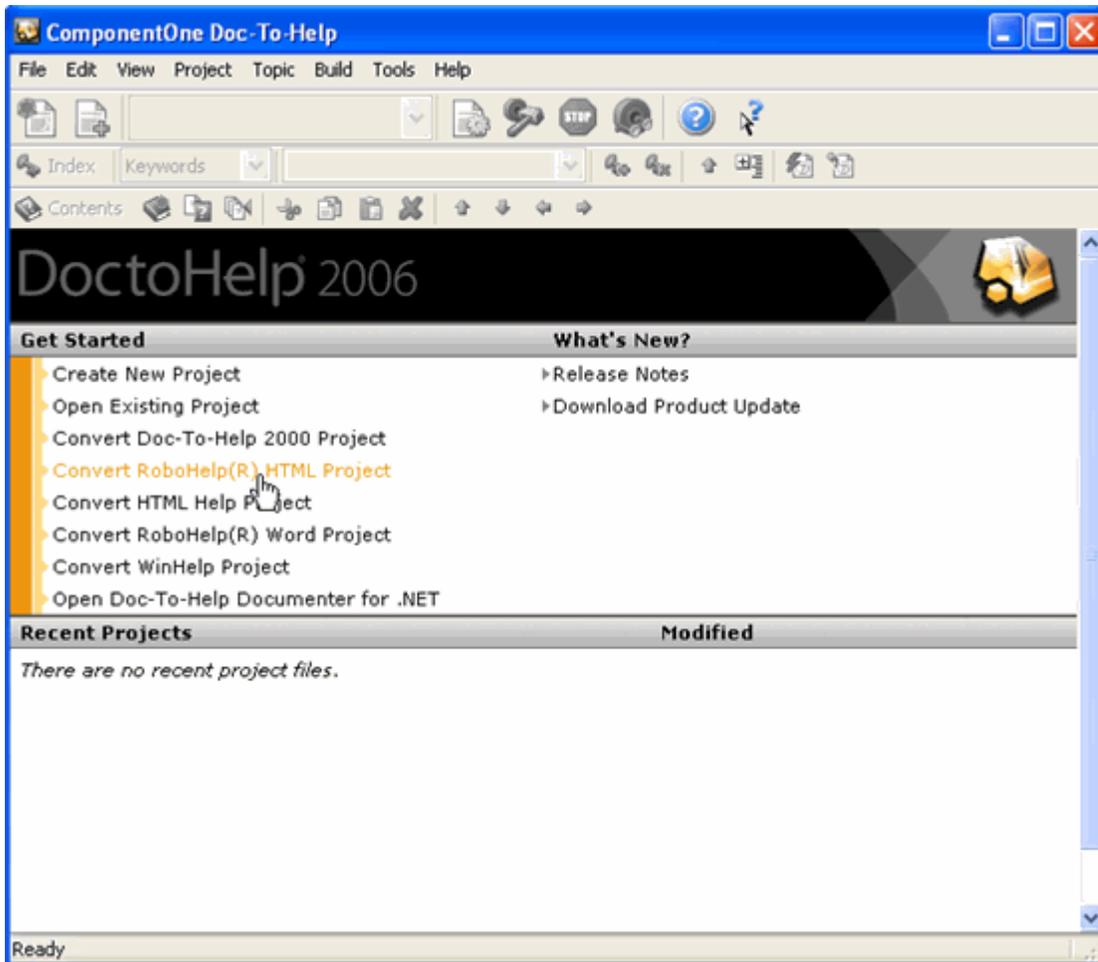
- Instructions to select submenu items are given in the **Menu | Submenu** format.
- Throughout the tutorial, dialog box names, control labels and controls are formatted in a bold font.
- The tutorial instructions assume that you installed Doc-To-Help in the default location. If you installed in another location, you must navigate to files in that location instead of the location specified in the instructions.
- The tutorial instructions use Microsoft Windows XP, Microsoft Word 2003 and Microsoft FrontPage 2003 for all explanations. If you are using a different operating system or a different version of Word or FrontPage, your commands and dialog boxes may vary somewhat.
- The terms, "make target", "rebuild target", and "compile" are used throughout the tutorials. If you are unfamiliar with how these terms are used in relation to Doc-To-Help 2006, please read [Understanding the Doc-To-Help Build Options](#) (page 22).

Converting Projects to Doc-To-Help 2006

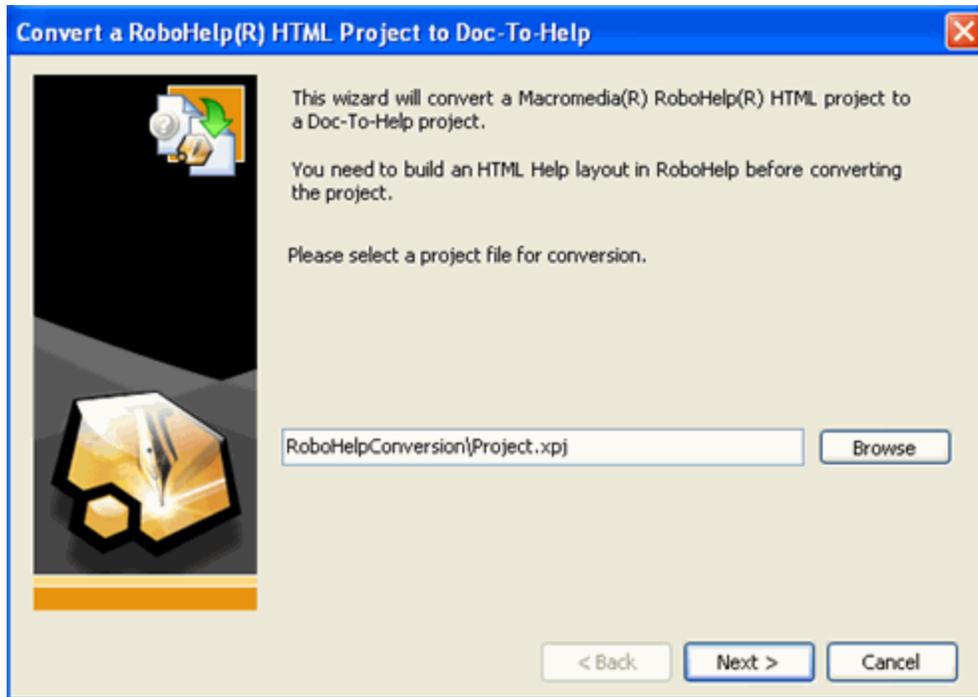
Converting a RoboHelp HTML Project to Doc-To-Help 2006

This tutorial uses a RoboHelp project for which the Microsoft HTML Help layout has already been generated. This HTML Help project (.hhp) is used by the Doc-To-Help RoboHelp converter and it must be generated before conversion can take place.

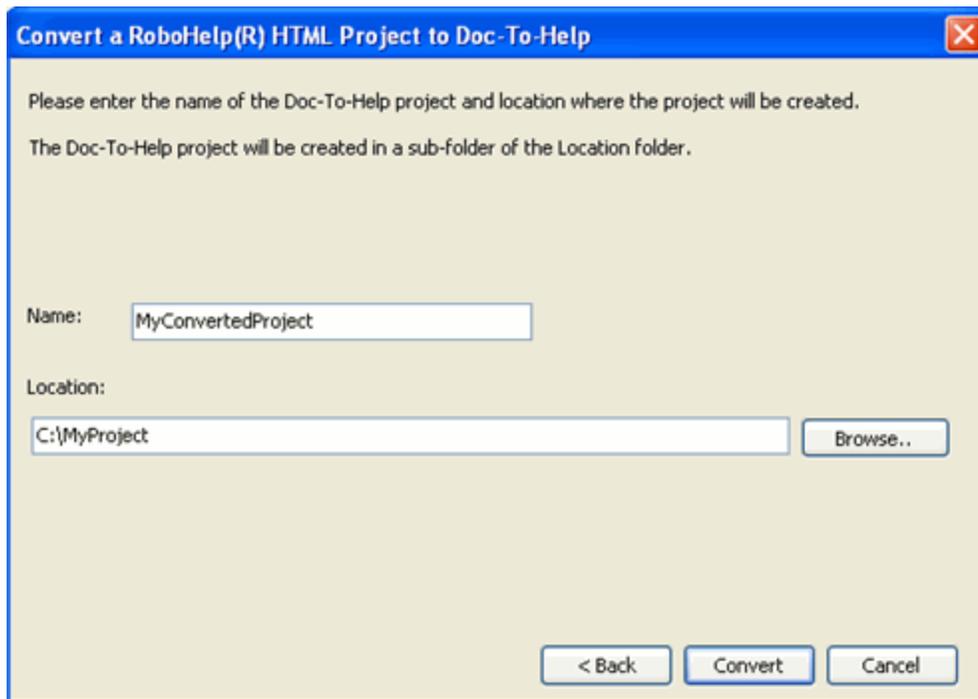
1. Open Doc-To-Help 2006 and click **Convert RoboHelp(R) HTML Project**.



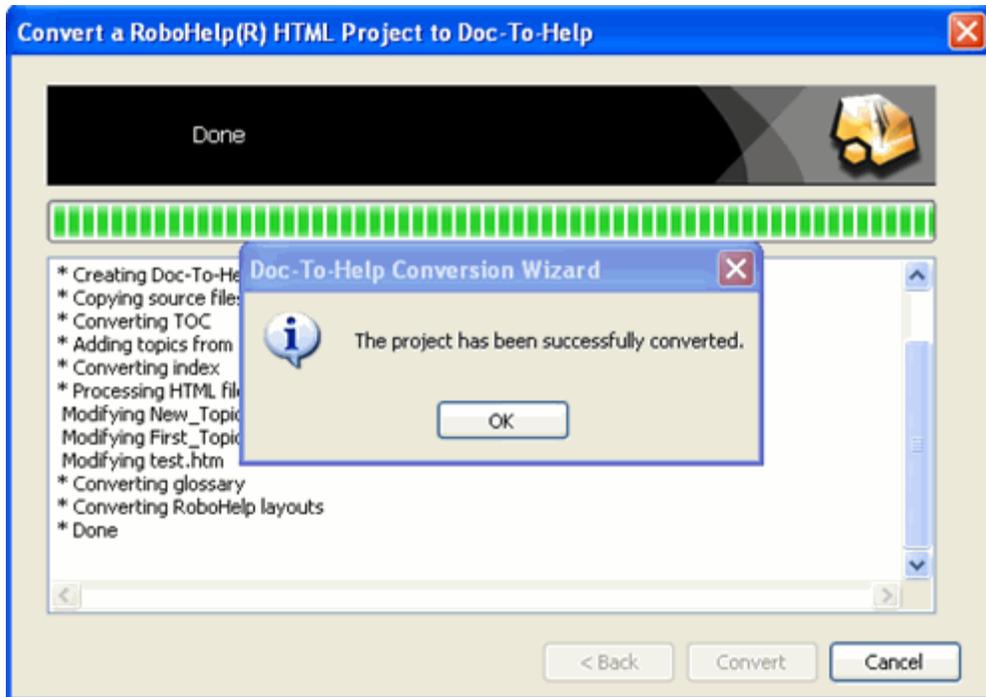
2. Select an **.xpj** project file and click **Open**.
3. Review the location and click **Next**.



4. Enter the name of the Doc-To-Help project to be created in the **Name** text box and browse for a location for the new project.



5. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
6. Once the conversion is complete, click **OK**.

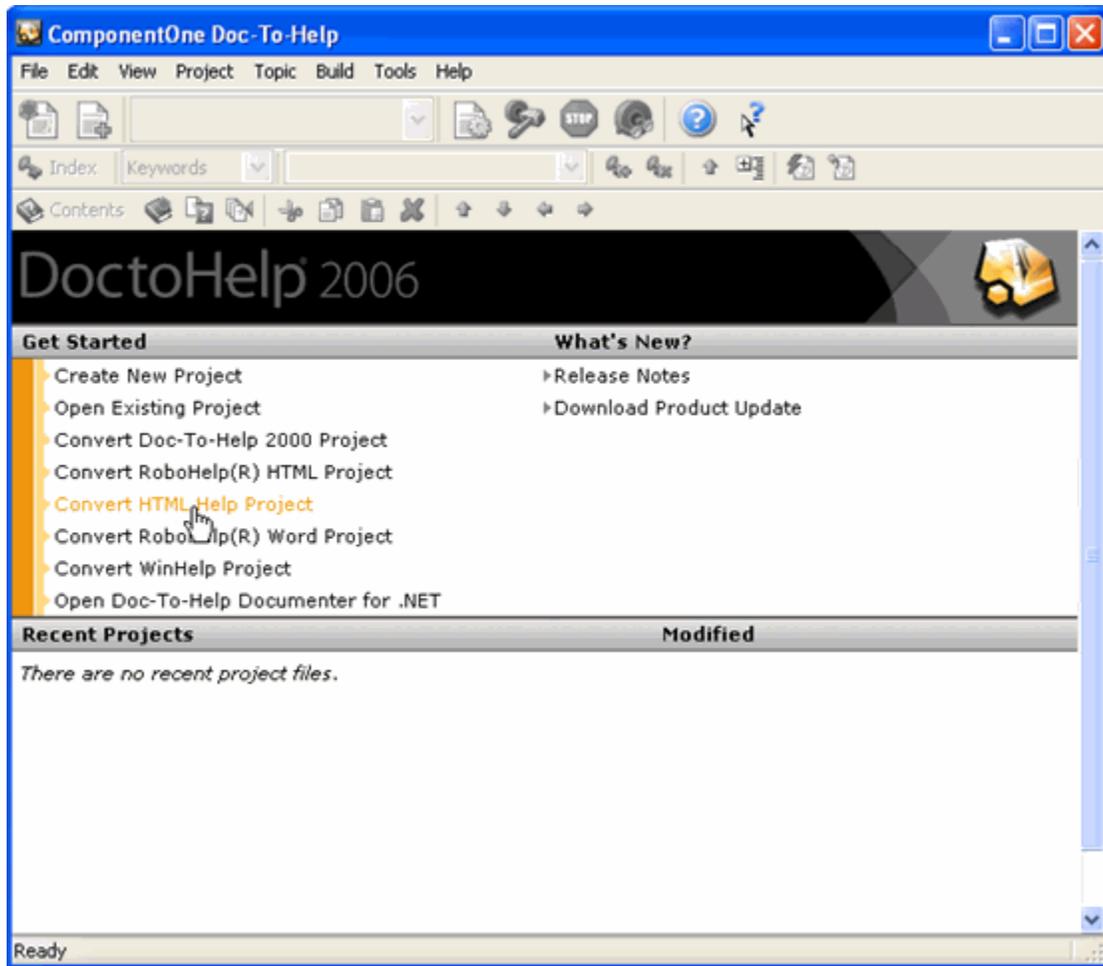


7. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the Doc-To-Help project editor. You can begin customizing and building your help files.

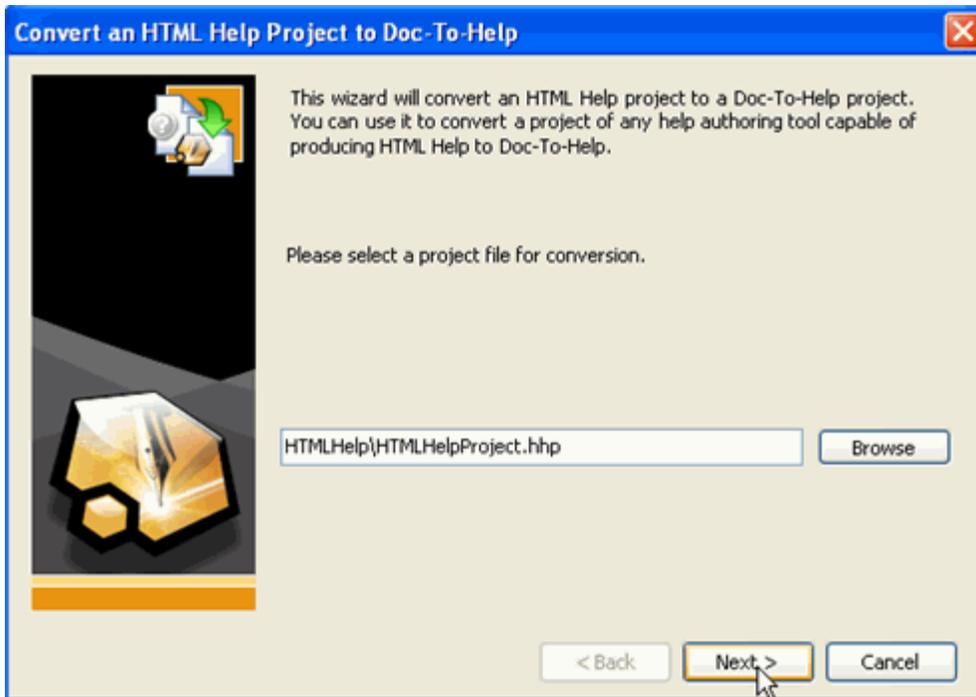
Converting an HTML Help Project to Doc-To-Help 2006

The HTML Help project used in this tutorial is located in the C:\Program Files\ComponentOne\DocToHelp\Tutorial\IceCream\HTML-Compiled-0 folder.

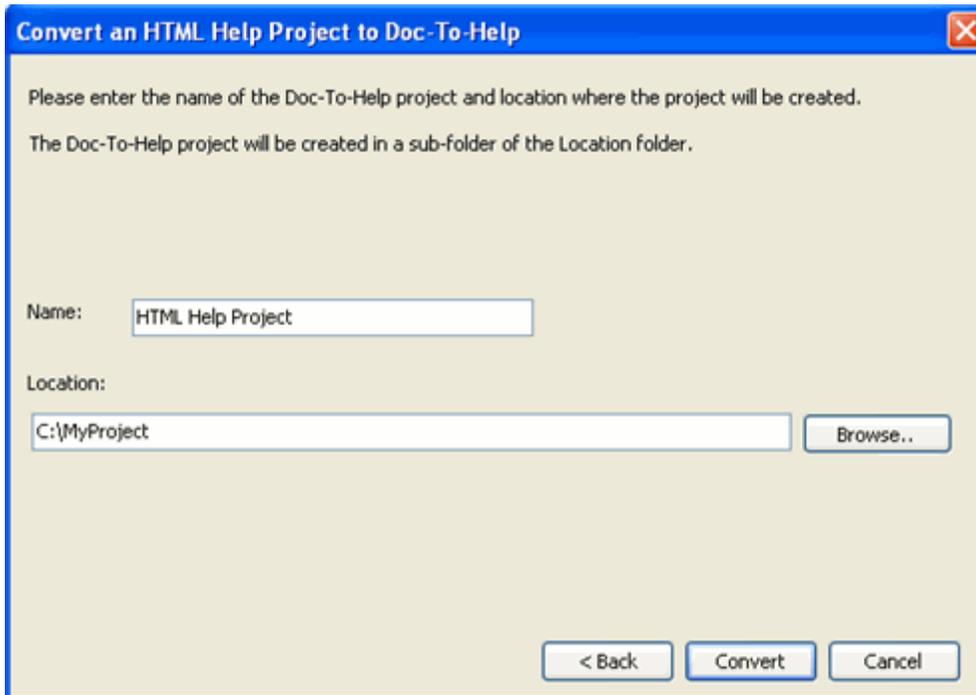
1. Open Doc-To-Help 2006 and click **Convert HTML Help Project**.



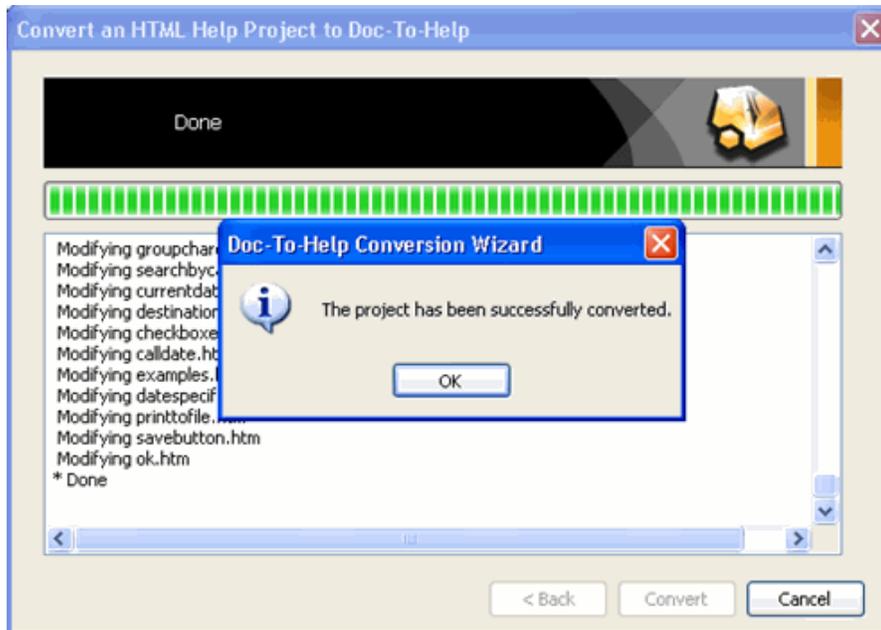
2. Browse to the C:\Program Files\ComponentOne\DocToHelp\Tutorial\IceCream\HTML-Compiled-0 folder and select the **icecream.hhp** file.
3. Click **Open**.
4. Review the location and click **Next**.



5. Enter the name of the Doc-To-Help project to be created in the **Name** text box.
6. Browse for a location for the new project.



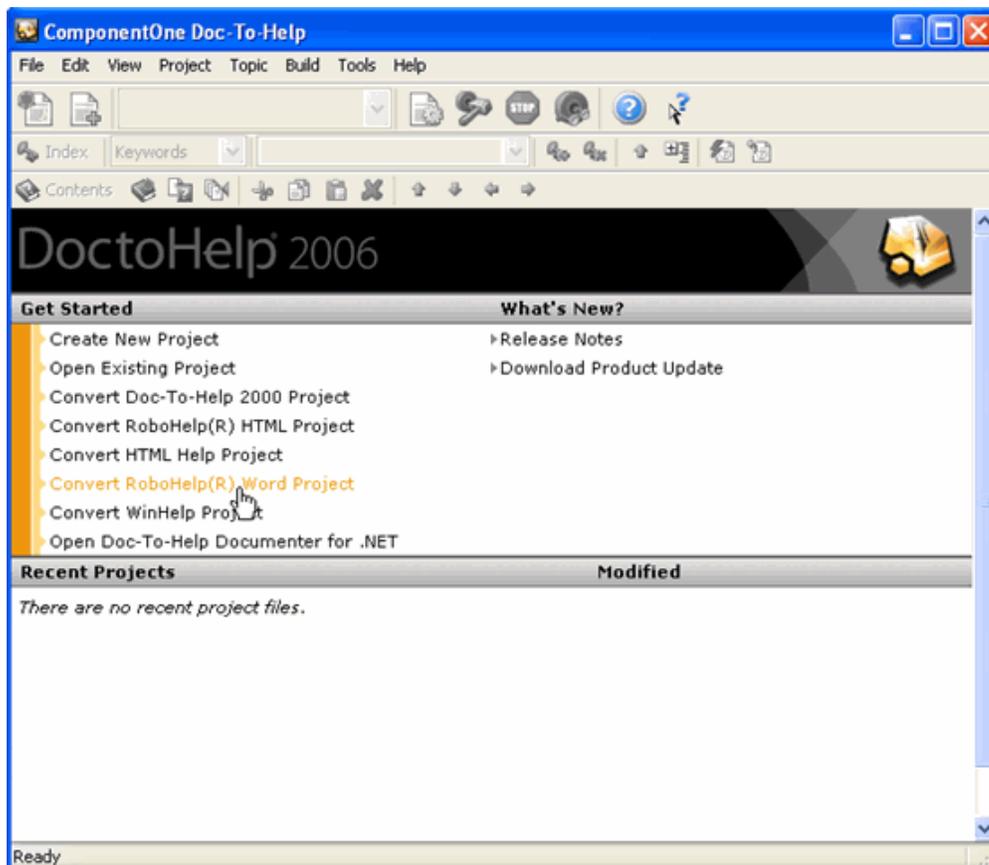
7. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
8. Once the conversion is complete, click **OK**.



9. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the project editor. You can begin customizing and building your help files.

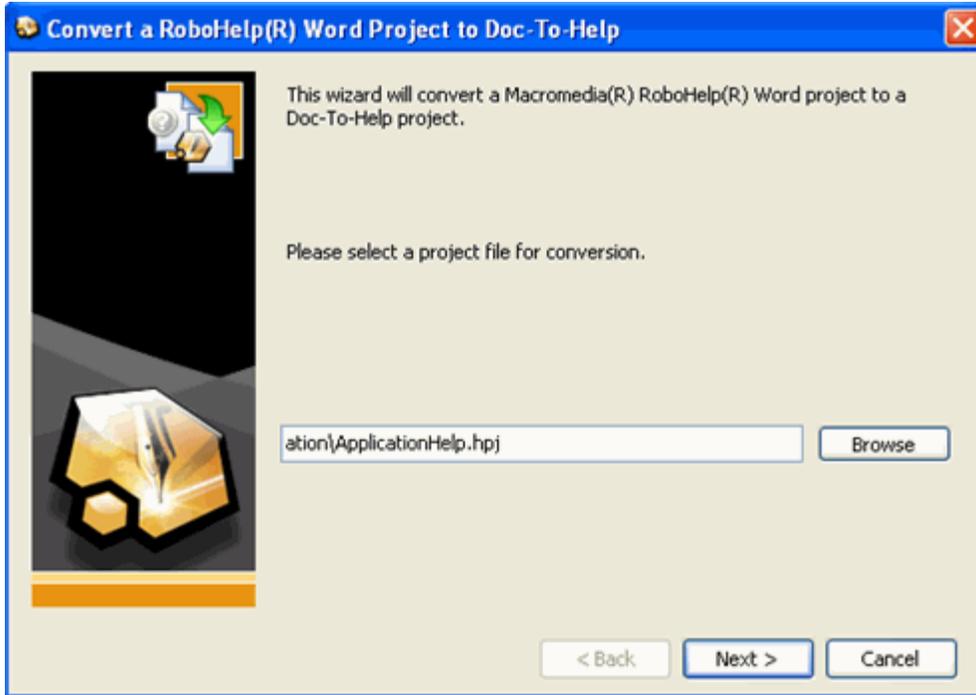
Converting a RoboHelp Word Project to Doc-To-Help 2006

1. Open Doc-To-Help 2006 and click **Convert RoboHelp(R) Word Project**.

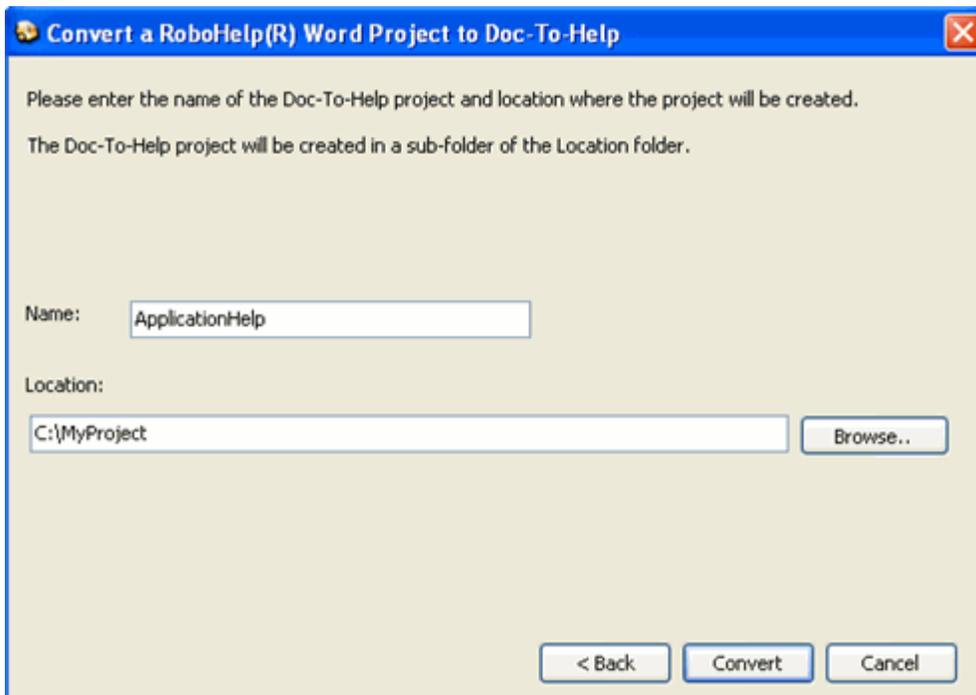


2. Browse, select an **.hjp** project file and click **Open**.

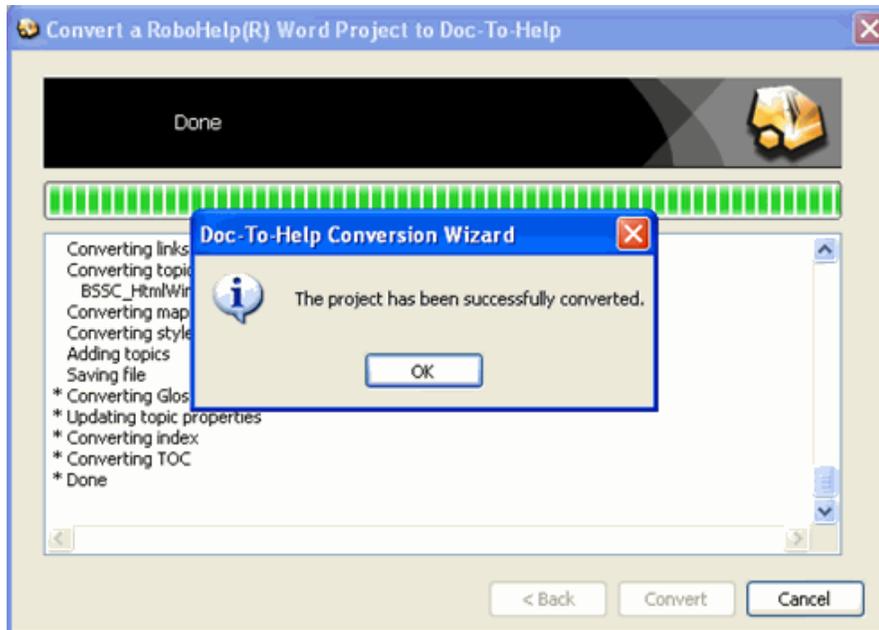
3. Review the location and click **Next**.



4. Enter the name of the Doc-To-Help project to be created in the **Name** text box and browse for a location for the new project.



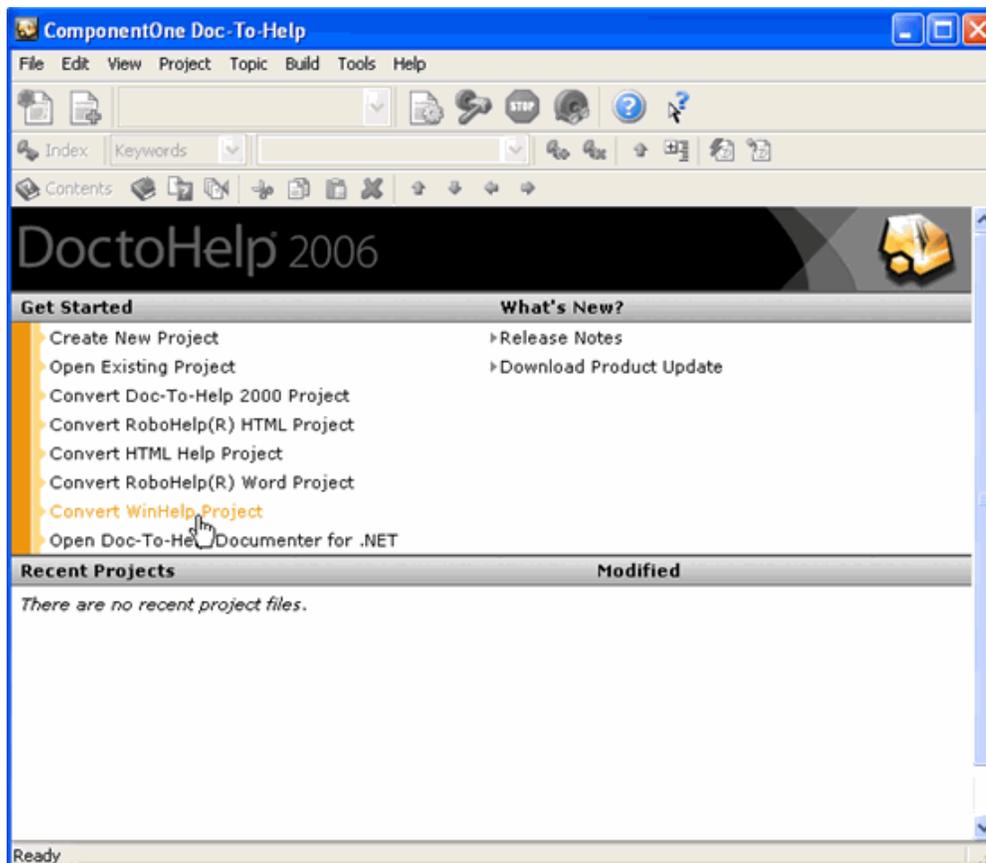
5. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
6. Once the conversion is complete, click **OK**.



7. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the Doc-To-Help project editor. You can begin customizing and building your help files.

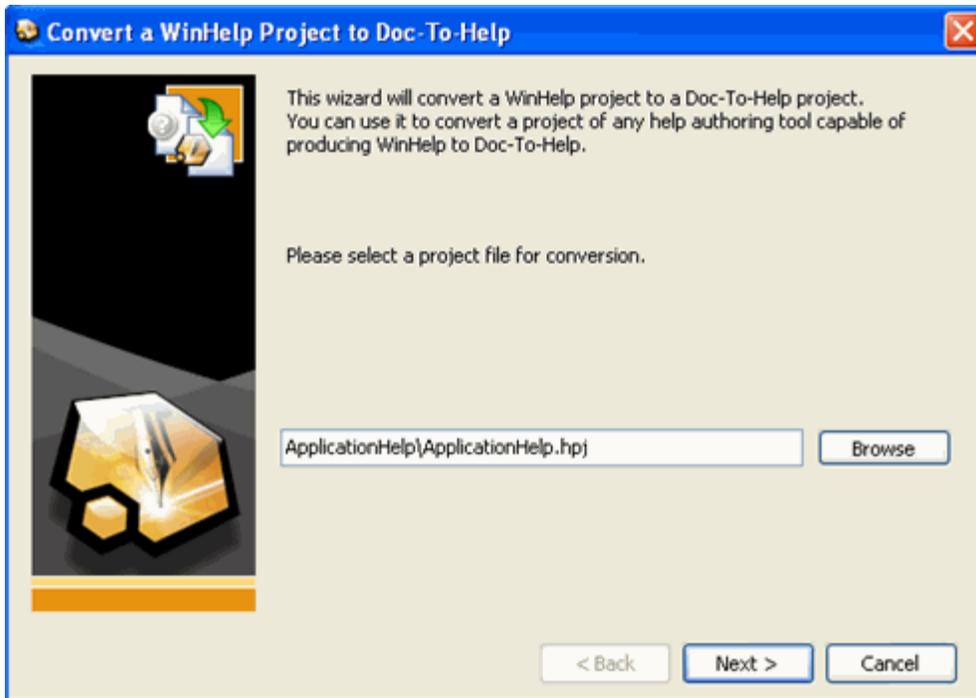
Converting a WinHelp Project to Doc-To-Help 2006

1. Open Doc-To-Help 2006 and click **Convert WinHelp Project**.

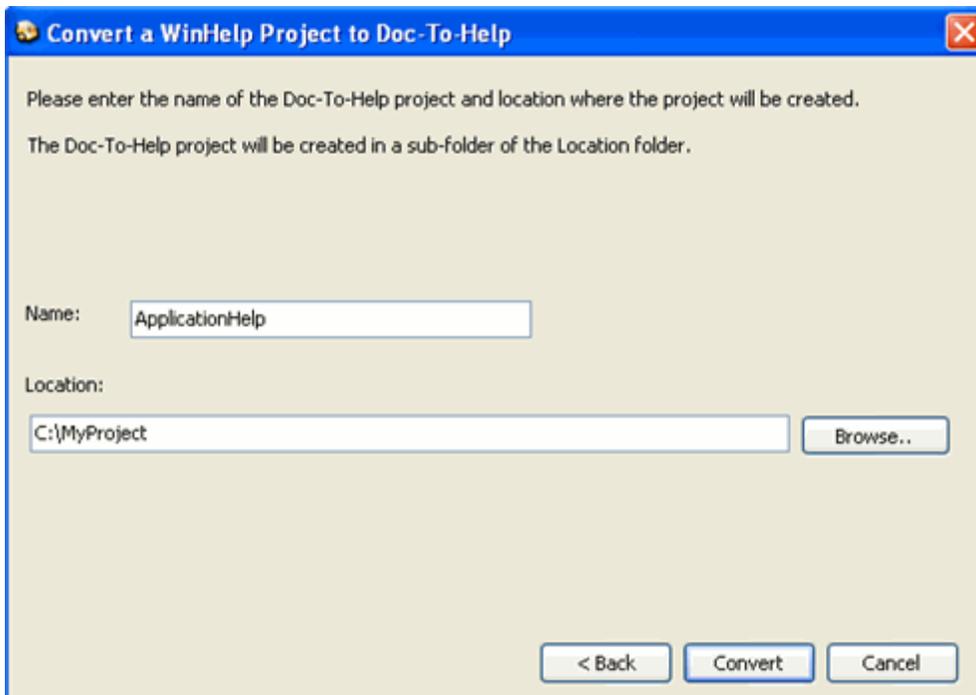


2. Browse, select an **.hjp** project file and click **Open**.

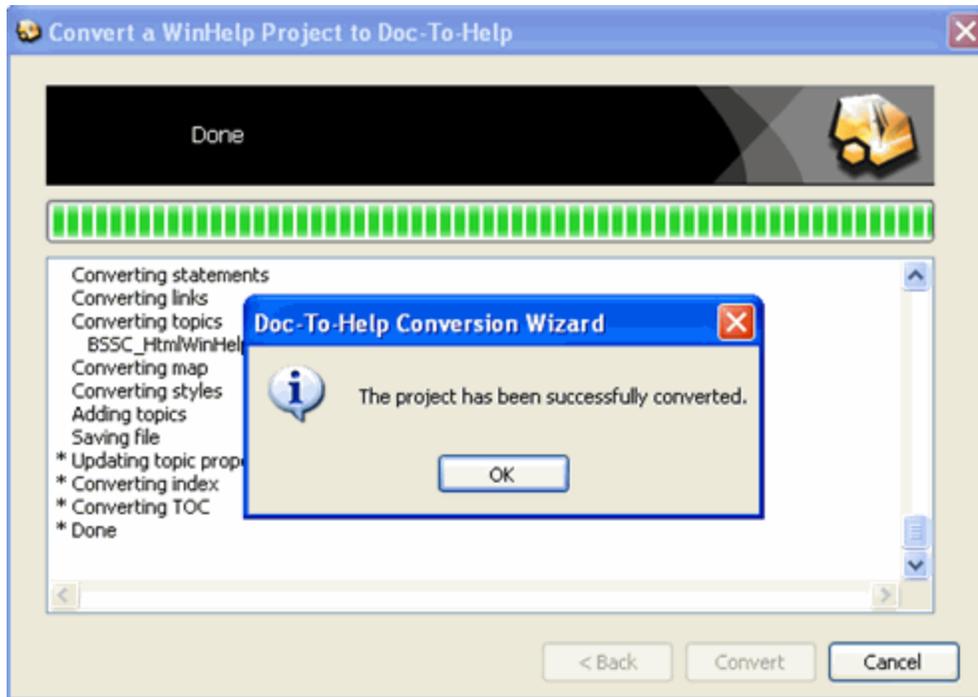
3. Review the location and click **Next**.



4. Enter the name of the Doc-To-Help project to be created in the **Name** text box.
5. Browse for a location for the new project.



6. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
7. Once the conversion is complete, click **OK**.



8. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the project editor. You can begin customizing and building your help files.

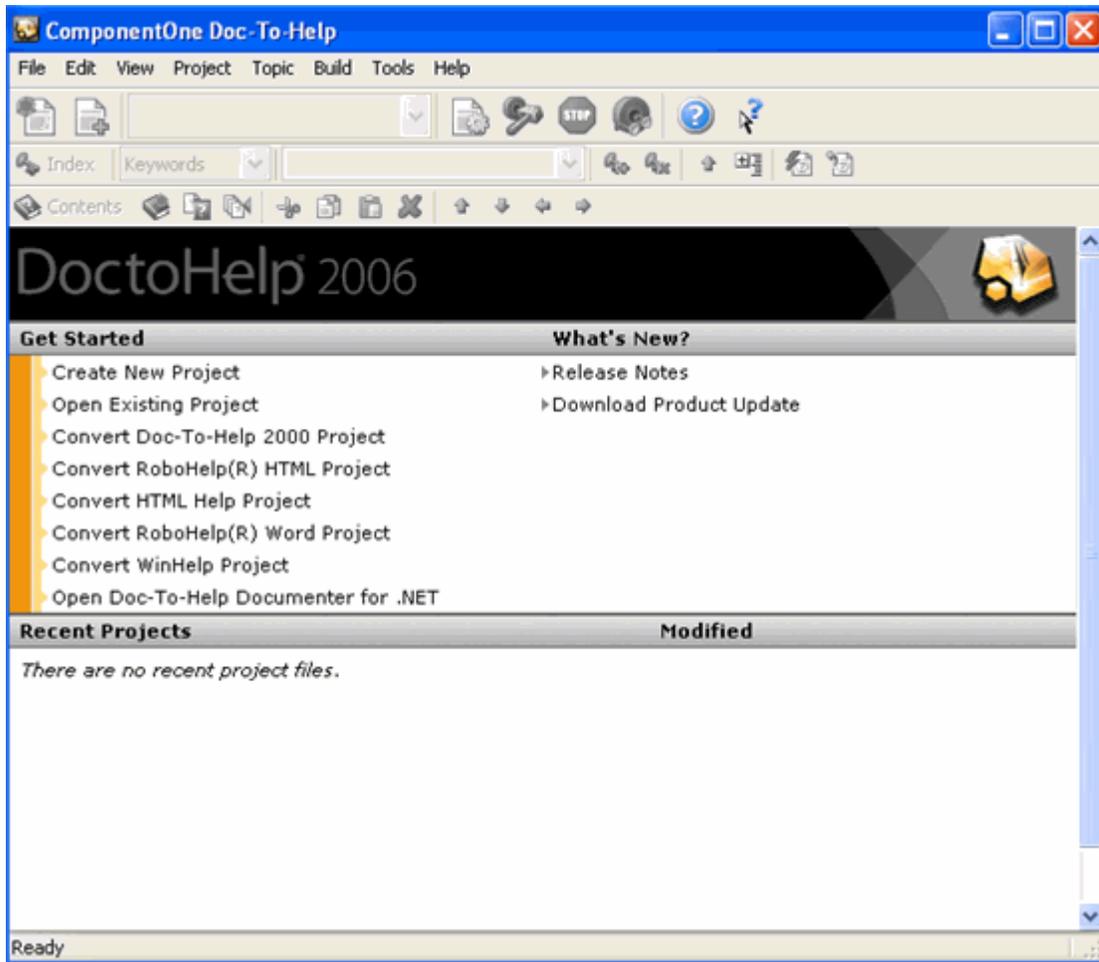
Using Microsoft Word

Before beginning this tour, it may be helpful for you to familiarize yourself with the **Doc-To-Help** toolbar by reading [Exploring the Doc-To-Help Toolbar for Microsoft Word](#) (page 64). Using the **Doc-To-Help** toolbar is just one option when formatting your Word source documents. You can also use the [D2HML Styles toolbar](#) (page 233) and dialog boxes to create Doc-To-Help Markup Language (D2HML). D2HML is the preferred and recommended method for formatting source documents. For more information, see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) and the D2HML tutorial [Using D2HML with Microsoft Word](#).

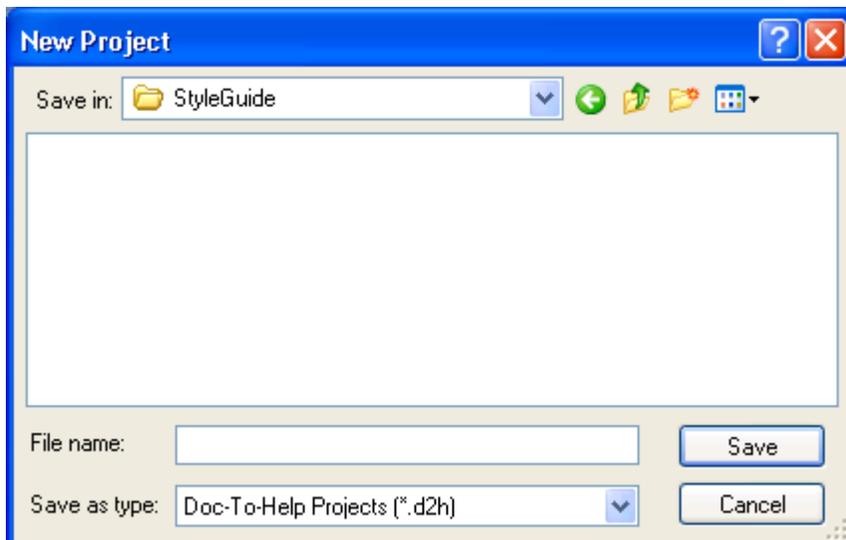
Starting a New Help Project

The first steps in starting a new Help project are to name the project file and select a folder where it will reside. Doc-To-Help 2006 provides you with an "all in one" project editor which allows you to create and edit source documents, add Help features, and build Help files for any platform.

1. Open the Doc-To-Help program. You are presented with the Doc-To-Help start page. From here you can choose some of your most recent projects, open an existing project, convert an older Doc-To-Help project, convert a RoboHelp or HTML Help project, or create a new project.

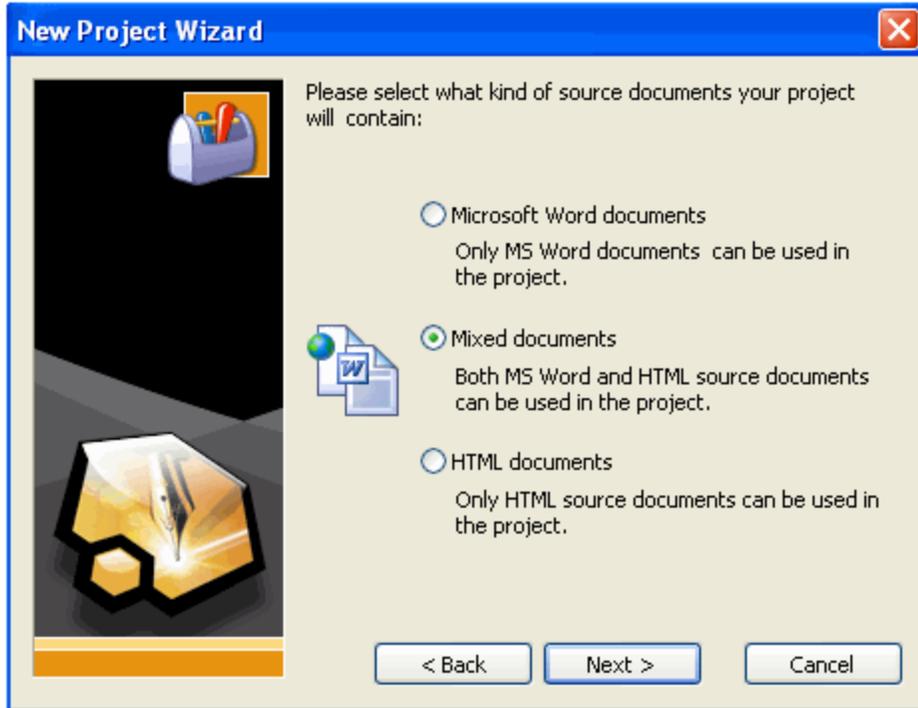


2. Click **Create New Project**. The first screen of the **New Project Wizard** opens.
3. Click **Browse**. The **New Project** dialog box opens as shown below.



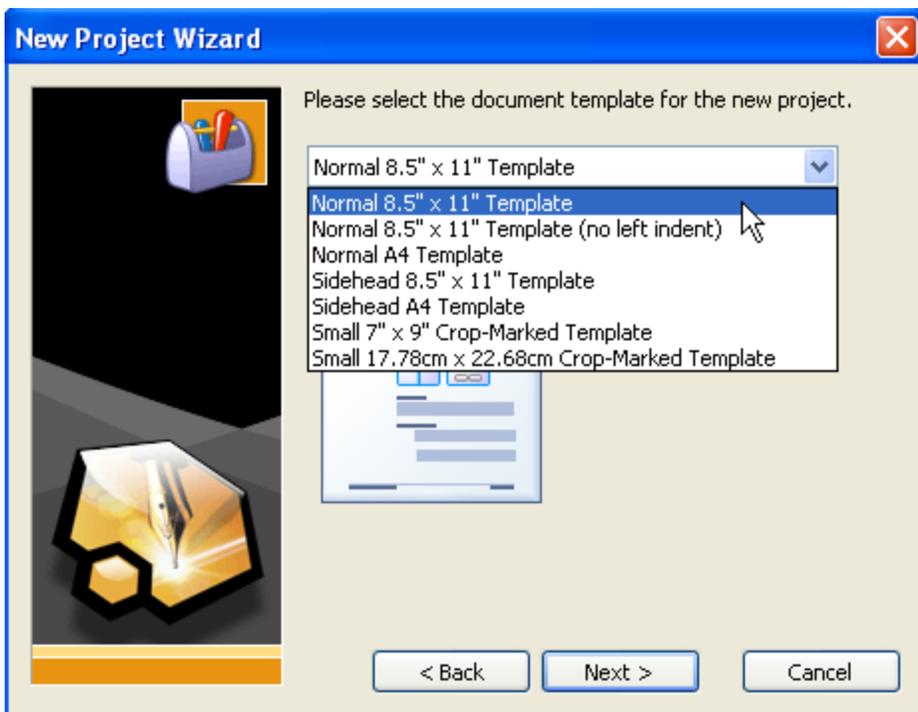
4. Open the C:\Program Files\ComponentOne\DocToHelp\Tutorial\StyleGuide folder.
5. Type **StyleGuide.d2h** in the **File name** textbox, then click **Save**.

6. Click **Next**.
7. Select **Microsoft Word documents** as the type of source documents to use in your project and click **Next**.

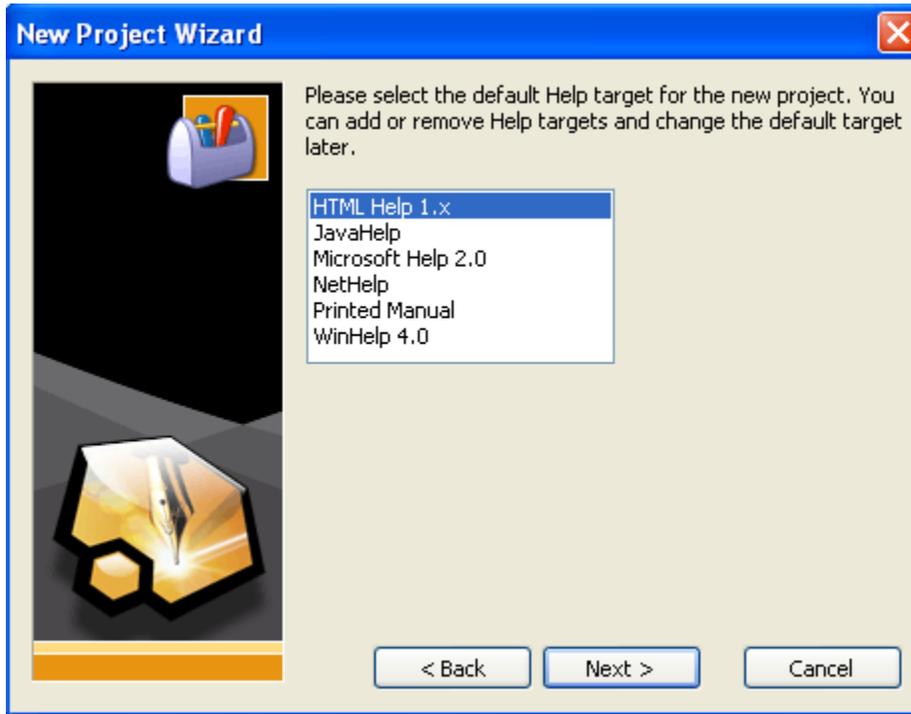


Note: You can change the type of source documents your project uses at any time by changing the [DocumentType](#) property under Project Settings.

8. From the drop-down list, choose the template that you wish to attach to your new project. In this case, make sure that **Normal 8.5" x 11" Template** is selected.



9. Click **Next**.
10. From the list, choose your default Help target. Make sure that **HTML Help 1.x** is selected.



Note: Though you can build any Help target for this project, this selection determines which Help target your project compiles initially.

11. Click **Next**.
12. Review your directory location and file name and click **Finish**. The Doc-To-Help Wizard produces a new Help project and opens the project editor. A blank "start up" and glossary document are automatically created in a WordDocuments folder within your project folder, and they are added to your project. The "start up" document has the same name as your project, StyleGuide in this case, and is automatically opened in Word.

Exploring the Doc-To-Help Toolbar for Microsoft Word

Doc-To-Help installs two global templates named C1D2HAuthor.dot and C1D2HEngine.dot into the Microsoft Office startup directory. These templates add the **ComponentOne Doc-To-Help** toolbar to Microsoft Word, which allows you to edit many of the Help authoring aspects of your source documents.



The **D2HML Styles** toolbar below is also added to your source document, but we will discuss this later in the documentation.



For additional information, see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233).

Toolbar Components

	Show Project Editor	Maximizes the Doc-To-Help project editor. For more information, see The Doc-To-Help Project Editor (page 41).
	Heading 1 Style	Formats the specified text with the Heading 1 paragraph style. For more information, see Working with Styles in Word (page 227).
	Heading 2 Style	Formats the specified text with the Heading 2 paragraph style. For more information, see Working with Styles in Word (page 227).
	Heading 3 Style	Formats the specified text with the Heading 3 paragraph style. For more information, see Working with Styles in Word (page 227).
	Heading 4 Style	Formats the specified text with the Heading 4 paragraph style. For more information, see Working with Styles in Word (page 227).
	Body Text Style	Formats the specified text with the Body Text paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Number Style	Formats the specified text with the C1H Number paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Number 2 Style	Formats the specified text with the C1H Number 2 paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Bullet Style	Formats the specified text with the C1H Bullet paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Bullet 2 Style	Formats the specified text with the C1H Bullet 2 paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Bullet 2A Style	Formats the specified text with the C1H Bullet 2A paragraph style. For more information, see Working with Styles in Word (page 227).
	C1H Continue List	Formats the specified text with the C1H Continue paragraph style. For more information, see Working with Styles in Word (page 227).
	Space Before	Adds or subtracts space before the current paragraph.
	Space After	Adds or subtracts space after the current paragraph.
	Special Bold	Formats the specified text with the Special Bold character style.
	Apply Default Page Setup	Automatically sets your page formatting to match the template attached to the document.
	Standard Table	Inserts the standard Doc-To-Help table at the cursor location. For more information, see Inserting Standard Tables with Doc-To-Help (page 368).
	Margin Note	Inserts a margin note table adjacent to the active paragraph. Creates a hyperlink between the highlighted text and a margin note. For more information, see Using Margin Notes in Word (page 323).
	Image Map Editor	Opens the Image Map Editor Utility. For more information, see Using the Image Map Editor in Microsoft Word (page 560).
	Insert Cross-reference	Inserts a cross reference hyperlinks for online Help and page references for print manuals. For more information, see Creating Cross References (page 320).
	Complete Cross-reference	Completes the cross reference. For more information, see Creating Cross References (page 320).

	Add Topic Link	Creates a hyperlink between highlighted text and a Help topic heading. For more information, see Using the Add Topic Link Dialog Box (page 319).
	Add Dynamic Link	Creates a hyperlink between the highlighted text and a index keyword or group of index keywords. For more information, see Using Dynamic Links (page 328).
	Apply Conditional Text	Creates conditional text from highlighted text or graphics. For more information, see Specifying Conditional Text (page 401).
	View Conditional Text	Modify the conditional text colors and determine which text is viewable in the source document. For more information, see Viewing Classic Doc-To-Help Conditional Text (page 406).
	Add Topic	Add a topic to your project without rebuilding your source documents. For more information, see Using the Topic Tools in Word (page 604).
	Rename Topic	Rename a topic to your project without rebuilding your source documents. For more information, see Using the Topic Tools in Word (page 604).
	Delete Topic	Delete a topic to your project without rebuilding your source documents. For more information, see Using the Topic Tools in Word (page 604).
	Add Glossary Terms	Inserts glossary terms into the glossary source document. For more information, see Using the Add Glossary Terms Tool in Word (page 336).
	Sort Glossary	Sorts the glossary entries in the glossary document alphabetically. For more information, see Sorting Glossary Entries in Word (page 338).
	Insert Help Macro	Inserts a WinHelp macro into your source document. For more information, see Using WinHelp Macros in Word (page 379).
	Insert HTML ActiveX Control	Inserts an HTML ActiveX control into your document. For more information, see Using HTML Help Object Tags in Word (page 378).
	View Field Codes	Toggles the field codes in Microsoft Word so that they are visible.
	Make Target	Compiles the currently selected Help target type. For more information, see Building the Current Help Target (page 164).
	View Target	Activates the currently selected Help target. For more information, see Viewing the Current Help Target (page 166).
	Help	Opens the Doc-To-Help online Help.

Adding the Toolbar to Custom Templates

The ComponentOne Doc-To-Help toolbar and the D2HML Styles toolbar for Microsoft Word reside in individual user templates, such as C1H_NORM.DOT. Therefore, these toolbars are only initially available in source documents that have a Doc-To-Help template attached. If you would like to use these toolbar with your own custom templates, you can copy them using Word's **Organizer** dialog box.

To copy the toolbars to your template:

1. Open your template in Word, and select **Tools | Templates and Add-Ins**.
2. Click the **Organizer** button and select the **Toolbars** tab in the **Organizer** dialog box.
3. Click the **Close File** button on the right-hand side, and click the button again when it changes to **Open File**.

4. Find and select the **C1H_NORM.DOT** or another Doc-To-Help template, and then click **Open**.
5. Select the desired toolbars on the right-hand side and click **Copy** to copy the toolbars to your custom template.

Working with the First Source File

You are now ready to start writing your first Help document. The default document in Microsoft Word contains a Chapter 1 placeholder where you can start entering your text.

Before working in the document, you may want to set Doc-To-Help and Word preferences to make your work easier. If you do not want to change your Word options, skip [Setting Word Preferences](#) (page 67) and start with [Entering Text in the Document](#) (page 67).

Setting Word Preferences

1. If it is not already open, double-click the **StyleGuide.doc** in the project editor to open it in Word.
2. Select **Tools | Options** to open the **Options** dialog box.
3. Select the **View** tab.
4. In the **Formatting marks** area, select **All** to display nonprinting characters.
5. In the Show area, select **HiddenText** if you want to view Word {XE} fields.
6. Click **OK**.

For more information, see [Word Source Documents](#) (page 185).

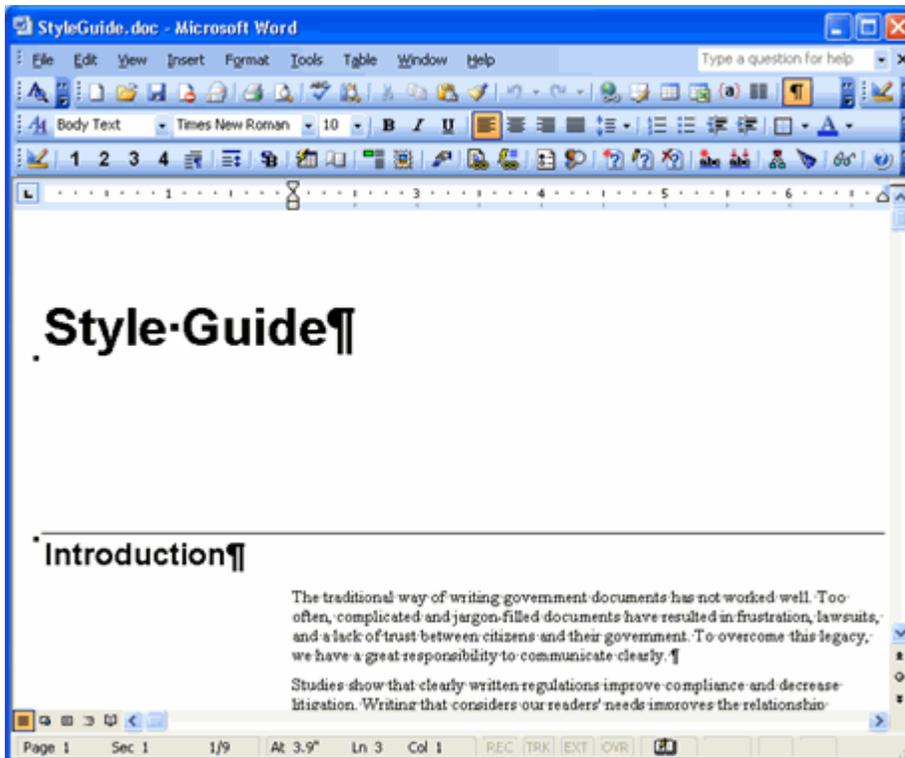
Entering Text in the Document

1. Highlight the Chapter 1 placeholder.
2. Change the text to "Style Guide" and then press **Enter**. Notice that the next paragraph has been formatted as heading 2 and has a line above it.
3. Type "Introduction" and press **Enter**. You should now have a heading 1 and a heading 2, followed by a third empty paragraph formatted as body text. You can delete the Topic 1 heading and the body text below it.

Inserting a File

It's time to add some text to your source document. To save time, we will insert a file containing the remaining style guide documentation.

1. With your cursor in the body text paragraph, select **Insert | File**. The **Insert File** dialog box opens.
2. Locate the file C:\Program Files\ComponentOne\DocToHelp\Tutorial\StyleGuide\StyleGuideContents.doc.
3. Select the file and click **Insert**. Doc-To-Help inserts the text for the style guide.



Saving the Document and Building a Help File

It is time to save your first source document and build a Help file. As you will see, Doc-To-Help allows you to build multiple Help files in multiple formats using your existing source documents.

Saving the Document

1. Select **File | Save**.

The first source document is saved to the WordDocuments folder with the same name as the project (in this case, StyleGuide.doc).

2. Close the StyleGuide document.

After closing the document, you will notice that the right pane of the project editor contains options for making and viewing the Help target and links to each of the current project documents (Glossary.doc and StyleGuide.doc). An empty Glossary.doc file is automatically produced when Doc-To-Help creates your Help project. Later in the tutorial, you will learn more about using glossaries.

Building a Help File

1. If necessary, select **View Toolbars | Project** to display the **Project** toolbar.
2. Verify that **StyleGuide HTML Help** appears in the **Help Target** drop-down list.



The **Help Target** drop-down list allows you to select from a wide range of Help targets including a **Manual** option.

3. To build the HTML Help file, click the **Make Target** button  or the **Make Help Target** link in the right Help pane. The right pane of the Doc-To-Help project editor shows the status of the Help file as it builds and compiles.
4. After the project compiles, you can examine your work by clicking the **View Target** button  on the tool bar.
5. When you are finished, close the Help file.

For more information on Help targets, see [Working with Projects](#) (page 129).

Using Styles to Enhance a Help Project

By customizing the properties of paragraph and character styles in Doc-To-Help, you can easily define and control the behavior of the compiled Help.

Doc-To-Help supplies templates and style sheets for you to author your source document in Word or HTML and to create output target Help files. Generally, the styles contained in the templates and style sheets will fill most of your needs, but you can further customize the Help behavior by adding your own styles to these files.

Creating a new style (whether paragraph or character) for your Help system is a 3 step process:

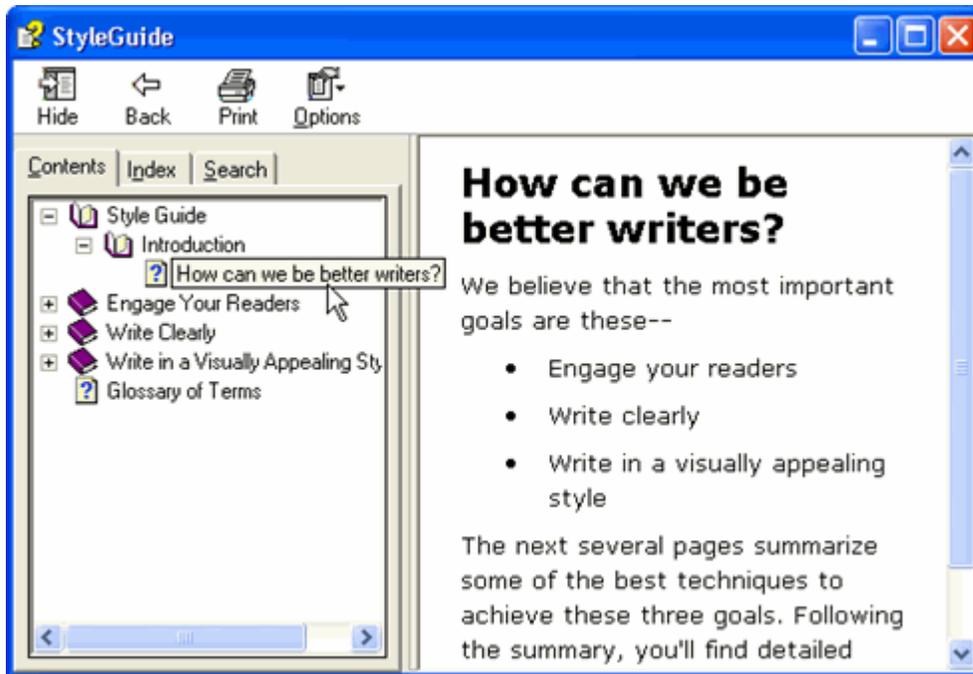
- Create a new style in Word or HTML - with properties to suit your formatting/appearance requirement for your source document.
- Create a like-named style in your output Help target template file (e.g., C1H_HTML.DOT for HTML and HTML Help output targets) or style sheet (e.g., C1H_HTML_full.css). This is to control how your Help output will appear/look in your Help target.
- Create a like-named style in Doc-To-Help – with properties to define the compiled Help behavior.

The sections below demonstrate the basic techniques to apply and create paragraph and character styles in Doc-To-Help.

Applying Paragraph Styles

The following steps guide you through the process of applying existing Doc-To-Help paragraph styles to create Help contents topics.

1. Click the **View Target** button.
2. Click the **StyleGuide** contents topic, then **Introduction**. Note that the "Introduction" topic is the lowest Help level.
3. Close the Help file.
4. From the Icon Bar, select **Project**.
5. In the left pane, select **Documents**.
6. In the right pane select StyleGuide.doc, right-click and select **Open** from the shortcut menu.
7. Under the "Introduction" section, find the "How can we be better writers?" text.
8. Place your cursor anywhere in that line.
9. Apply the **Heading 3** style to that line.
10. Save and close Word and return to the Project Editor.
11. Click the **Make Target** button.
12. Click the **View Target** button. Note that the "How can we be better writers?" text is now a topic sub-heading under the "Introduction" heading.



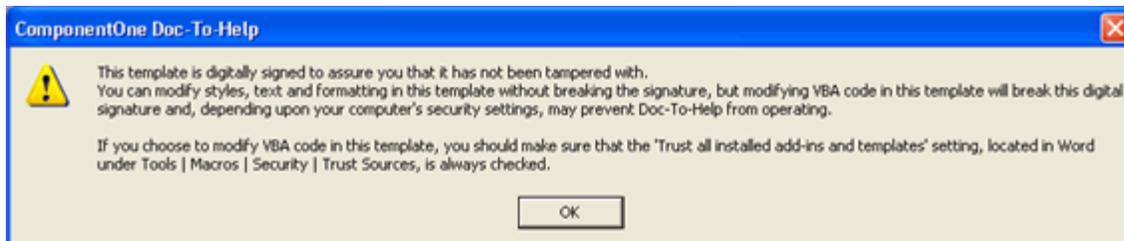
For more information on Help targets, see [Applying Paragraph Styles in Microsoft Word](#) (page 227).

Defining a Style in Word

Doc-To-Help provides a wide variety of styles that can be used to format your source documents. In rare cases, these existing styles may not be exactly what you need. If necessary, you can define a new style (paragraph or character) and add it to your "source" template. Please note that creating styles is an advanced feature and is not necessary in most cases.

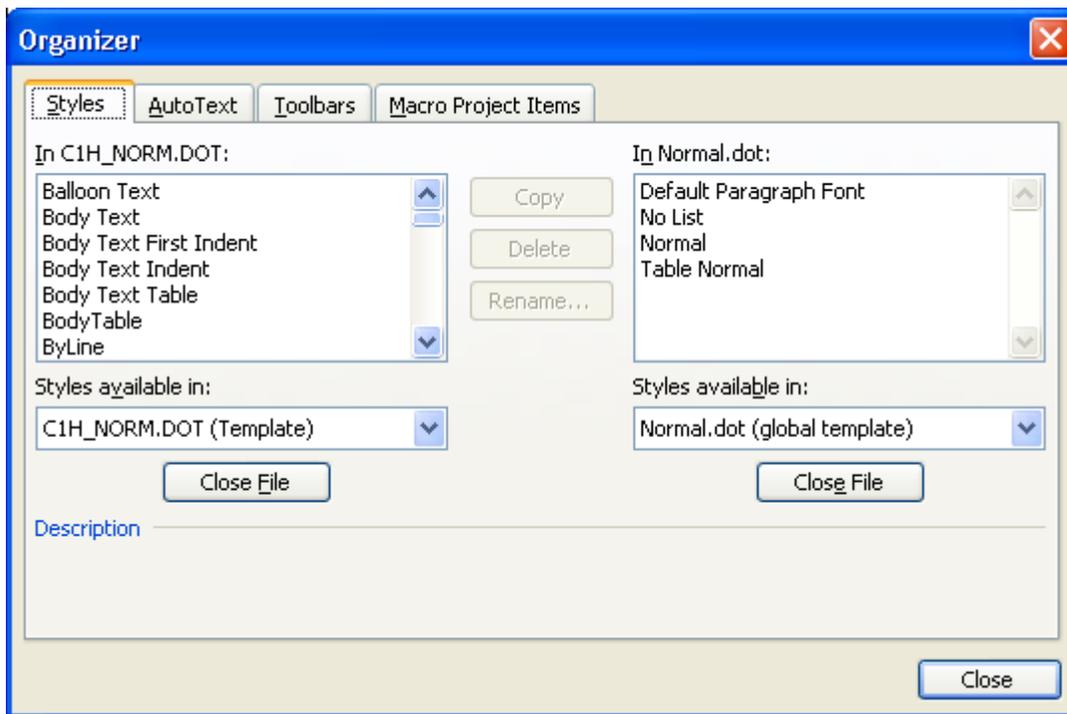
1. In the **Icon Bar**, click **Project**.
2. In the left pane, select **Templates**.
3. In the right pane select C1H_NORM.DOT, right-click and select **Open** from the Shortcut menu. This opens the C1H_NORM template that is used as the default template for formatting the source documents in Doc-To-Help.

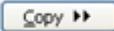
Note that the following message explaining the Doc-To-Help digitally signed template displays when you open any of the preloaded templates.



4. Scroll down to the bottom of the template file and place your cursor on an empty line.
5. Select **Format | Styles and Formatting**. The Styles and Formatting window opens.
6. Click **New Style**.
7. From the **New Style** dialog box, type My Heading 2 in the **Name** textbox.
8. From the **Style based on** drop-down, select the **Heading 2** style.

- Click the **Organizer** button. The **Organizer** dialog box opens. Note that the left side of the organizer contains the styles from the currently opened template, while the right side contains the Normal.dot template.

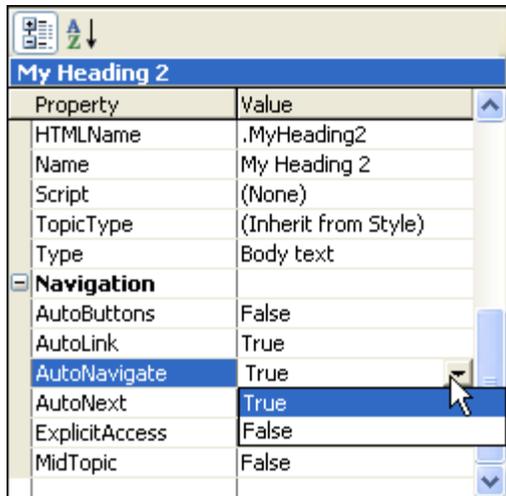


- Close the Normal.dot by clicking **Close File** on the right side of the organizer.
- To select the HTML Help template (C1H_HTML.DOT), click the **Open File** button on the right side of the organizer.
- Select the C1H_HTML.DOT template from the **Open** dialog box and click **Open**.
- In the In C1H_NORM.DOT listbox, select My Heading 2.
- Click the **Copy** button  to copy the style to C1H_HTML.DOT.
- Click the **Close File** button for the C1H_HTML.DOT.
- You are prompted "Do you want to save changes to C1H_HTML.DOT?", click **Yes**.
- Close the organizer and close the C1H_NORM.DOT file.

Defining Styles in Doc-To-Help

After creating a new paragraph style in your source template and copying it to your Help target template, the next step is to create a matching paragraph style in Doc-To-Help to control your Help system behavior.

- If necessary, open the Doc-To-Help project editor.
- In the **Icon Bar**, click **Project**.
- In the left pane, select **Paragraph Styles**.
- In the right pane, right-click and select **New**.
- Type **My Heading 2** in the name column and press **Enter**.
- In the property pane, change the **Type** property to **Level 2**.
- To allow this topic to be included in the navigation sequence, set the **AutoNavigate** property to **True**.



For more information on Help targets, see [Using Styles in Doc-To-Help](#) (page 211).

Applying a User Defined Style

Applying user-defined paragraph styles is performed in the same manner as applying built-in styles. The steps below demonstrate how to apply the paragraph style we defined above in our source file.

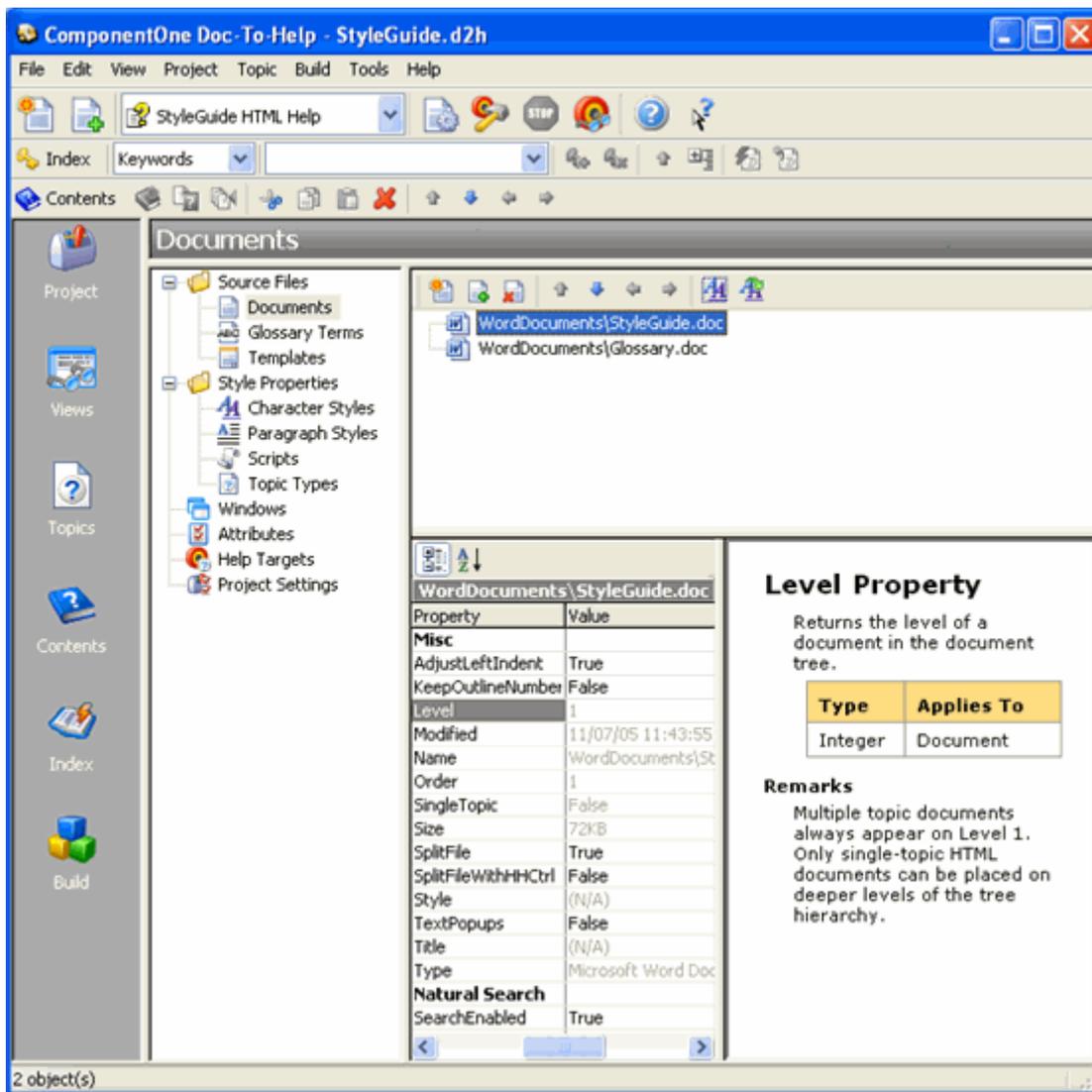
1. In the **Icon Bar**, click **Project**.
2. In the left pane, click **Documents**.
3. In the right pane select StyleGuide.doc, right-click and select **Open** from the shortcut menu.
4. Scroll down to find the "Engage Your Readers" section, find the "How can you engage your readers?" text.
5. Place your cursor anywhere in that line.
6. Apply the **My Heading 2** style to that line. Note that if the style does not appear in the **Styles and Formatting** window, select **Tools | Templates and Add-Ins**, and check the *Automatically update document styles* checkbox. Click **OK** to close the window.
7. Save and close Word and return to the Project Editor.
8. Click the **Make Target** button.
9. Click the **View Target** button.
10. Click the "Engage Your Readers" Topic.
11. Click the "How can you engage your readers?" sub-topic. Note that the "How can you engage your readers?" font has changed to reflect your custom style.
12. When you're finished, close the Help file.

Adding Multiple Word Source Documents to a Help Project

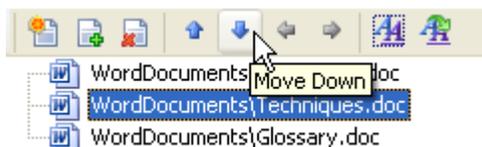
One of the features of Doc-To-Help is the ability to add and easily manipulate separate Word and HTML source documents in a Help project. One of the advantages of using separate source documents for each major "Heading 1" topic is that it provides you with the flexibility of having several authors working on the project simultaneously. Additionally, when changes are made to the documents, Doc-To-Help only needs to compile the source documents that have been modified, greatly reducing the compile time. Let's add a second source document to our StyleGuide Help project.

1. In the Icon Bar, click the **Project** icon.

- In the left pane of the project editor, select **Documents**. The right pane shows the project documents, while the properties pane shows the properties for the highlighted project document.



- Select **Project | Add Documents**. The **Add Documents** dialog box opens.
- Locate the file C:\Program Files\ComponentOne\DocToHelp\Tutorial\StyleGuide\Techniques.doc.
- Select the file and click **Open**. Notice that the Techniques.doc has been added to the document list in the right upper pane of project editor. The order of the documents in your Help file is noted by the order property in the properties pane. To change the order of the documents, simply select the document you wish to move and use the **Move Up** or **Move Down** arrows located on the **Documents** toolbar.



Note: When a document resides outside the project folder, Doc-To-Help displays the full path to the document. Adding a document to a project does **not** move it to the project folder. If you want the document to reside in the project folder, move it to the project folder before adding it to the project.

6. Add the new document to the Help file by clicking the **Make Target** button.
7. After the Help file is compiled, click the **View Target** button. Notice that the new documentation has been added to your Help file.
8. When you are finished, close the Help file.

For more information on Help targets, see [Adding a Document to a Project](#) (page 191).

Adding Tables in Doc-To-Help

In many cases, it is necessary to use a table to clarify complex lists of information. With Doc-To-Help, you can use Word's table feature to include formatted tables in your documentation. In this section, you will format existing text as a table and include it in your project.

1. Right-click the StyleGuide.doc and select **Open** from the shortcut menu.
2. Scroll down the document until you locate the "Use tables" section.
3. Highlight the tab-separated text.

Component	→	Minimum Requirements
Processor	→	Pentium 75-MHz or greater processor
Memory	→	32-MB of RAM-Minimum
Hard-Disk Space	→	10-MB hard-disk space for application files
Operating System	→	Microsoft Windows ME, NT (SP 3 or later), 2000
Microsoft Word	→	MS Word 97 (version 8.0) or greater

4. Select **Table | Convert | Text to Table**. The **Convert Text to Table** dialog box opens.
5. Select the **AutoFit to contents** option from the dialog box.
6. Select **Separate Text at Tabs**.
7. Click **OK**.
8. Highlight the top row of the new table.

Components	System Requirements

9. Select **Format | Borders and Shading**.
10. Select the **Shading** tab and change the **Fill** to pale blue.
11. Click **OK**.
12. Save and close the StyleGuide document.
13. Click the **Make Target** button.
14. Click the **View Target** button.
15. View the table. Note that the colored table header was carried into your help file.

Use tables

Tables are used to present related information concisely. Use tables to visually display information in a way that dense text never can. The simple "System Requirements" table below presents the information very clearly:

Components	System Requirements
Processor	Pentium 75 MHz or greater processor
Memory	32 MB of RAM Minimum
Hard Disk Space	10 MB hard disk space for application files
Operating System	Microsoft Windows ME, NT (SP 3 or later), 2000
Microsoft Word	MS Word 97 (version 8.0) or greater

Note: Table formatting is not available in WinHelp. Although the WinHelp compiler supports Word tables, it does not support table borders or shaded cell backgrounds. For more information on adding custom tables to a WinHelp file, see [Inserting Objects to Create Special Effects in Word Documents](#) (page 366).

16. When you're finished, close the Help file.

For more information on inserting and formatting standard tables, see [Inserting Standard Tables with Doc-To-Help](#) (page 368).

Adding Topic Links and Jumps

One of the keys to a good Help file is providing the reader with auxiliary information on specific topics by using links and jumps. With the click of a mouse, the reader can view additional reference materials on any number of related subjects.

Using [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) is the recommended way for creating topic links and jumps. An alternative method in Word is to use the **Add Topic Link** dialog box. The steps below demonstrate how to create several jumps, including pop-ups and jumps to separately defined windows using both methods.

Adding a Jump

Using D2HML

Using D2HML, you can create a topic link, or jump, by clicking the **Jump Topic Link** button on the [D2HML Styles](#) (page 233) toolbar.

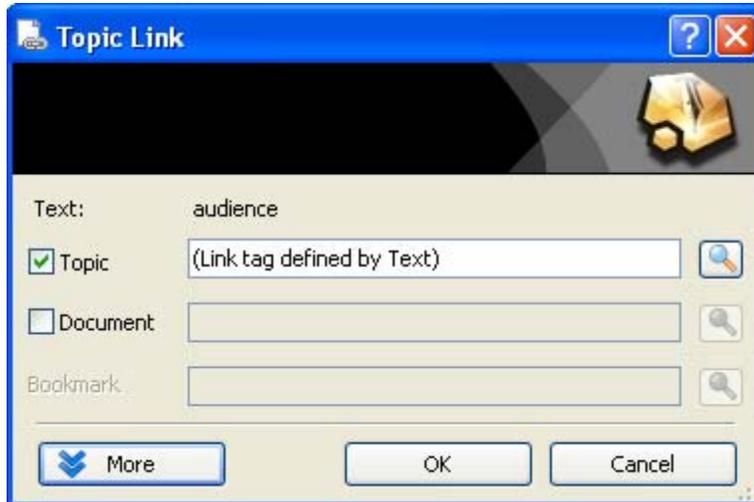
1. In the Icon Bar, click the **Topics** icon.
2. In the right pane topics list scroll down until you find the **Identify your audience** topic.
3. Select, then right-click the **Identify your audience** topic and select **Open** from the shortcut menu to open the StyleGuide.doc in Word at that topic.

4. Select the text *audience* as shown below.

Identify your audience

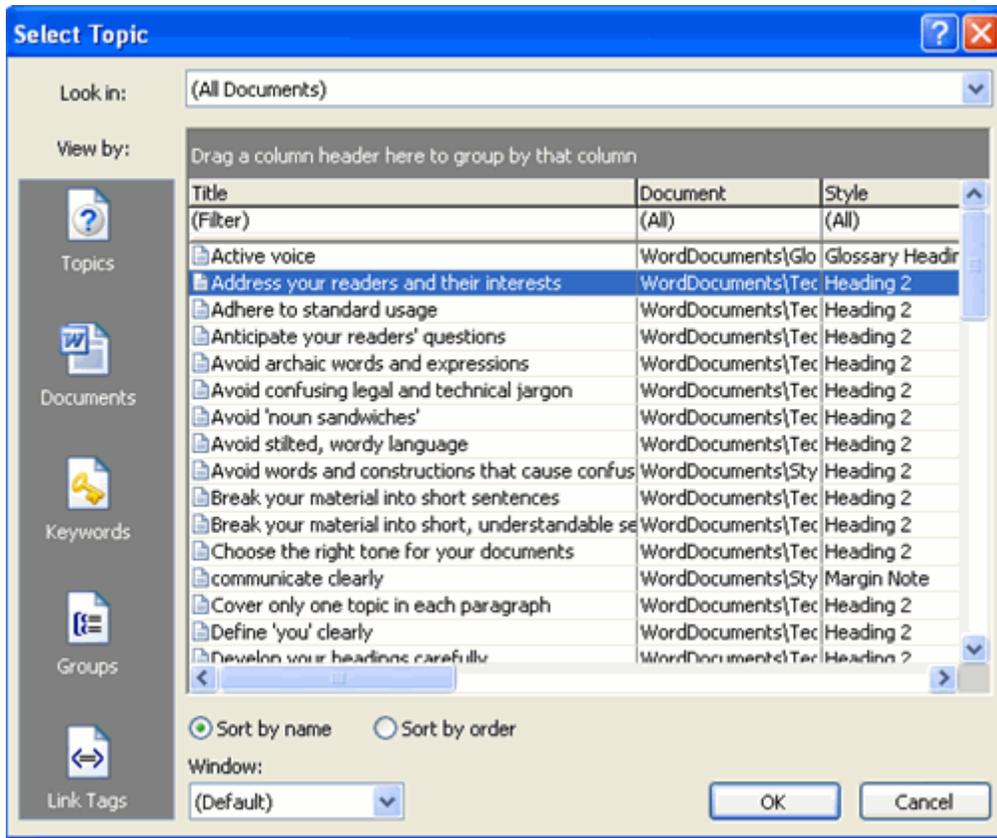
Identify your **audience** early and think about why the reader needs to read the document. Identify people who will be interested, even if they are not directly affected. Write to everyone who is interested, not just to technical or legal experts. Keep in mind the average reader's level of technical expertise.

5. On the **D2HML Styles** toolbar, click the **Jump Topic Link** button  to open the **Topic Link** dialog box.



Note: If the topic you are linking to matches the link text, you can simply click **OK** and the link is created.

6. Click the **Browse** button next to the **Topic** text box. The **Select Topic** window appears.
7. Select *Address your readers and their interests*.

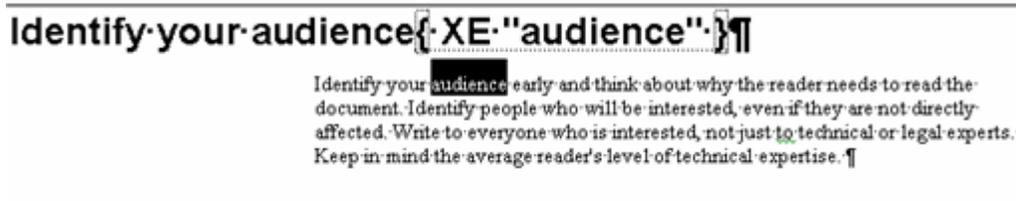


- Click **OK** to close the **Select Topic** window, and click **OK** again to close the **Topic Link** dialog box.

Using Add Topic Link Button

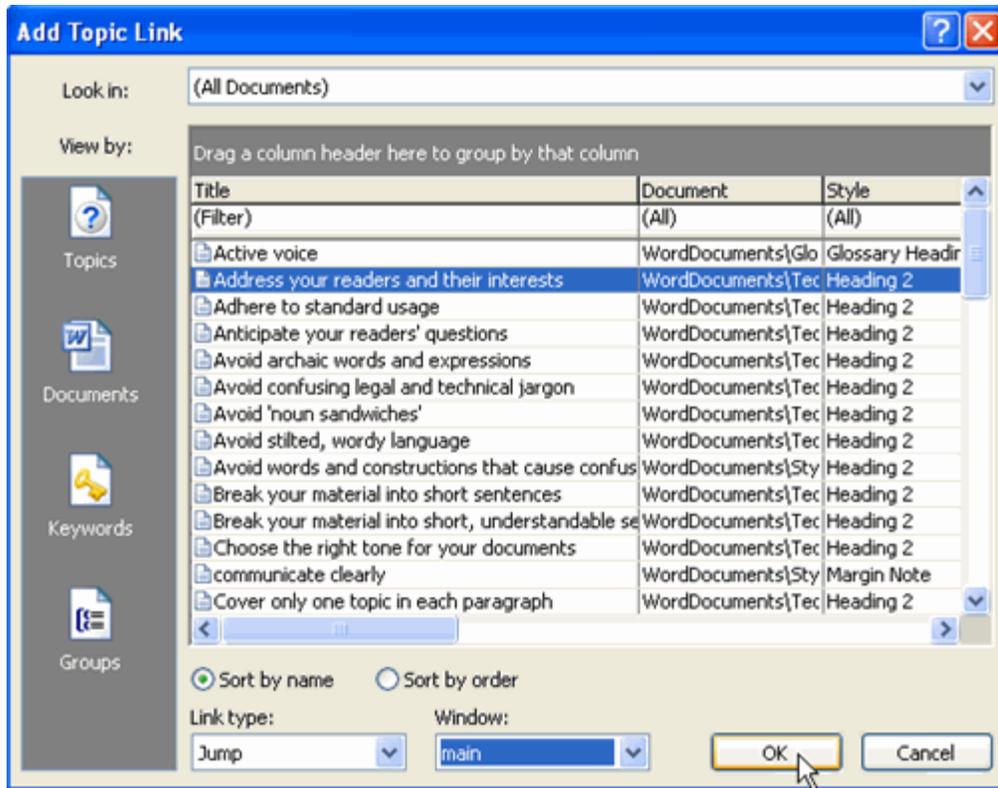
You also have the option of using the **Add Topic Link** button in Word to create a topic link or jump. You must use the **Add Topic Link** dialog box to select a topic, unlike D2HML, where you can simply click the **Jump Topic Link** button and click **OK** if the link text matches the topic to which you are linking.

- In the Icon Bar, click the **Topics** icon.
- In the right pane topics list scroll down until you find the **Identify your audience** topic.
- Select, then right-click the **Identify your audience** topic and select **Open** from the shortcut menu to open the StyleGuide.doc in Word at that topic.
- Highlight the word "audience" as shown below.



- On the **Doc-To-Help** toolbar in Word, click the **Add Topic Link** button  to open the **Add Topic Link** dialog box.
- In the Title column, select the **Address your readers and their interests** topic.
- Select **Jump** from the **Link Type** drop-down.
- Select **Main** from the **Window:** drop-down.

9. Click **OK**.



Note: If your hidden text is turned on, you will notice that Doc-To-Help places a [Help1] comment beside the word you had highlighted.

Adding a Popup

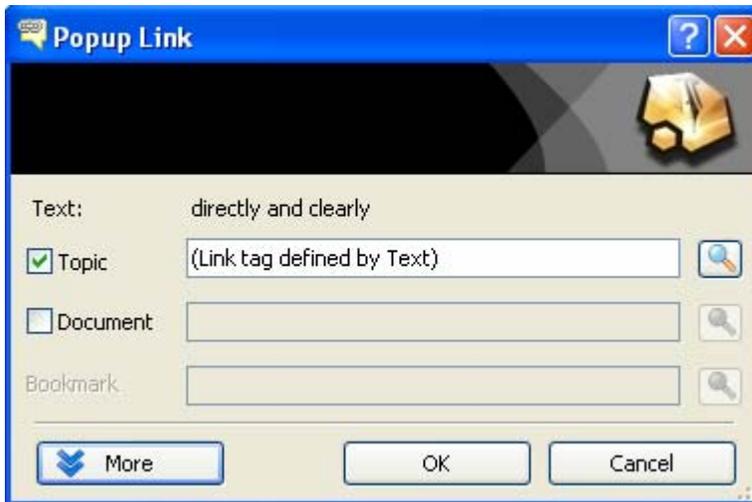
Using D2HML

1. Within the StyleGuide document, locate the topic **How can you engage your readers**.
2. In the paragraph that follows **How can you engage your readers**, select the text *directly and clearly* as shown below.

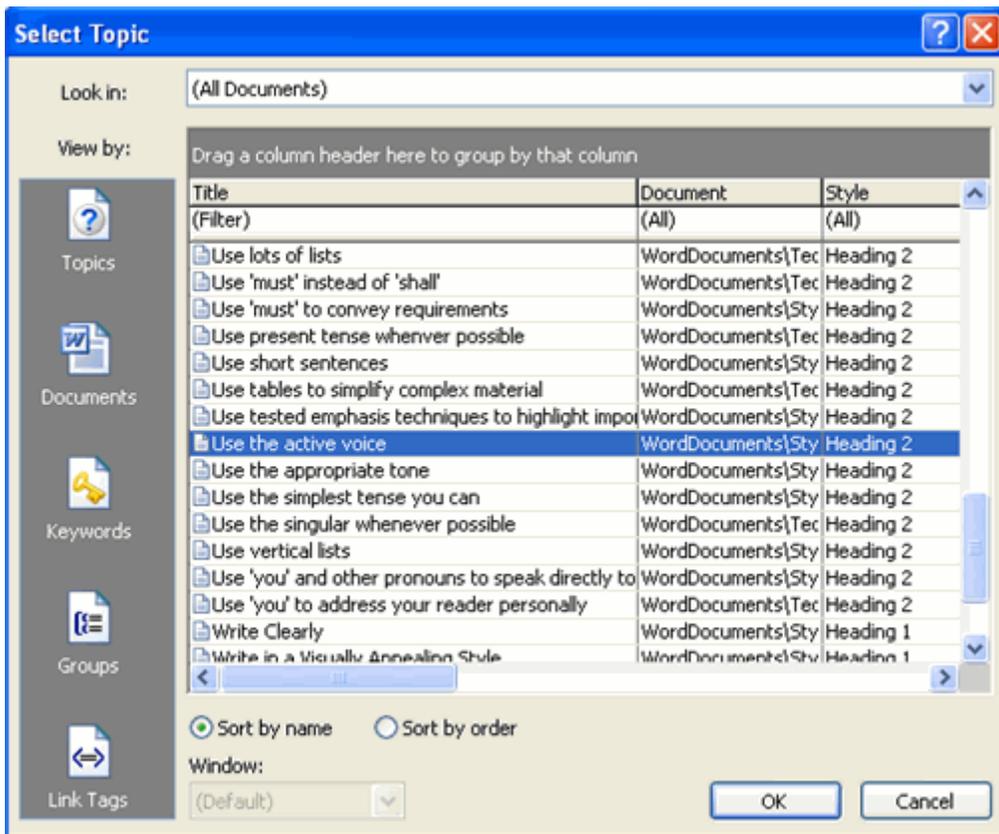
How can you engage your readers?

You engage readers by speaking to them **directly and clearly** and by organizing your message in a structure that reflects their interests. Here are some key ways--

3. Click the **Popup Topic Link** button to open the **Popup Link** dialog box.



4. Click the **Browse** button next to the **Topic** text box, and the **Select Topic** dialog box appears.
5. Select the "Use the active voice" topic.



6. Click **OK** to close the **Select Topic** window, and click **OK** again to close the **Popup Link** dialog box.

Using Add Topic Link

1. Within the StyleGuide document, locate the paragraph with the title, "How can you engage your readers?".
2. In the paragraph that follows, highlight the phrase "directly and clearly" as show below.

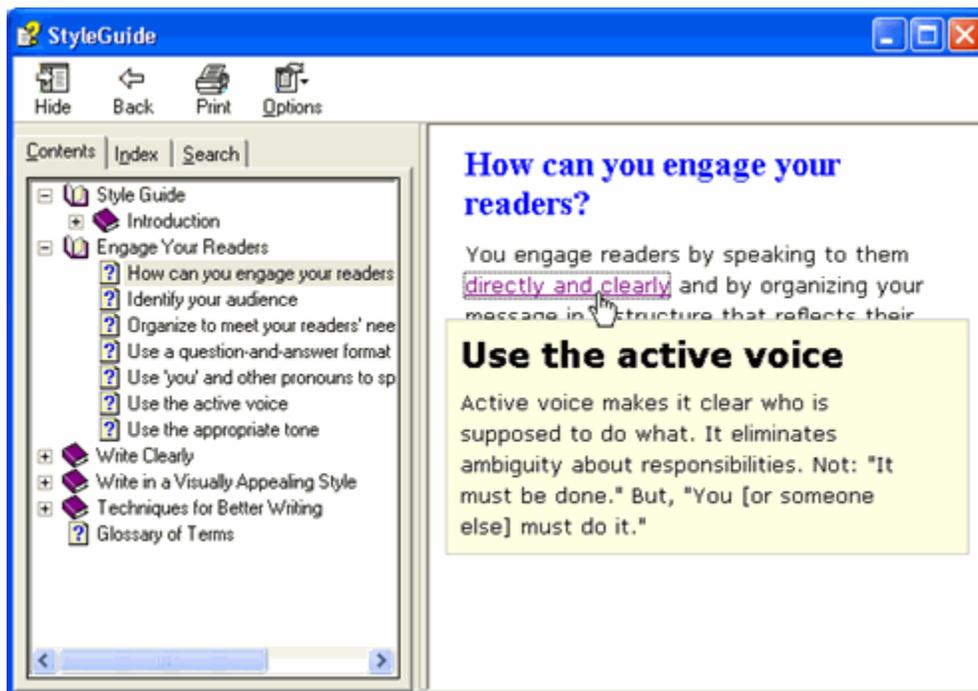
How can you engage your readers?¶

You engage readers by speaking to them directly and clearly and by organizing your message in a structure that reflects their interests. Here are some key ways--¶

3. Click the **Add Topic Link** button.
4. Select the **Use the active voice** topic.
5. Select **Popup** from the **Link Type** drop down and click **OK**.
6. Save the StyleGuide.doc and close Word.

Viewing the Hyperlinks

1. Restore the project editor and click the **Make Target** button.
2. Once the compile is complete, click the **View Target** button.
3. Choose the "How can you engage your readers?" topic and click the **directly and clearly** hyperlink. A pop-up window displays the "Use the active voice" topic text.



4. Click outside the pop-up to close it.
5. In the Help contents pane, select the "Identify your audience" subtopic.
6. Click the **audience** hyperlink to open the **Address your readers and their interests** topic in the right Help pane.
7. When you're finished, close the Help file.

For more information, see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) and [Links and Hot Spots](#) (page 309).

Adding Graphics

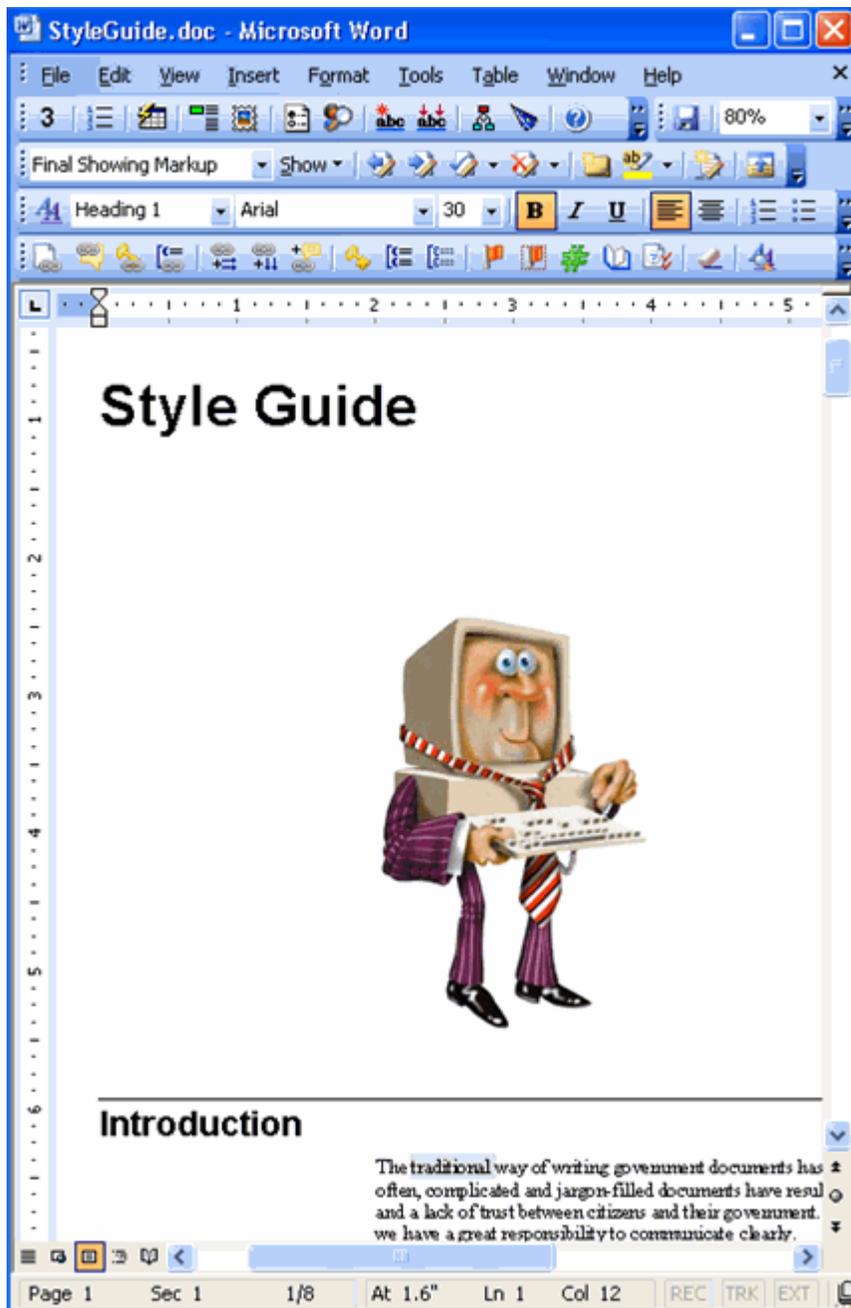
When using Doc-To-Help 2006, it is no longer necessary to scale an image in a graphics package then insert it into the Help document. Doc-To-Help allows you to scale your graphics by using the functionality contained within Microsoft Word.

You may have noticed that we start with a blank Help window pane whenever we open the Help file. We can modify this by changing one of the topics to the **Default Topic**.

In the following steps, let's add a graphic, scale it and set our Style Guide topic as the default topic.

Inserting Graphics

1. Click the **Project** icon.
2. In the left pane, select **Documents**.
3. Select StyleGuide.doc, right-click, and select **Open** from the shortcut menu.
4. Place your cursor at the end of the Heading 1 topic "Style Guide" and press **Enter**.
5. Format the new paragraph as **Body Text**.
6. Select **Insert | Picture | From File**.
7. Locate the file C:\Program Files\ComponentOne\DocToHelp\Tutorial\StyleGuide\StyleGuide.jpg.
8. Select the file, then click **Insert**.



Scaling Graphics

1. Select the graphic you just inserted, right-click, and select **Format Picture** from the shortcut menu.
2. Select the **Size** tab.
3. In the Scale area, change **Height** and **Width** to **75%**.
4. Click **OK**.
5. Save and close the Word document.
6. In the Project Editor, click the **Project** icon.
7. Select **Help Targets** from the left pane.

8. Select the correct Help Target from the right pane. In the property pane, verify that the [ScaleGraphics](#) property is set to *Scale in build*.
9. Select **Paragraph Styles** from the left pane.
10. Select **Heading 1** from the right pane.
11. In the property pane, set the [TopicType](#) property of Heading 1 from **Contents** to **Conceptual**.
The **Contents** topic type limits text formatted with a Heading 1 style to the table of contents pane of your Help target. By selecting **Conceptual**, you can view your Heading 1 topics and the related text and graphics in the same manner as you would view any of your topics.
12. Click the **Make Target** button to compile the Help target.
13. Open the "Style Guide" topic in the Help file to view the scaled topic.
14. When you're finished, close the Help file.

Working With Margin Notes

The **Margin Note** drop down contains a variety of commands that allow you to create and format your margin notes. For this tutorial we will use the **Create Margin Note** and **Set Link** commands to produce a margin note with hyperlink.

The **Create Margin Note** command allows you to place text or graphics in the left margin, next to the main body of the text. Margin notes do not appear in the Help unless you explicitly link them to the text, in which case they become pop-ups. Text is automatically formatted to appear in Margin Note style, smaller than Body Text and italicized.

The **Set Link** command creates a hyperlink between the text or graphic in the right cell and the margin note in the left cell. If you're linking the margin note to text, that text appears underlined and blue in Help. When the blue, underlined text or the selected graphic is clicked, the margin note displays in a pop-up window.

Creating a Margin Note

1. Click the **Project** icon.
2. Select **Documents** from the left pane.
3. Select StyleGuide.doc from the right pane, right-click and select **Open** from the shortcut menu.
4. Place your cursor anywhere in the first paragraph after the "Introduction" heading.
5. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
6. Select **Create** from the drop-down menu. Doc-To-Help inserts a two-column table with the existing text in the right table cell.

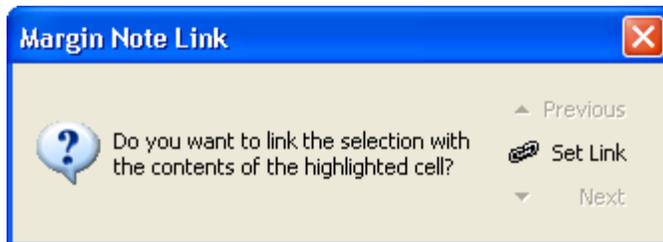
Note: If Gridlines is selected in the **Table** menu, you'll see a dotted outline of the table, but these gridlines do not print.

7. In the left cell, type the margin note "Heading 1 topics can now have text!".

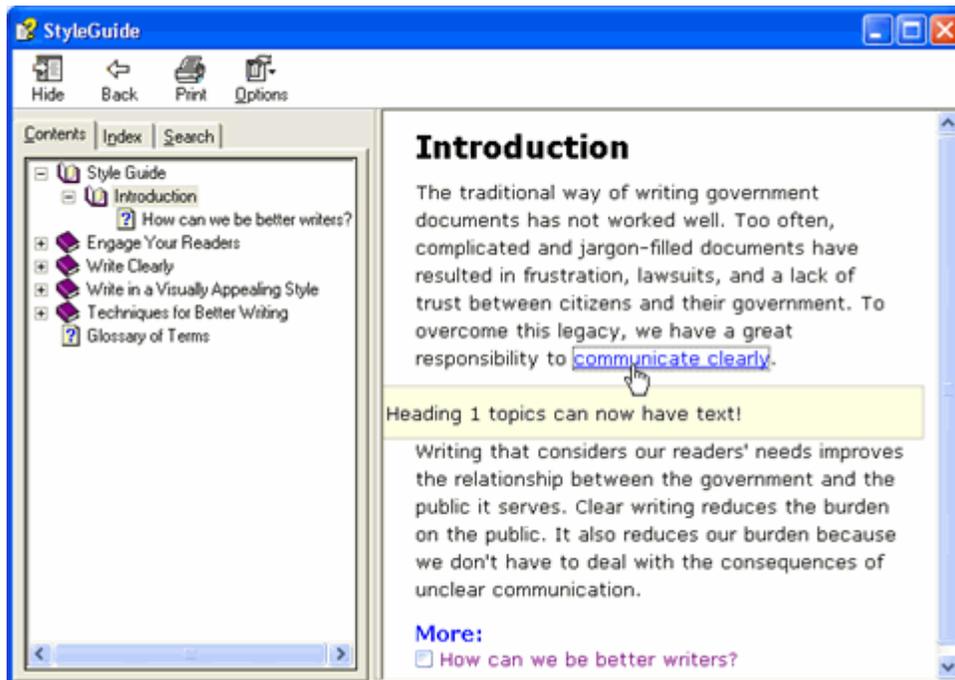
<i>Heading 1 topics can now have text!</i>	The traditional way of writing government documents has not worked well. Too often, complicated and jargon-filled documents have resulted in frustration, lawsuits, and a lack of trust between citizens and their government. To overcome this legacy, we have a great responsibility to communicate clearly.
--	--

Linking a Margin Note

1. In the right cell, highlight the "communicate clearly" body text at the end of the paragraph.
2. Click the **Margin Note** button on the **Doc-To-Help** toolbar.
3. Select **Set Link** from the drop-down menu. The Margin Note Link box opens. If you have more than one margin note, you can select the note you want by using the **Previous** and **Next** icons.



4. Since we only have only one margin note, click the **Set Link** icon. Doc-To-Help places a Word comment next to the link text.
5. Save and close the Word document.
6. Click the **Make Target** button.
7. After the target is compiled, click the **View Target** button.
8. Open the "Introduction" topic, then click the communicate clearly hyperlink to see the pop-up.



9. When you're finished, close the Help file.

For more information, see [Using Margin Notes in Word](#) (page 323).

Adding Glossary Terms

To assist in developing an online and printed glossary, Doc-To-Help automatically creates an empty glossary document whenever a new project is created. If you format your glossary entries with the **Glossary Heading** paragraph style, Doc-To-Help examines the project documents for text that matches the glossary entries, converting these matches

into hyperlinks when you build your Help target. Whenever the user clicks on one of the hyperlinks, a pop-up opens containing the corresponding glossary entry description.

By opening and adding entries to the glossary document within Doc-To-Help, you can quickly create and update your glossary as needed.

1. If necessary, click the **Project** icon.
2. Select **Documents** from the left pane.
3. Select Glossary.doc from the right pane, right-click and select **Open** from the shortcut menu.
4. Add the following text in the glossary document beneath the Glossary heading:

```
Active voice¶  
A style of writing in which the subject of a sentence performs or causes the action expressed by the verb.¶  
Jargon¶  
A special vocabulary for a particular group.¶
```

5. Format "active voice" and "jargon" with the Glossary Heading style.

Note: The **Glossary Heading** and **Glossary Heading (no auto links)** paragraph styles are reserved for glossary entries.

6. Format the other text with the **C1H Popup Topic Text** style.

When steps 4 – 6 are complete, the glossary text should look like the following example.

Active voice

A style of writing in which the subject of a sentence performs or causes the action expressed by the verb.

Jargon

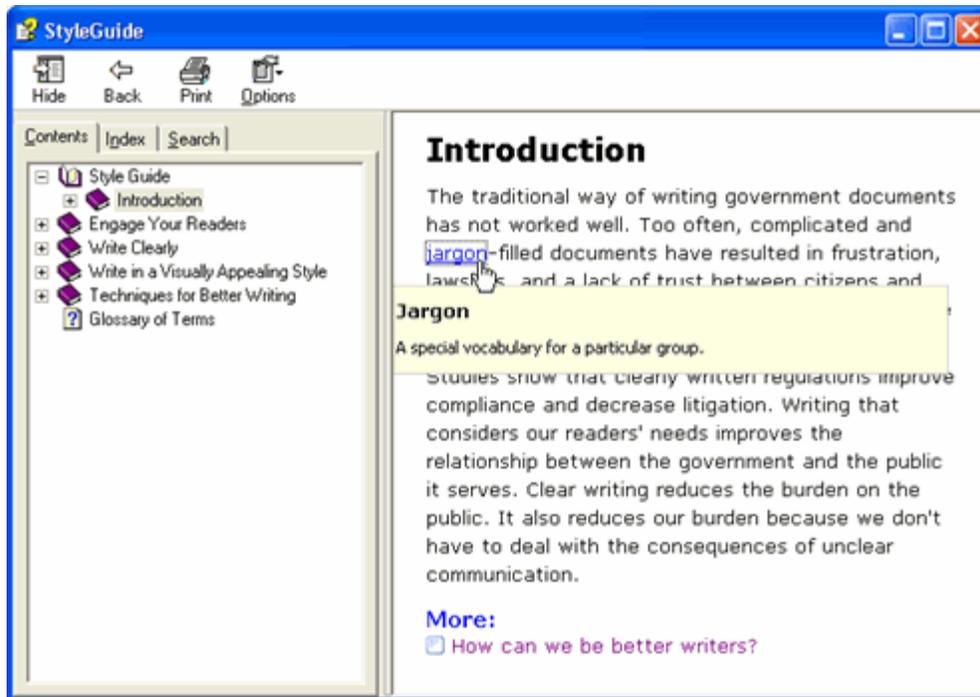
A special vocabulary for a particular group.

7. Save and close the Word document.
8. Restore the project editor.
9. Select **Build | Rebuild Target**.

Note: The **Make Target** command compiles only the source documents that have been modified since the last build. The **Rebuild Target** compiles all source documents. By selecting **Rebuild Target**, Doc-To-Help scans all source documents for the new glossary terms.

10. After the target is compiled, click the **View Target** button.

Wherever the Help text matches your glossary entries (Glossary Heading), Doc-To-Help has formatted the text as a pop-up.



11. When you're finished, close the Help file.

For more information, see [Creating a Glossary](#) (page 333).

Adding Conditional Text

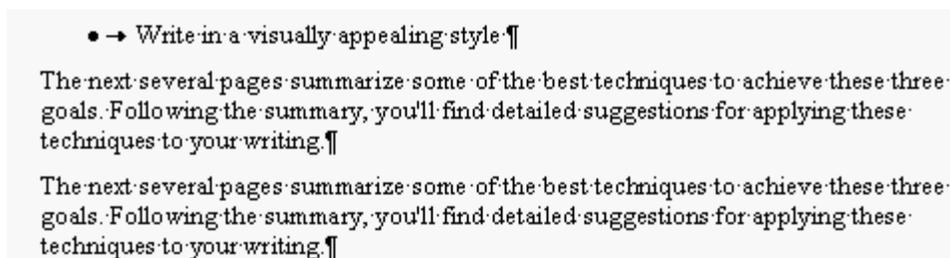
The conditional text feature allows you to determine the conditions under which certain text or graphics are incorporated into your Help targets.

When defining conditional text, Doc-To-Help provides three target options; **Platforms**, **Help Targets** and **Attributes**.

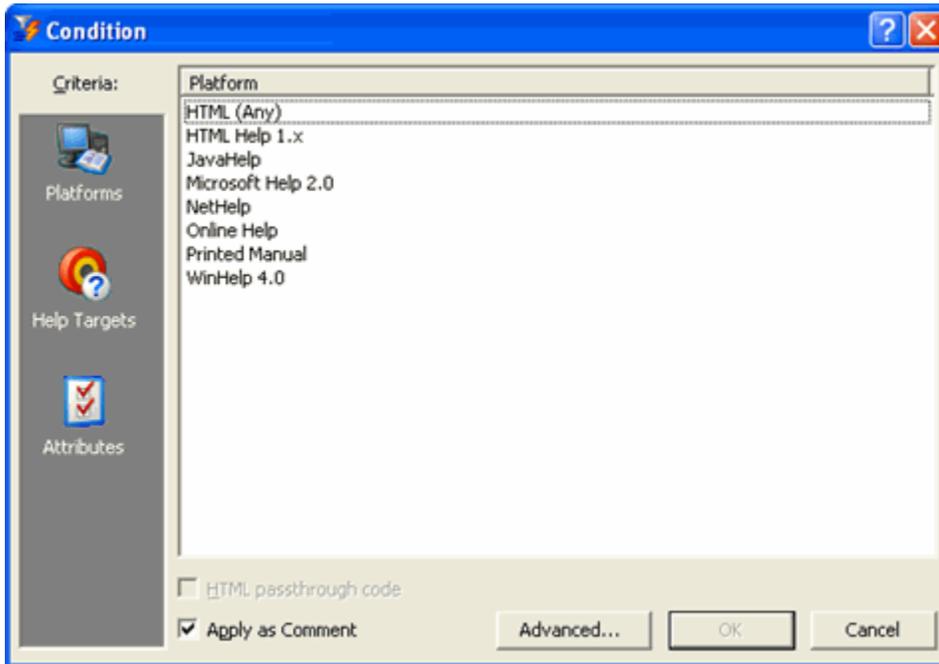
Platforms	This option allows you to incorporate conditional text into any Help file defined with a specific platform.
Help Targets	This option allows you to incorporate conditional text into a specific default or user defined Help target.
Attributes	This option allows you to incorporate conditional text into a specific version of a Help file.

Applying Conditional Text to Printed Manuals

1. Select StyleGuide.doc from the right pane, right-click and select **Open** from the shortcut menu.
2. Copy the last paragraph of the Introduction topic and insert it beneath the existing last paragraph so that you have two duplicate paragraphs as shown below.



3. Highlight the first duplicate paragraph.
4. Click the **Conditional Text** button  on the **D2HML Styles** toolbar. The **Condition** dialog box opens.



5. From the **Platform** pane, select **Printed Manual** then click **OK**. Doc-To-Help places a conditional text D2HML comment note after the first paragraph.

Applying Conditional Text to Online Help

Next, you'll change the second duplicate paragraph so that it will "fit" better as part of the online Help.

1. Change the second paragraph from:

The next several pages summarize some of the best techniques to achieve these three goals. Following the summary, you'll find detailed suggestions for applying these techniques to your writing.

To:

The following three topics summarize some of the best techniques to achieve these three goals. Following the summary topics, you'll find detailed suggestions for applying these techniques to your writing.

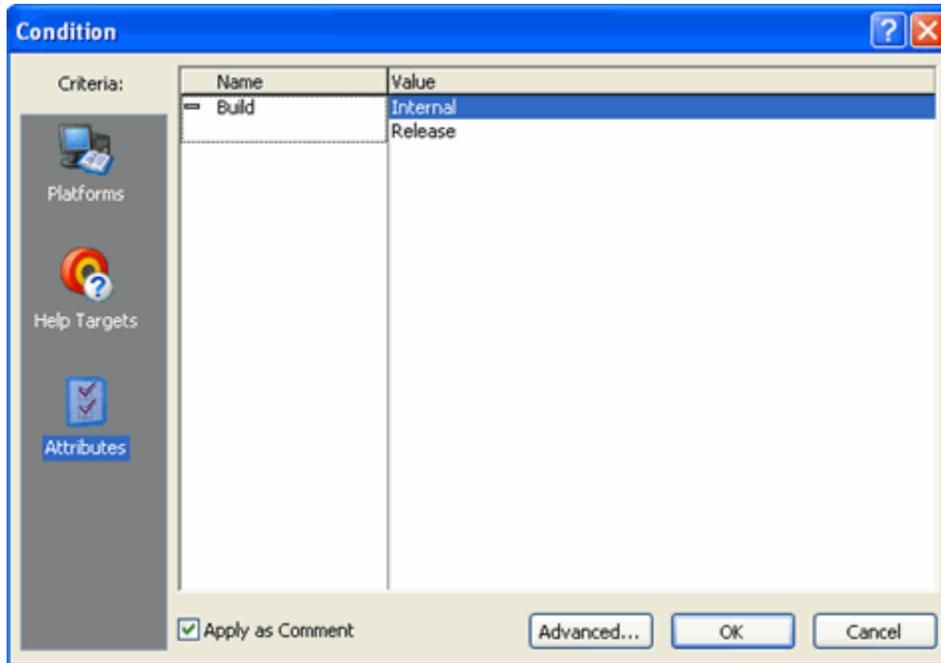
2. Highlight the modified paragraph.
3. Click the **Online Text** button  on the **D2HML Styles** toolbar. Doc-To-Help places another conditional text D2HML comment after the second paragraph.

Applying Conditional Text with Attributes

In some cases you may have two Help files using the same Help target, but with somewhat different text. To do this, you can set the conditional text by attribute.

1. Create an empty paragraph immediately above the "Introduction" topic.
2. Place your cursor in the empty paragraph, and type "Version History".
3. Apply the **Heading 2** style to the "Version History" text.

4. Highlight the "Version History" heading.
5. Click the **Conditional Text** button  on the **D2HML Styles** toolbar. The **Condition** window appears.
6. In the **Criteria:** pane, click the **Attributes** icon.
7. From the **Value** pane, select **Internal**.

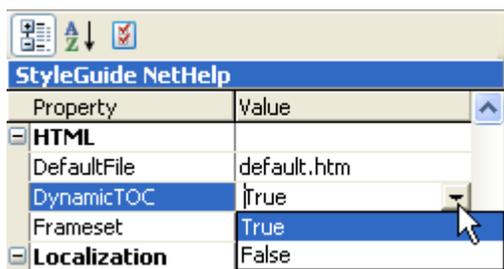


8. Click **OK**.
9. Save and close the Word document.

Building the Conditional Help

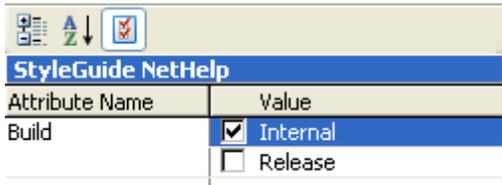
In the following steps, you will set the Help project properties and attributes, and compile your source documents for both HTML and HTML Help.

1. If necessary, click the **Project** icon.
2. Select the **Help Targets** item in the left pane.
3. Choose the **StyleGuide NetHelp** target from the right pane.
4. In the **StyleGuide NetHelp** properties pane, make sure that the [DynamicTOC](#) is set to **True**.



Though this property is not necessary to create the conditional text, it provides a Dynamic HTML table of contents for our NetHelp target. Note that this feature is only supported by Internet Explorer.

5. Click the **View Attribute** icon  in the **StyleGuide NetHelp** properties pane.
6. In the Value column, select **Internal**.



7. Click the **Make Target** button to build the HTML target.
8. After the NetHelp target is compiled, select **StyleGuide HTML Help** from the Help Target drop-down list.
9. Click the **Make Target** button again. In this way, you have made sure that both the NetHelp and HTML Help targets have been compiled.
10. After both targets are compiled, click the **View Target** button to open the HTML Help file.

Notice that the online Help no longer contains the paragraph marked as manual or the Version History topic. If we were to build the printed manual Help target, the paragraph marked manual would be used and the paragraph marked online and the Version History topic would be omitted.

11. Close the HTML Help.
12. Select **StyleGuide NetHelp** from the Help Target drop-down list.
13. Click the **View Target** button. The Help file opens in your browser. By examining the NetHelp file you will notice that the Version History topic and the paragraph marked online are included. The left pane contains a Dynamic HTML table of contents with books containing subtopics.

For more information, see [Conditional Text and Attributes](#) (page 401).

Applying Lists in Doc-To-Help

To assist in developing lists, Doc-To-Help provides six list styles to allow you to create nested lists in your documentation.

Creating a Bulleted List

1. On the Icon bar, click **Project**.
2. In the left pane, select **Documents**.
3. Select and right-click StyleGuide.doc from the right pane, and select **Open** from the shortcut menu.
4. Scroll down the document until you locate the "Use vertical lists" section.
5. Highlight the text as show below.

```
lists--||
Highlight levels of importance||
Help the reader understand the order in which things happen||
Make it easy for the reader to identify all necessary steps in a process||
Add blank space for easy reading||
```

6. On the **Doc-To-Help** toolbar, click the **C1H Bullet Style** button . Doc-To-Help applies a bulleted list as shown below.

- lists--¶
- Highlight levels of importance¶
 - Help the reader understand the order in which things happen¶
 - Make it easy for the reader to identify all necessary steps in a process¶
 - Add blank space for easy reading¶

Nesting a Bulleted List

1. Place your cursor at the end of the second bulleted line and press **Enter**.
2. On the new line, type **Step One is first** and press **Enter**.
3. On the second new line, type **Step Two is second** and press **Enter**.
4. On the third new line, type **This is how you get "nested" lists**. You should have a bulleted list that looks like the figure below.

- lists--¶
- Highlight levels of importance¶
 - Help the reader understand the order in which things happen¶
 - Step One is first¶
 - Step Two is second¶
 - This is how you get "nested" lists¶
 - Make it easy for the reader to identify all necessary steps in a process¶
 - Add blank space for easy reading¶

5. Highlight the lines that read "Step One is first" and "Step Two is second".
6. Click the **C1H Number 2 Style** button . Doc-To-Help applies a nested numbered list as show below.

- lists--¶
- Highlight levels of importance¶
 - Help the reader understand the order in which things happen¶
 - 1.→ Step One is first¶
 - 2.→ Step Two is second¶
 - This is how you get "nested" lists¶
 - Make it easy for the reader to identify all necessary steps in a process¶
 - Add blank space for easy reading¶

7. Highlight the line that reads "This is how you get nested lists".

8. Click the **Continue List** button . Doc-To-Help applies an unnumbered list at the same level as the previous numbered list.

```
lists--||
    •→ Highlight levels of importance||
    •→ Help the reader understand the order in which things happen||
        1.→ Step One is first||
        2.→ Step Two is second||
        This is how you get "nested" lists||
    •→ Make it easy for the reader to identify all necessary steps in a process||
    •→ Add blank space for easy reading||
```

9. Save and close the StyleGuide document.
10. Click the **Make Target** button.
11. After the target is compiled, click the **View Target** button.
12. Open the "Use vertical lists" topic to see your finished nested list.
13. When you're finished, close the Help file.

For more information, see [Using List Styles in Microsoft Word](#) (page 229).

Using HTML Source Documents

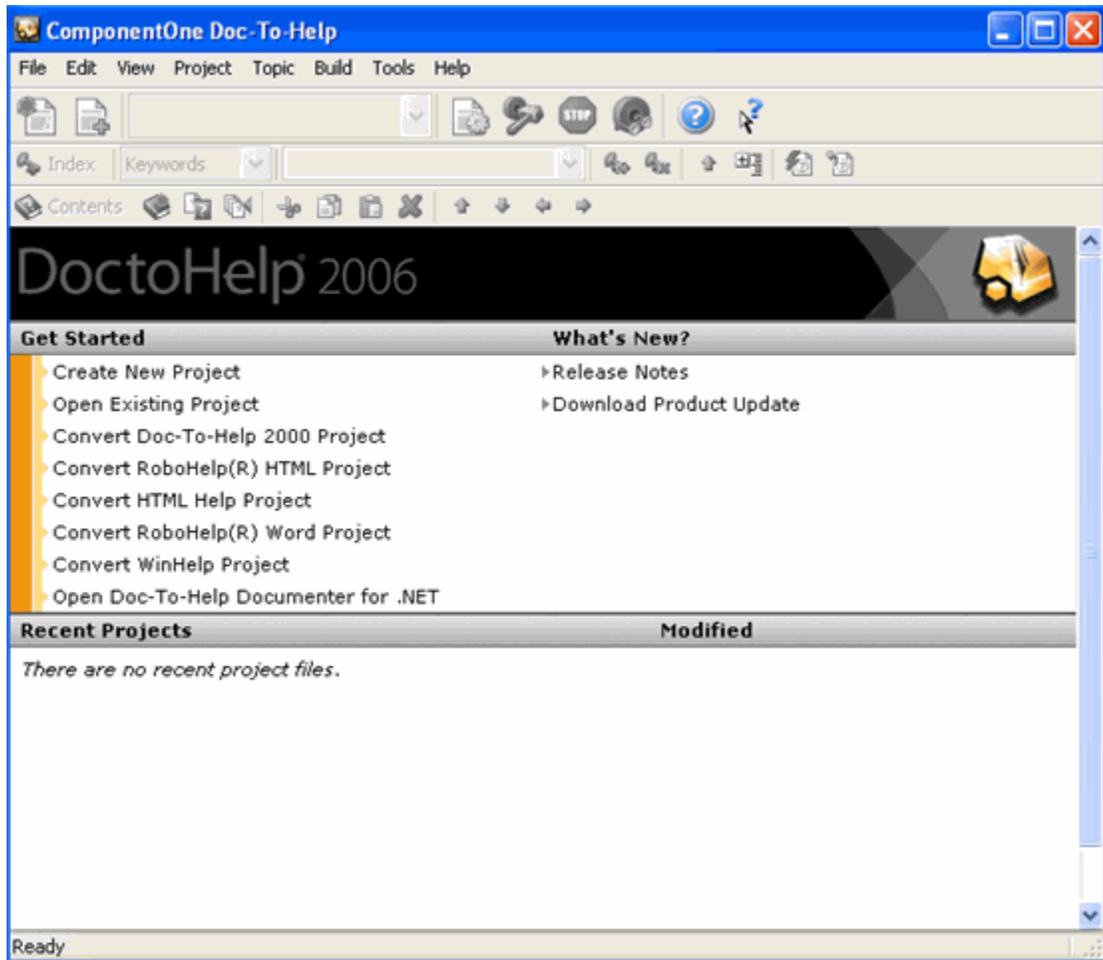
Any HTML editor can be used to format your HTML source documents, but the guided tour uses Microsoft FrontPage 2003. The steps may be slightly different, depending on the HTML editor you are using.

This tour assumes the new project will be in a folder called "c:\GuidedTour". Create this folder, or you may use a different folder if you prefer.

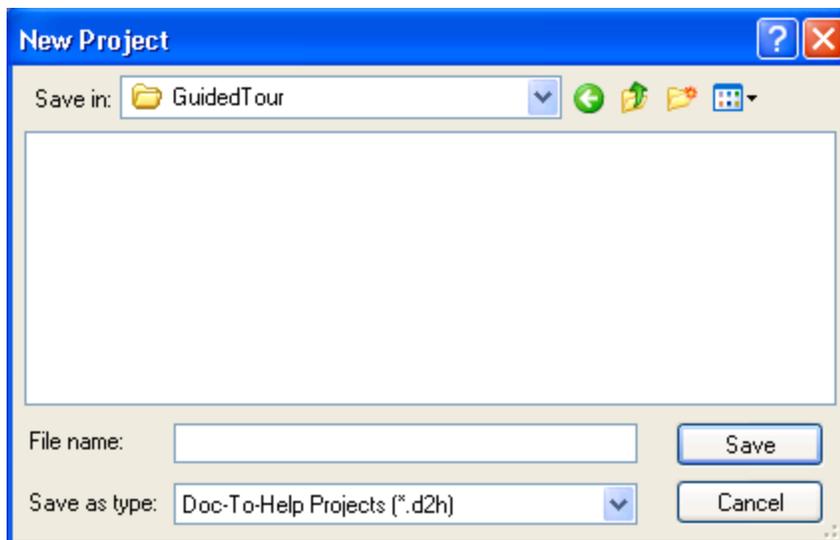
Starting a New Help Project

The first step in starting a new Help project is to name the project file and select a folder where it will reside. Doc-To-Help 2006 provides you with an "all in one" project editor which allows you to create and edit source documents, add Help features, and build Help files for any platform.

1. Open the Doc-To-Help program. You are presented with the Doc-To-Help start page. From here you can choose some of your most recent projects, open an existing project, convert an older Doc-To-Help project, convert a RoboHelp or HTML Help project, or create a new project.

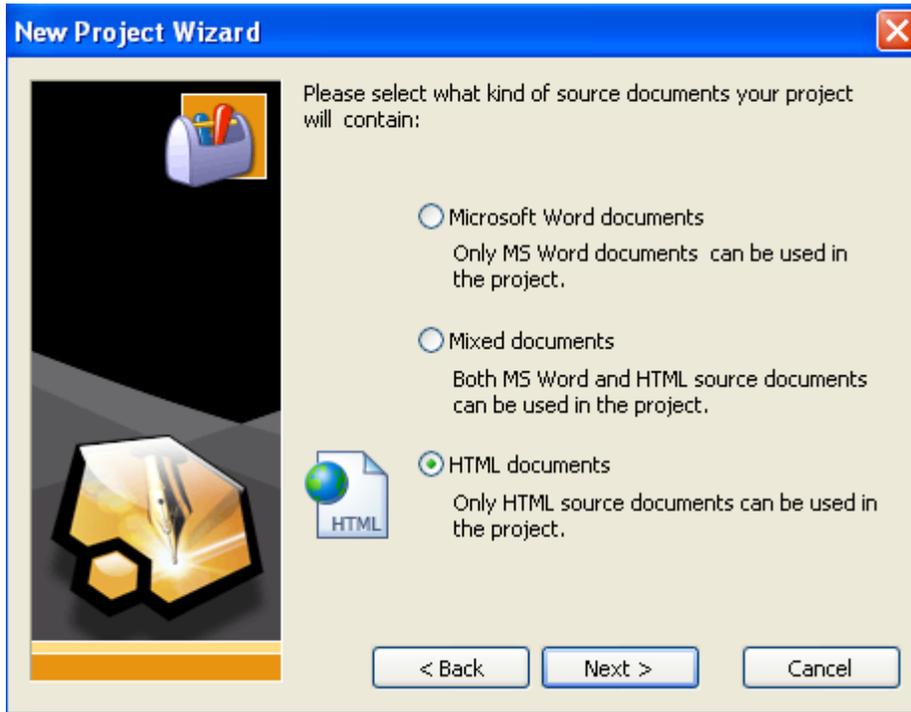


2. Click **Create New Project**. The first screen of the New Project Wizard opens.
3. Click **Browse**. The **New Project** dialog box opens as shown below.



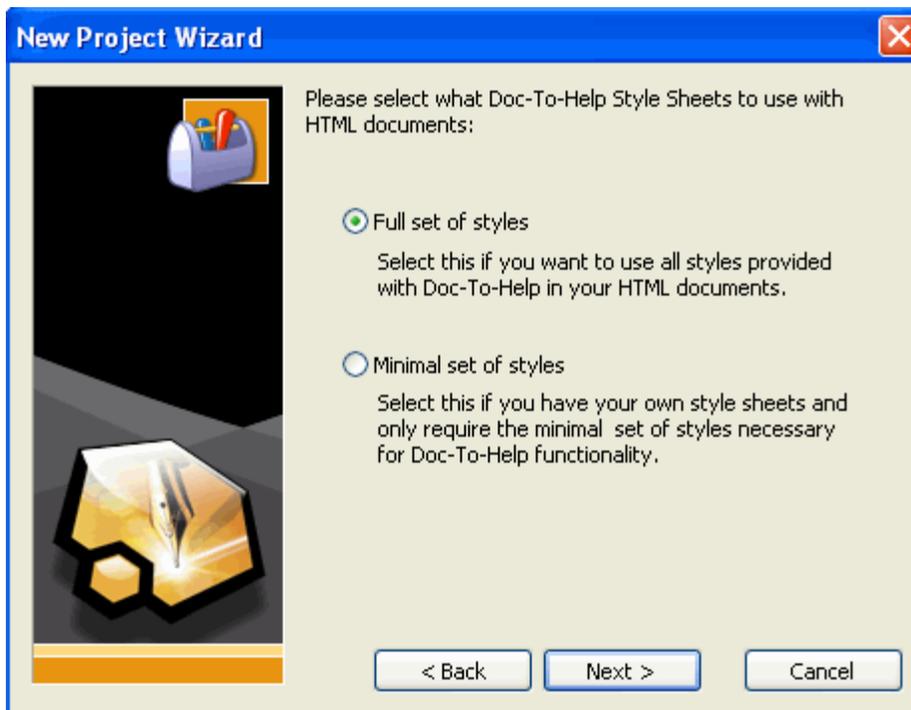
4. Open the C:\GuidedTour folder.
5. Type **Scuba Dive.d2h** in the **File name** textbox, then click **Save**.

6. Click **Next**.
7. Select **HTML documents** as the type of source documents to use in your project and click **Next**.

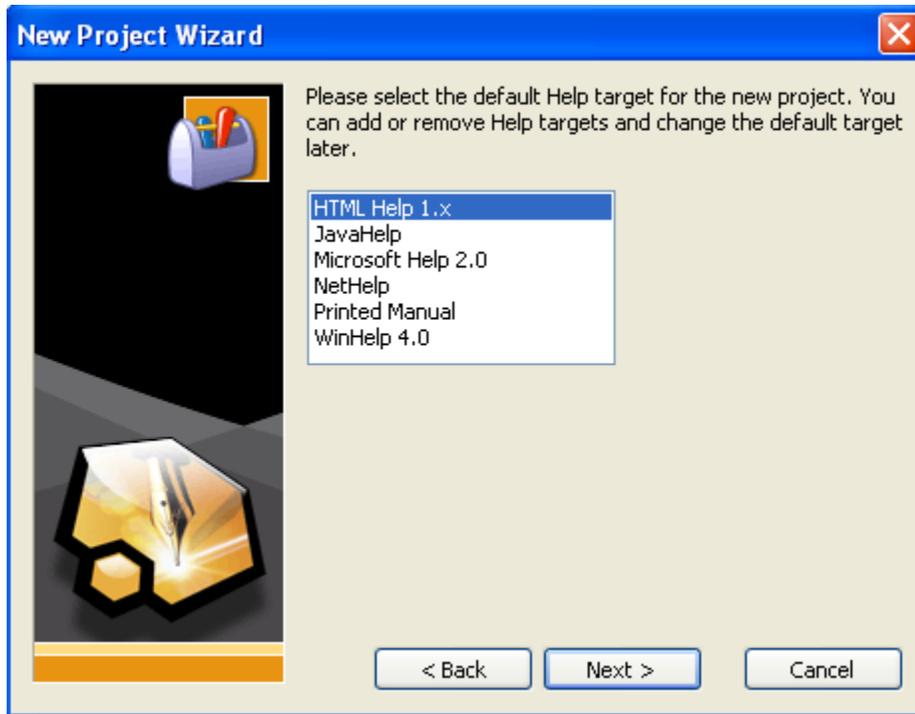


Note: You can change the type of source documents your project uses at any time by changing the [DocumentType](#) property under Project Settings.

8. You are prompted to select the Doc-To-Help Style Sheets you want to use with the HTML documents. Select **Full set of styles** to use all the styles provided with Doc-To-Help and click **Next**.



- From the list, choose your default Help target and click **Next**. Make sure that **HTML Help 1.x** is selected.

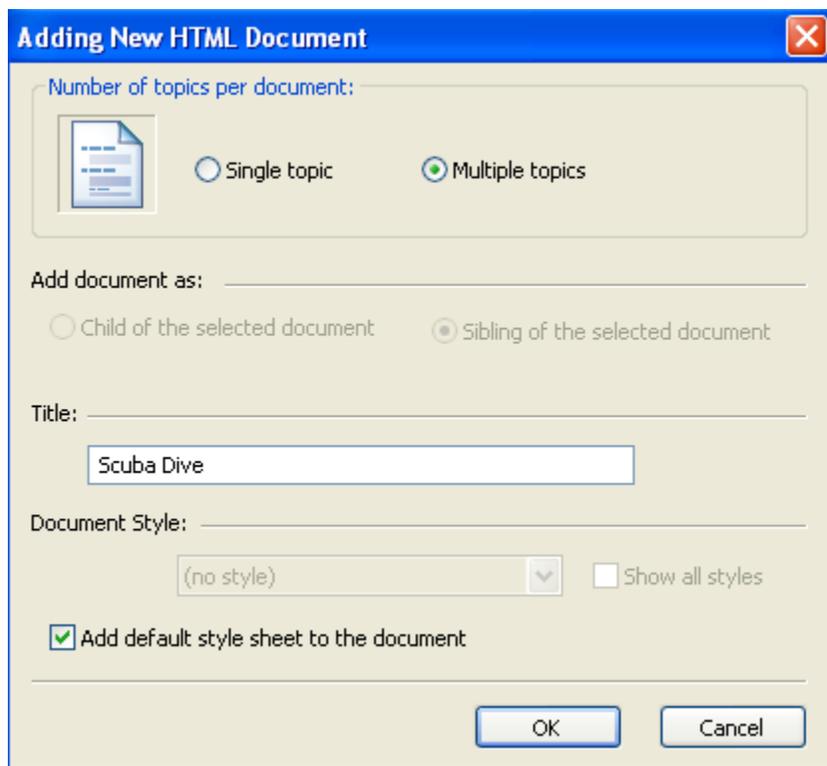


Note: Though you can build any Help target for this project, this selection determines which Help target your project compiles initially.

- Review your directory location and file name and click **Finish**.

The Doc-To-Help Wizard produces a new Help project and opens the project editor. A **New Document** dialog box appears, and **Scuba Dive.htm** appears in the **File name** text box.

- Click **Save**. The **Adding New HTML Document** dialog box appears.



- Select **Multiple topics** if it is not already selected.
- Enter **Scuba Dive** in the **Title** text box, which specifies the first topic in the document.
- **Add default style sheet to the document** is checked by default. Make sure it is selected.

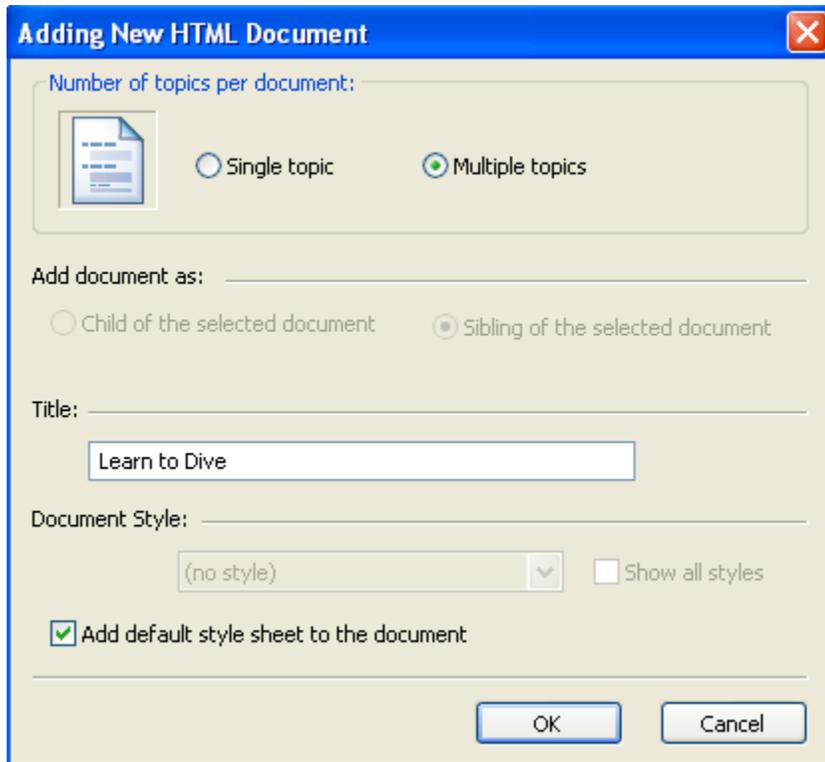
12. Click **OK** to accept the settings.

The Scuba Dive.htm page is opened in your default HTML editor and you can begin adding content to the project. You can close this document for now; we will open it again in a later section.

Adding Content to the Project

The Doc-To-Help project editor lists all of the documents within your project in the document tree in the right pane. You will see the Scuba Dive.htm document and a Glossary.htm document that was automatically added when the project was created. We are going to add some additional documents.

1. If necessary, click the **Project** icon and select the **Documents** item from the left pane Tree View.
2. Right-click anywhere in the right pane and select **New Document**.
3. Enter **Learn to Dive** in the **File name** text box and click **Save**. The **Adding New HTML Document** dialog box appears.
4. Select **Multiple topics** under the *Number of topics per document* if it is not already selected.



5. Click **OK** to accept the remaining default properties for the document and add it to the project.
6. Repeat this process to add another file to the project. Name it **Equipment**. At this point, the project contains four empty documents.

Note: You can change the order of the documents by dragging them up and down the list or by using the arrow buttons above the list.

Project Contents and Structure

Doc-to-Help projects may contain two types of documents:

- **Multiple-Topic documents**

Multiple-topic documents have an internal hierarchy determined by the styles used in the document. They should start with a "Heading 1" title, followed by other headings and body text. Each paragraph formatted with a heading is displayed as a separate topic in the help system.

Multi-topic documents are always at the top level in the project hierarchy. They cannot be moved to deeper levels.

- **Single-Topic documents**

Single-topic documents have no internal hierarchy. Each file is displayed as a single topic in the help system.

The position and hierarchy of the topic within the project is determined by its position in document tree and by the style assigned to the document in the project (the styles actually used in single-topic documents do not affect the structure of the help system, only the appearance of the topic).

For additional information, see [Single and Multiple Topic Documents](#) (page 183).

Changing a document from single to multi topic (or vice versa) is easy, but it affects the overall structure of the help system. So it is a good idea to plan the project structure before you start adding content to the documents.

The decision between using single and multi-topic documents is up to you. If you prefer to work with a few large documents, and the structure of the help system is relatively static, multi-topic documents are better. If you prefer to work with small files and to manage the structure of the help system from Doc-to-Help, then single-topic documents are a better choice.

You may combine single and multi-topic documents in a single project to get the best of both worlds. That is the approach we will take here.

Now let's add some content to our project. Instead of actually writing each topic, copy all of the HTML files from the tutorial folder into the new project folder.

- Copy all HTML files from the tutorial folder (**c:\Program Files\ComponentOne\DocToHelp\Tutorial\ScubaDive\HTMLDocuments**) to the project folder (**c:\Guided Tour\HTMLDocuments**). Replace any existing files.

The files you just copied contain enough content to get the help system started. We will enhance them later with additional formatting, images, links, and more.

Building the Project

Before we start enhancing the "Scuba Dive" help system, let's build the project to see what it looks like now.

To build the project, follow these steps:

1. If necessary, select **View Toolbars | Project** to display the **Project** toolbar in the Doc-To-Help project editor.
2. Check that the **Help Target** list has the type of target you want (we will use HTML Help, but there are many other options).

3. Click the **Make Target**  button.

4. After the project compiles, you can examine your work by click the **View Target**  button on the toolbar.

You should now be looking at the initial version of the help file.

The **Contents** tab shows the topic hierarchy. The **Index** tab is limited at this point (we will add an index later). The **Search** tab is fully functional.

We got a simple but functional help system just by adding some regular Html files to the project and building it. Close the help file when you are finished viewing it.

Before moving on, let's examine the structure of the content files.

Document Structure and Styles

In the Doc-To-Help project editor, double-click the **Learn To Dive.htm** file to open it in your default HTML editor.

The file starts with a "Heading 1" title, immediately followed by a "Heading 2" title, and then some body text. The structure of the help system is determined by the heading styles. By default, the following conventions apply:

- | | |
|-------------------------|--|
| Heading 1 | Creates a "book" entry in the help Contents tab. This entry contains sub-topics, but no topic content of its own. |
| Headings 2, 3 | Creates a topic with contents and possibly more sub-topics. If there are sub-topics, Doc-to-Help automatically adds links after the regular content. |
| Heading 4, 5 | Similar to Heading 2, except the topic title is not added to the "Contents" tree and the content is shown in a separate window. |
| Glossary Heading | Reserved for glossary entries. |

You can edit these conventions by changing the properties of the paragraph styles and topic types.

The same style conventions apply to HTML and Microsoft Word files.

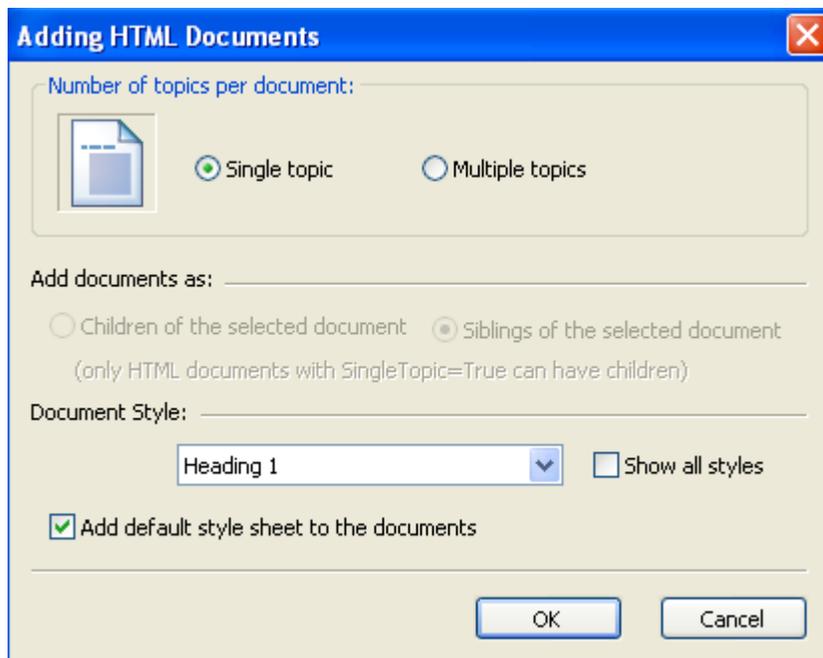
Adding Single-Topic Documents

Multiple topic documents are convenient when you want to organize hierarchical information in a single document.

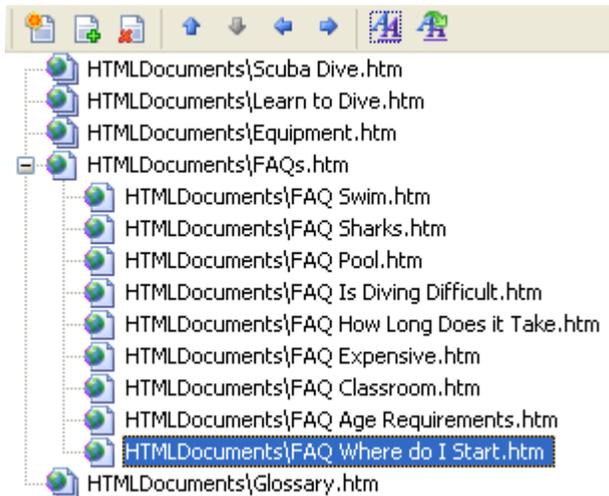
In some cases, though, it may be more convenient to work with multiple small documents. For example, if your help system contains an FAQ (frequently asked questions) section, you may have a team of contributors submitting content. Rather than consolidate all the information into a single file, it may be more convenient to keep it in separate files.

To demonstrate this, let's add some single-topic documents to our project.

1. In the document tree, right-click the **Equipment.htm** file.
2. Select **Add Documents**. The **Add Documents** dialog box appears.
3. Select **FAQs.htm** and click **Open**.
4. Select **Single topic** under *Number of topics per document* if it is not already selected.



5. Click **OK** to accept the settings and add the document to the project.
6. Right-click the **FAQs.htm** and select **Add Documents**.
7. In the **Add Documents** dialog box, select all files that start with "FAQ" and click **Open** (there should be nine files).
8. In the **Adding HTML Documents** dialog box, select the option of adding the files as "Children of the Selected Document" and click **OK**. The document tree should now look like this:



Let's rebuild the project to see the new topics:

1. Click the **View Target** button.
2. Click **Yes** when prompted to rebuild the help file. Notice how the FAQs are now displayed in the help system.

Note: The Contents pane shows the Title assigned to each single-topic document. The initial value used is the file name, which is a logical choice but may not work in all cases. To change the titles, select the document in the document tree, and enter a title next to the [Title](#) property in the Property Pane.

Because we are using single-topic documents in this part of the help project, adding new FAQs to the project amounts to adding individual files to the project. You don't have to modify any of the existing files.

Before moving on to the next topic, let's explore the document tree a little more. Try selecting one of the FAQ documents and moving it deeper into the tree hierarchy.

Now try rebuilding the project. Doc-to-Help will display a warning stating that the Style property of some documents doesn't match their position in the tree. To fix this, you can either:

- Drag the document back to its original position OR
- Use the Move Left arrow on the **Documents** toolbar.

Using Styles to Enhance the Help project

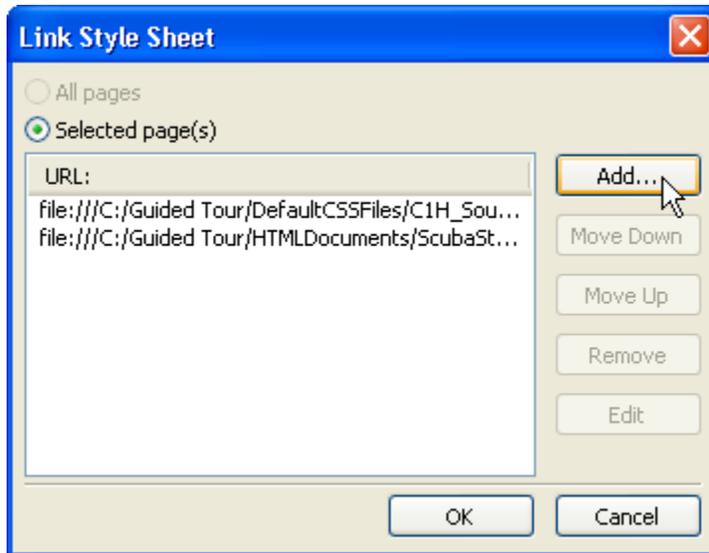
One of the nice things about using powerful HTML editors such as Microsoft FrontPage or Dreamweaver is all the formatting abilities these editors offer. You can easily create bulleted and numbered lists, change fonts and colors, add borders, spell-check your document, and so on. You can pretty much forget you are working on a help system and concentrate on the documents.

In addition to using regular in-line formatting, you can use cascading style sheets to define and use custom styles. Style sheets are great because they guarantee consistency and separate formatting from content. If you decide to change the way your document should look, just change the style sheet and rebuild the help system. You don't have to touch the content at all.

Getting back to our tour, let's create a style sheet with a couple of styles and use them to enhance the appearance of our content using Microsoft FrontPage (if you are using a different editor, the steps will be slightly different, but the main mechanisms will be the same).

1. Create a new text file and save it as **ScubaStyles.css** in the folder that contains all the tutorial content.
2. In the Doc-To-Help project editor, click the **Project** icon and select the **Documents** item from the left pane Tree View.
3. Double-click the **Scuba Dive.htm** document to open it in FrontPage or your default HTML editor.

- In FrontPage, select the **Format | Style Sheet Links** menu option. The **Link Style Sheet** dialog box opens.



- Click the **Add** button and use the **Look in** drop-down to locate the **ScubaStyles.css** file we created in step 1.
- Select the file and click **OK**.
- Repeat the last steps to add the new style sheet to the **Learn to Dive** and **Equipment** documents.

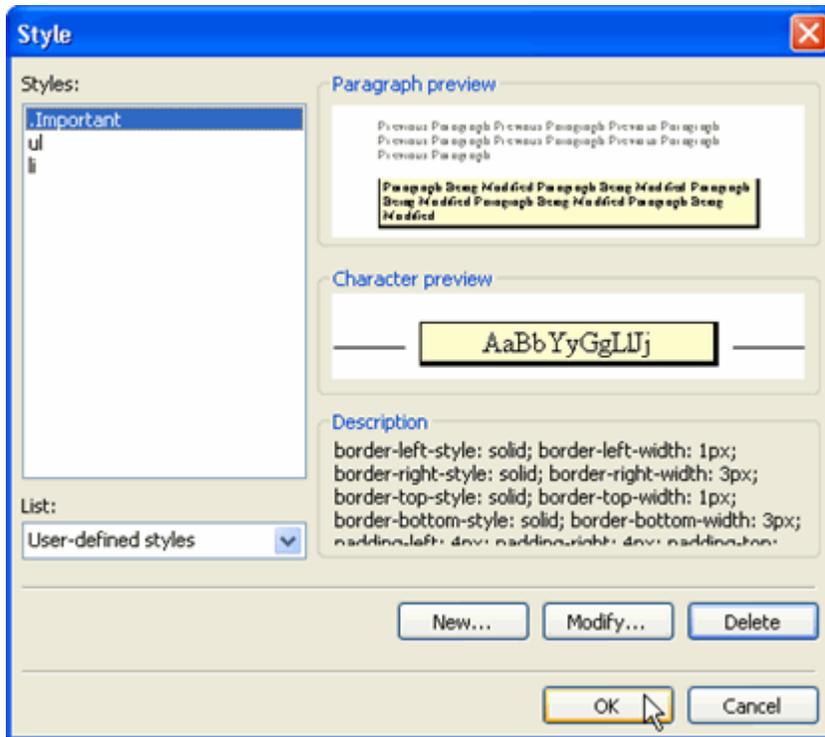
The new style sheet has been added to the documents. The HTML header in each file should now have a line that looks like this:

```
<link rel="stylesheet" type="text/css" href="ScubaStyles.css">
```

The documents look exactly the same, because the new style sheet is empty. You can now use FrontPage to add styles to the css file and to modify the appearance of the styles associated with HTML styles.

We will now create a new paragraph style called "Important" and modify the styles associated with the "UL" and "LI" Html tags to customize the appearance of bullet lists.

- In FrontPage, select the **Format | Style Sheet Links** menu option again.
- In the dialog box, select the **ScubaStyles.css** file and click **Edit**.
- In the **Style** toolbar, click the **Style** button . In the **Style** dialog, click **New** to create the new style.
- Enter the new style name **Important** and use the **Format** button to customize the border, background color, and font for the new style (use any options you want for now). Click **OK** to close the **New Style** dialog box when you are finished.



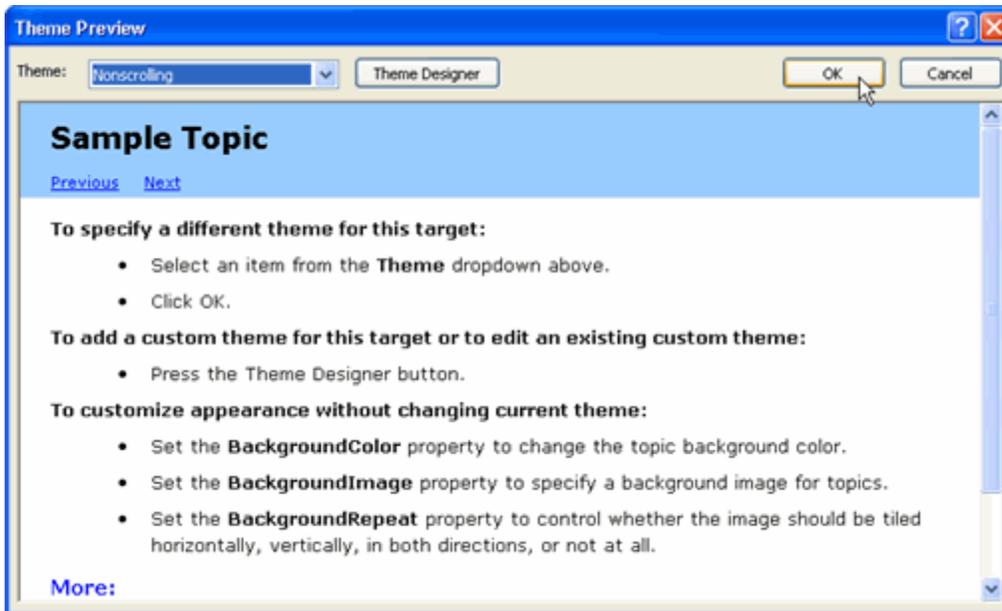
5. Back in the **Style** dialog box, select **HTML Tags** from the **List** drop-down.
6. Select the **UL** item (unordered list) from the list of styles and click **Modify** to customize the style. Repeat the process to customize the **LI** (list item) style.
7. Save and close the **ScubaStyle.css** file.

That's how you create and customize css styles. If you want to keep your document consistent with the tutorial, copy the **ScubaStyles.css** file from the tutorial folder into your working folder.

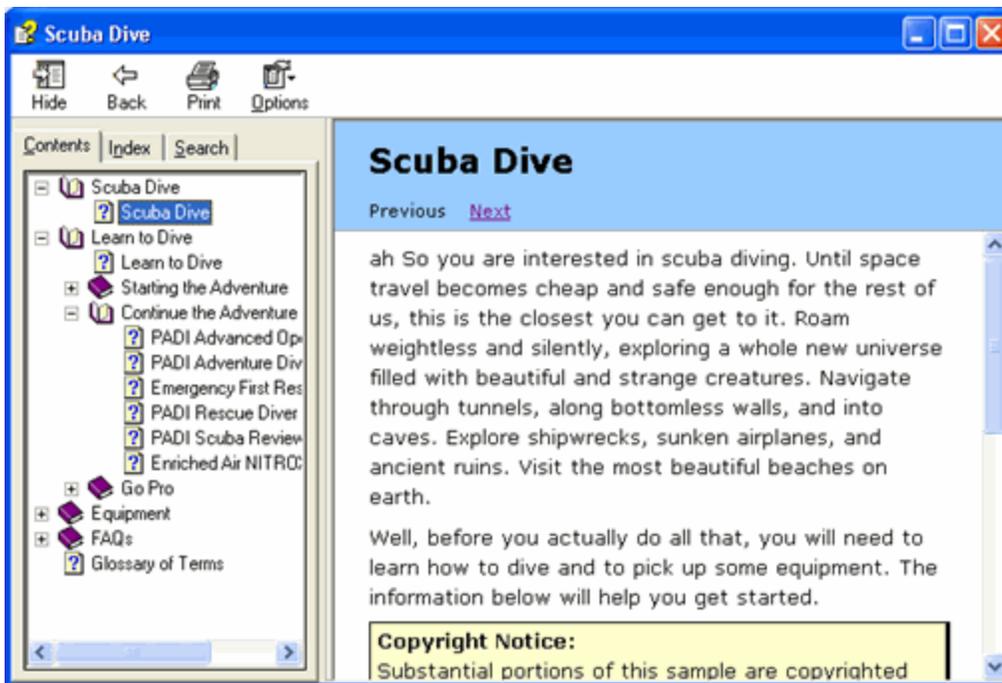
The styles associated with HTML tags are applied automatically. The custom style **Important** needs to be applied manually, by selecting a paragraph in FrontPage and picking the style from the style drop-down list. This was already done for you in the tutorial files, but feel free to highlight more items as **Important** if you want.

Before rebuilding the project to see the changes, let's also change the built-in theme.

1. In the Doc-To-Help project editor, click the **Project** icon and select the **Help Targets** item from the left pane Tree View.
2. Make sure the HTML Help target is selected in the right pane.
3. Click the **Theme** property in the Properties Pane. The **Theme Preview** window appears.
4. Select **Nonscrolling** from the **Theme** drop-down.



5. Click **OK** to close the **Theme Preview** window.
6. Build and view the target. The help file should now look like this:



Notice the blue non-scrolling title that was defined by the new theme, and the highlighted information formatted with the **Important** style.

Navigate the help system to see the appearance of the bullet lists, which now have a background and a border.

Making help content look good is easy with Doc-to-Help (and a good HTML editor).

Note: Doc-to-Help doesn't know about the custom .css file, so it won't detect changes and automatically offer to re-build the help file. You may have to use the **Rebuild Target** button to force an update and see the changes.

Adding Hyperlinks

Our help system already has a **Contents** pane that allows users to see the structure of the system and navigate it. It also has a **Search** pane that can locate topics that contains certain terms.

The next step in enhancing help navigation is the addition of hyperlinks, allowing users to click on certain parts of the document and automatically jump to other topics or show additional information.

There are several types of hyperlink mechanisms in help systems. All links are defined by special styles which are contained in Doc-to-Help style sheets and are automatically available in your source documents.

You can create links using [Doc-To-Help Markup Language \(D2HML\)](#) (page 233). Doc-To-Help provides the **D2HML Styles** toolbar and dialog boxes that integrate with FrontPage, Dreamweaver and Microsoft Word.

To create links, all you have to do is select the text, click a button in the **D2HML Styles** toolbar, and make a few choices in a dialog box.

After you have added a link, the changes are visible in the source HTML, so you can see exactly how it is done, edit the HTML markup, or create your own links without using the toolbar at all if you prefer.

The image below shows the **Doc-to-Help D2HML Styles** toolbar as it appears in FrontPage:

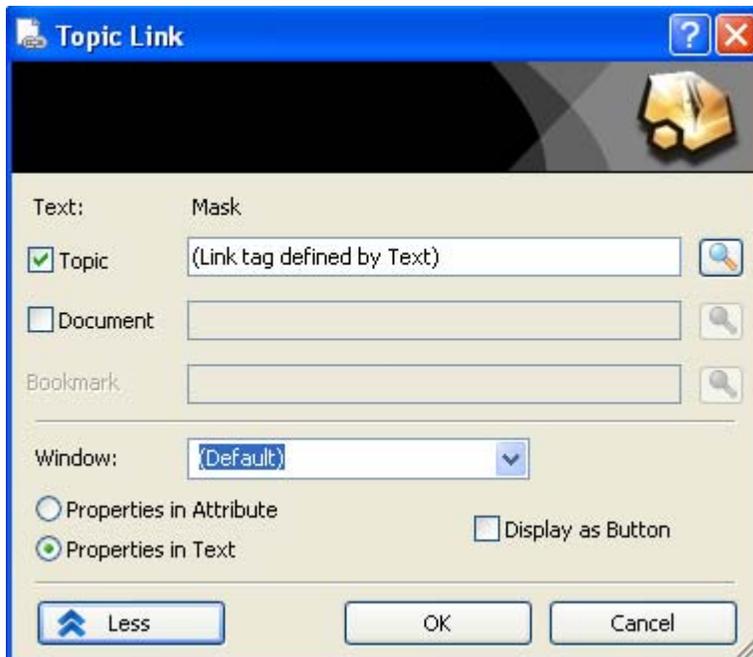


Topic Links

The first button in the toolbar creates the most common type of link, called a **Topic Link** or **Jump Link**. Some text is highlighted in the document, and when the user clicks it, a different topic appears.

Let's add a few jump links to our project.

1. In the Doc-To-Help project editor, double-click the **Equipment** document to open it in FrontPage.
2. Scroll down the document until you see the table that contains equipment prices.
3. Double-click the first item, *Mask*, to select it, and click the **Jump Link** button  in the toolbar. The **Topic Link** dialog box opens.



4. Click the **Browse** button  next to the **Topic** text box to see a list of available topics.
5. Select the topic **Masks** from the topic list and click **OK**.
6. Click **OK** to close the dialog box.
7. Save the document.
8. Repeat the process to add links to the other topics on the table (except *Booties & gloves*, which doesn't have a separate topic).

Before you rebuild the project, take a look at the HTML source code for the links. The **Masks** link we just added should look like this:

D2HML syntax: Adding Topic Links

```
<span class="C1HJump"><b>Mask</b>
<span style="display:none">|tag=Masks</span></span>
```

The button added a `` tag around the link text and a hidden `` that contains the name of the target topic. This is simple plain HTML. Doc-to-Help will turn this into a link when it builds the help project.

If you want to test the links, this is a good time to rebuild the project and try them out.

Note: After adding hyperlinks, it is good practice to rebuild the project and check the **Errors** and **Unresolved Links** lists in the Doc-to-Help **Build** tab.

Picking topics from a list reduces the chances of mistakes, but eventually you may change some topics names and that could break a few links.

Pop-up Links

The next button in the toolbar is used to create pop-up links.

Pop-up links are similar to topic links, except they show the target topic in a pop-up window. They are useful for reference material, things a user may want to take a quick look at in the context of the current topic.

For example, in our equipment page, a user may be looking at an individual item wondering how important that item is in the overall picture. He would like to take a quick look at the price table. This is a perfect application for a pop-up link.

1. If it not already open, double-click the **Equipment** document in the Doc-To-Help project editor to open it in FrontPage.
2. Scroll down the document to bottom of the **Masks** topic.
3. Add a paragraph with some descriptive text like **View retail prices**.
4. Highlight the new paragraph and click the **Popup Link** button  in the toolbar. The **Popup Link** dialog box opens.



5. Click the **Browse** button  next to the **Topic** text box to see a list of available topics.
6. Select the topic **So, What's This Going to Cost Me** on the topic list and click **OK**.

Check the HTML source code to see what the link looks like in HTML. It's almost the same as the topic link we added before, except for the style name.

D2HML syntax: Adding Popup Links

```
<span class="C1HPopup">View retail prices.
<span style="display:none">|tag=So, What's This Going to Cost Me?</span></span>
```

For consistency, copy this link to the bottom of the other equipment topics. Then rebuild the project and try out the pop-up links.

Here's what they look like:

The screenshot shows a web browser window with a blue title bar and a navigation toolbar. The main content area is titled 'Exposure Protection Suits' and contains several sections of text, a table, and a 'Water Temp.' section. A yellow callout box highlights the table and surrounding text.

Exposure Protection Suits
[Previous](#) [Next](#)

Form-fitting exposure suits are usually made of foam neoprene rubber (wetsuits) or spandex-like materials (skins), sometimes with fleece added.

What They Do
 Exposure suits tr...
 water can rob yo...
 The thickness an...
 thermal insulatio...
 Dry suits are rec...
 air between you...

What to Look F
 Fit and comfort...
 loose, however...
 heat loss.

Cost
 Wetsuits and ski...

Our Advice
 As long as a wet...
 limited to basic r...

Comfort Zone
 Recommended sk...

Water Temp.
 75 - 85F
 70 - 85F
 65 - 75F
 50 - 70F
 35 - 65F

So, What's This Going to Cost Me?
 No doubt about it: scuba is a gear-intensive activity. But scuba gear is also built to last. When properly cared for and regularly maintained, your first set of gear could conceivably be your last. Here's a breakdown based on suggested retail prices of gear reviewed in the pages of [Podate's Scuba Diving](#).

	Budget	Moderate	High-End
Mask	\$30	\$70	\$150
Fins	\$50	\$100	\$175
Snorkel	\$15	\$30	\$50
Bootsies & gloves	\$30	\$80	\$120
Wetsuit	\$120	\$300	\$550
BC	\$200	\$400	\$630
Regulator	\$200	\$400	\$1,600
Computer	\$250	\$500	\$1,300
Total	\$895	\$1,880	\$4,575

So we are looking at a bill between one and five thousand dollars. But remember, you can rent most of these items.

You can get started with the basic set (mask, fins, and wetsuit). This would cost between \$200 and \$1,000, depending on the quality of the gear.

[View retail prices.](#)

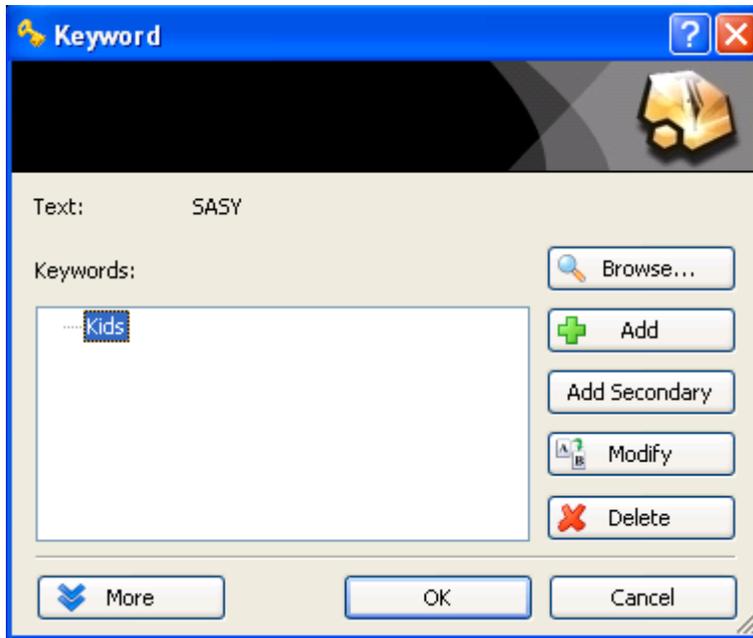
Keyword Links (KLinks)

The topic and pop-up links we discussed above are one-to-one links. They take the reader from the current topic directly to another.

Keyword links are one-to-many links. They may take the reader from the current topic to any topic that contains the **keyword** specified by the link. (When the user clicks the link, a dialog box appears listing all the available destinations.)

So before we add any keyword links to our sample project, we need some keywords to link to.

1. Open the **Learn to Dive** document in your HTML editor.
2. Find the *PADI Scuba Experiences for Kids* topic.
3. Select **SASY**, the first word in the topic text.
4. Click the **Keyword** button  in the toolbar. The **Keyword** dialog box opens.
5. Click the **Add** button in the dialog and type the keyword **Kids**.



6. Click **OK** to close the dialog box.

The keyword **Kids** is now associated with the **PADI Scuba Experiences for Kids** topic. Look at the HTML source and notice how that was done. As before, it is just a `` tag with a special Doc-to-Help style (`C1HIndex`).

D2HML syntax: Adding Keywords

```
<span class="C1HIndex">SASY</span>
<span class="C1HIndex">SASY
<span style="display:none">|keyword=Kids</span></span>
```

Now that we have a keyword defined, let's create a link to it.

1. Scroll back to the top of the document and locate the **Starting the Adventure** topic.
2. Select the word *Kids* in the second paragraph (**Kids are welcome too**).
3. Click the **Keyword Link** button  in the toolbar. The **Keyword Link** dialog box opens.
4. Click **OK** to accept the default and use the selected text as a target keyword.
5. Close the file.

D2HML syntax: Adding Keyword Links

```
<span class="C1HKeywordLink">Kids</span>
<span class="C1HKeywordLink">Kids
<span style="display:none">|keyword=Kids</span></span>
```

So far we have added a new keyword to the document (it will appear in the Index tab in the help system) and we have also added a link to it.

This is not very interesting because the keyword is associated with a single topic. To create a one-to-many link as we described above, let's assign the *Kids* keyword to a different topic.

1. Open the **FAQ Age Requirements** document in your HTML editor.
2. Select **12 years old** in the topic text.

3. Click the **Keyword** button  in the toolbar. The **Keyword** dialog box appears.
4. Click the **Add** button in the dialog and type the keyword **Kids** again.
5. Click **OK** to close the dialog box.
6. Save and close the file.

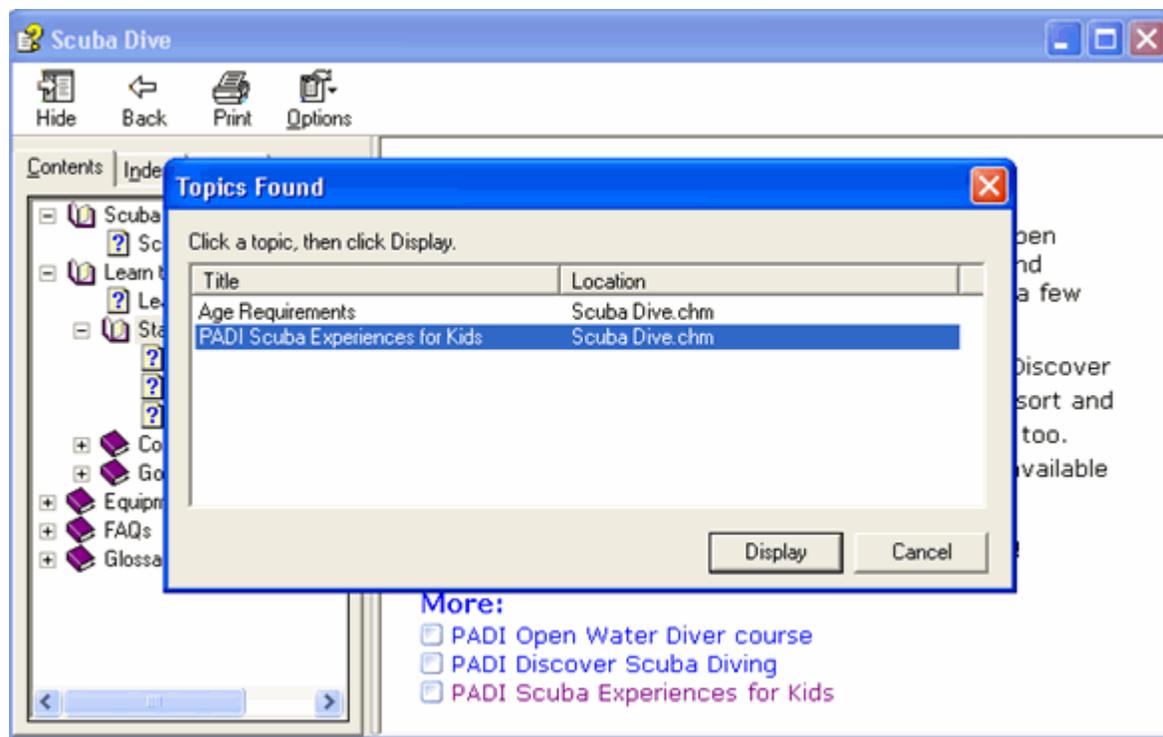
D2HML syntax: Adding Keywords

```
<span class="C1HIndex">12 years old</span>
<span class="C1HIndex">12 years old
<span style="display:none">|keyword=Kids</span></span>
```

Now rebuild the help file and open it.

Select the **Starting the Adventure** topic and click the keyword link **Kids** we just added.

A dialog box will pop up showing the two topics that were assigned the keyword *Kids*, as shown below.



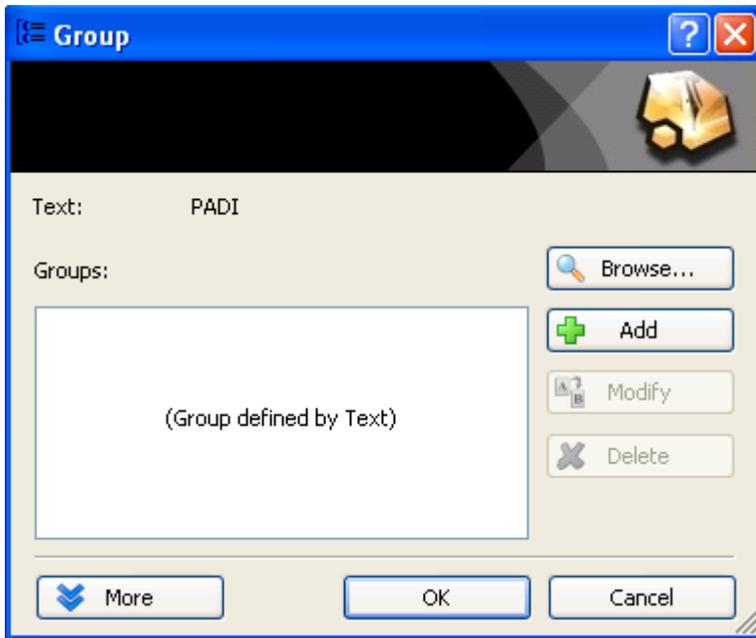
Keyword links are closely related to the help system's index, an important feature we will discuss later. For now, it is enough to remember that a keyword may be associated with one or more topics, and that a keyword link will take the reader to any of those topics.

Group Links (ALinks)

Group links are also known as ALinks (for associative links). They are similar to keyword links in the way that they are created and used, except for one important difference: keywords appear in the index, and groups don't.

To see the difference, let's create a new topic group called PADI and add a group link to show how it works.

1. Open the **Learn to Dive** document in your HTML editor.
2. Select the word **PADI** in the third topic title, **PADI Open Water Diver course**.
3. Click the **Group** button  in the toolbar. The **Group** dialog box opens.



4. Click **OK** to accept the selected text as the new group name.
5. Repeat this procedure for all other topics in the file that start with the word "PADI" (or do only three or four; there are ten).

D2HML syntax: Adding Groups

```
<span class="C1HGroup">PADI</span>
<span class="C1HGroup">PADI
<span style="display:none">|group=PADI</span></span>
```

Now that the new group has been created and assigned to a few topics, let's create the group link.

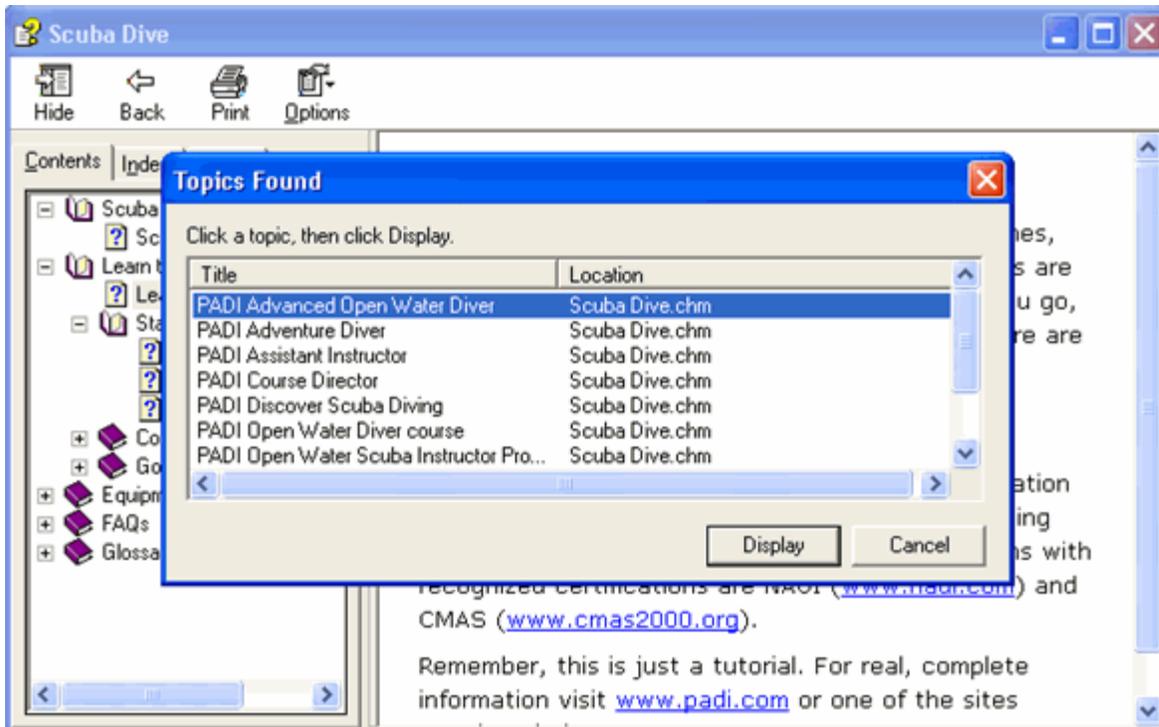
1. Scroll to the top of the document and select the word **PADI** in the second paragraph ("certifications offered by PADI").
2. Click the **Group Link** button  in the toolbar. The **Group Link** dialog box opens.
3. Click **OK** to accept the selected text as the name of the target group.

D2HML syntax: Adding Group Links

```
<span class="C1HGroupLink">PADI</span>
<span class="C1HGroupLink">PADI
<span style="display:none">|group=PADI</span></span>
```

To see the new group link in action, rebuild the document and navigate to the second topic, **Learn to Dive**. Click the **PADI** link and you will see a list with the ten topics that belong to the group.

Notice in the image below that the group name "PADI" doesn't appear in the index.



Group links are especially useful in situations where you are merging a collection of help files. For more details about groups, please see the Doc-to-Help reference.

Adding Image Maps

Image maps allow you to define areas within images and use each area as a hyperlink. They are more effective than menus when the options can be presented logically in an image.

For example, in our sample help project, an image map would be ideal to let the user explore the different types of diving equipment by clicking the image of a diver.

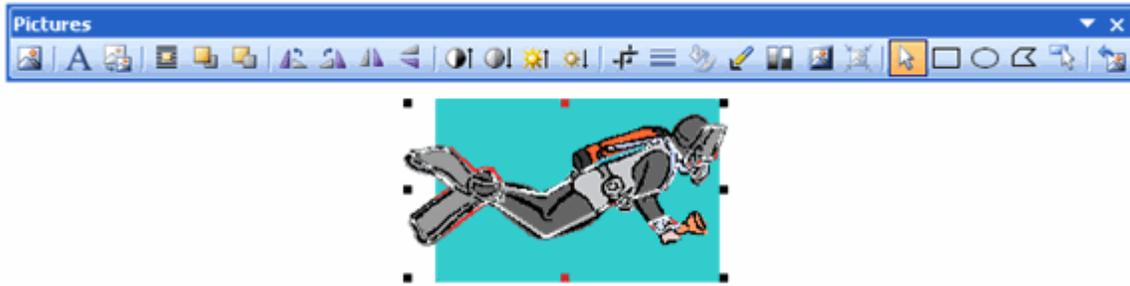
If you are creating your help content with Microsoft Word, you can create image maps using the Image Map Editor provided with Doc-to-Help. For more information, see [Using the Image Map Editor in Microsoft Word](#) (page 560).

If you are using HTML, you can draw a hot spot and then use D2HML to create the link.

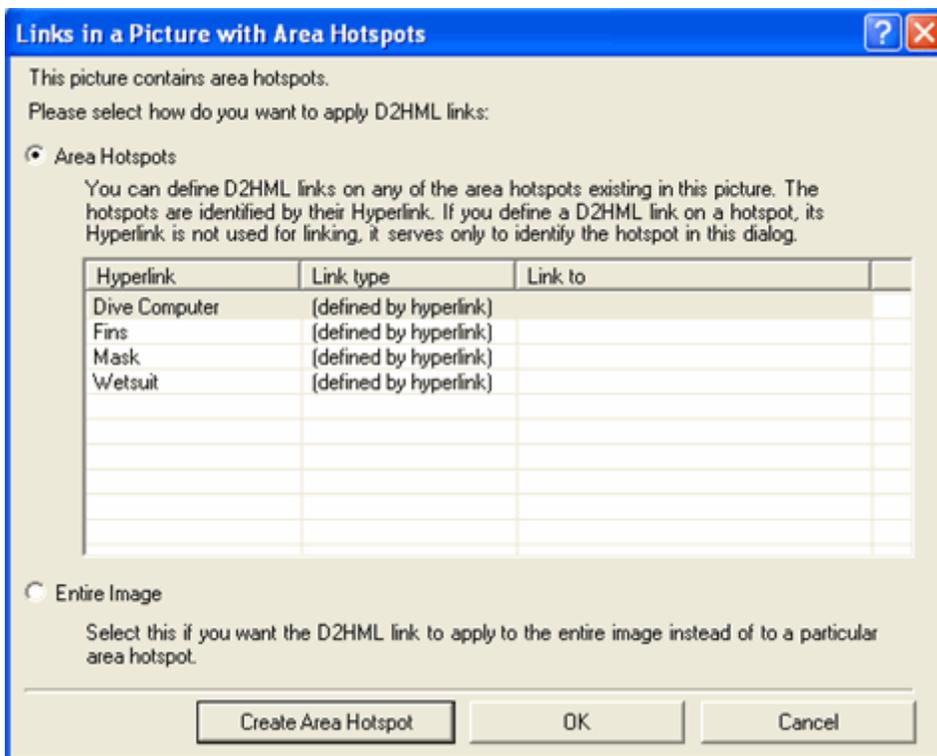
Follow the steps below to add an image map to our sample project (we are using Microsoft FrontPage, the steps will be slightly different with other editors).

1. Open the **Equipment** document.
2. Move the cursor to the end of the first topic and add a new empty paragraph.
3. Select the **Insert | Picture | From File** menu and insert the **Equipment.png** file from the C:\Program Files\ComponentOne\DocToHelp\Tutorial\ScubaDive\HTMLDocuments folder.
4. Select **View | Toolbars | Pictures** to make sure the **Pictures** toolbar is visible in FrontPage.
5. Select the image, click the **Polygonal Hotspot** button and draw an area around the diver's fins (see picture below). When you close the polygon, the **Insert Hyperlink** dialog box appears.
6. Enter **Fins** in the **Address** text box and click the **ScreenTip** button to enter a tooltip. Then click **OK** to close the dialog boxes.
7. Repeat this process and add hotspots around the mask, wetsuit, and dive computer on the diver's wrist (see picture below). Enter the following, corresponding names for each in the **Address** text box:
 - Mask

- Wetsuit
- Dive Computer



8. Select the **Fins** hot spot and click the **Jump Link** button  on the **D2HML Styles** toolbar. The **Links in a Picture with Area Hotspots** dialog box appears.



9. Select **Fins** in the **Hyperlink** column and click **Create Area Hotspot**. The **Topic Link D2HML** dialog box appears.
10. Click the **Browse** button and select **Fins** in the **Select Topic** dialog box.
11. Click **OK** to close the **Select Topic** and **Topic Link** dialog boxes.
12. Create an area hotspot for all of the hot spots in the **Hyperlink** column and link to the following, corresponding topics:
- Masks
 - Exposure Protection Suits
 - Dive Computers
13. Click **OK** to close the **Links in a Picture with Area Hotspots** dialog box.

14. Close the editor and rebuild the help file to test the image map. Moving the mouse over the image will show tooltips, if you provided them, and clicking the hotspots will take you directly to the topic with tips about each type of equipment.

Adding Pop-Up and Drop-Down Text

Sometimes you may want to attach additional content to explain a term, but don't want to create a whole new topic for that short explanation. There are several ways to do that using Doc-to-Help.

Inline pop-ups

Inline pop-ups show additional content in a pop-up window. They look like pop-up links, except the content is not a separate topic.

To add inline pop-ups to your document, use the **Inline Popup** button  in the **D2HML Styles** toolbar:

Select the text to which you want to attach more information and click the **Inline Popup** button. Then type the text that should be displayed in the pop-up and click **OK** to create the pop-up.

To add an inline pop-up to our sample project, follow these steps:

1. Open the **Scuba Dive.htm** document.
2. Copy the **Copyright Notice** paragraph and paste a new copy in the document.
3. Cut the text after the **Copyright Notice** title to the clipboard.
4. Select the **Copyright Notice** title and click the **Inline Popup** button. The **Inline Popup Link** dialog box appears.
5. Paste the text into the dialog box and click **OK**.

Note: The dialog box only allows you to enter plain text. However, once the text has been added to the document, it's HTML as usual. You can edit the text and add custom formatting as usual.

D2HML syntax: Adding an Inline Popup

```
<span class="C1HInlinePopup"><b>Copyright Notice (pop-up):</b></span>
<span class="C1HPopupText">Substantial ...<br>
  <a href="http://www.padi.com/">www.padi.com</a><br>
  <a href="http://www.scubadiving.com/">www.scubadiving.com</a><br>
  <a href="http://www.newscuba.com">www.newscuba.com</a></span>
```

Inline drop-downs and expands

Inline drop-downs and expands are similar to inline pop-ups, except the additional text is displayed within the document instead of in a pop-up.

Drop-downs show the additional text below the selection, as a paragraph that can be expanded or collapsed. Expands show the additional text within the current paragraph.

Creating inline drop-downs and expands is similar to creating inline pop-ups, except you will use different buttons in the toolbar.



These examples have already been created for you in the **Scuba Dive.htm** document:

D2HML syntax: Adding an Inline Dropdown

```
<span class="C1HInlineDropdown"><b>Copyright Notice (drop-down):</b>
</span><span class="C1HDropdownText">Substantial ...<br>
  <a href="http://www.padi.com/">www.padi.com</a><br>
  <a href="http://www.scubadiving.com/">www.scubadiving.com</a><br>
  <a href="http://www.newscluba.com">www.newscluba.com</a></span>
```

D2HML syntax: Adding an Inline Expand

```
<span class="C1HInlineExpand"><b>Copyright Notice (pop-up):</b>
</span><span class="C1HExpandText">Substantial ...<br>
  <a href="http://www.padi.com/">www.padi.com</a><br>
  <a href="http://www.scubadiving.com/">www.scubadiving.com</a><br>
  <a href="http://www.newscluba.com">www.newscluba.com</a></span>
```

Glossary popups

Glossary popups can be created automatically by Doc-to-Help. They are based on a word list and apply to the first instance of a glossary term in each topic throughout the Help system. For more information, see [Creating a Glossary](#) (page 333).

Creating a Glossary

A glossary is a list of specialized words with their definitions, often placed at the end of a book.

Doc-to-Help enhances this concept by compiling this word list and, if you choose, automatically attaching a pop-up with the definition wherever the words appear in the help system. You can also create glossary pop-ups manually.

Creating a glossary in Doc-to-Help is extremely easy, and we have already added one to our sample project.

Open the **Glossary.htm** document to see how it's done. This glossary is simply a list containing the terms, all formatted with the **Glossary Heading** style, followed by their definition, formatted as **C1H Popup Topic Text**.

Our sample glossary contains about 20 terms, and can be seen as a regular help topic. If you navigate around the help system, you will notice that the first instance of each word in a topic that matches a glossary term has a pop-up that shows the definition.

That was done automatically by Doc-to-Help, all we had to do was create and format the word list.

Creating an Index

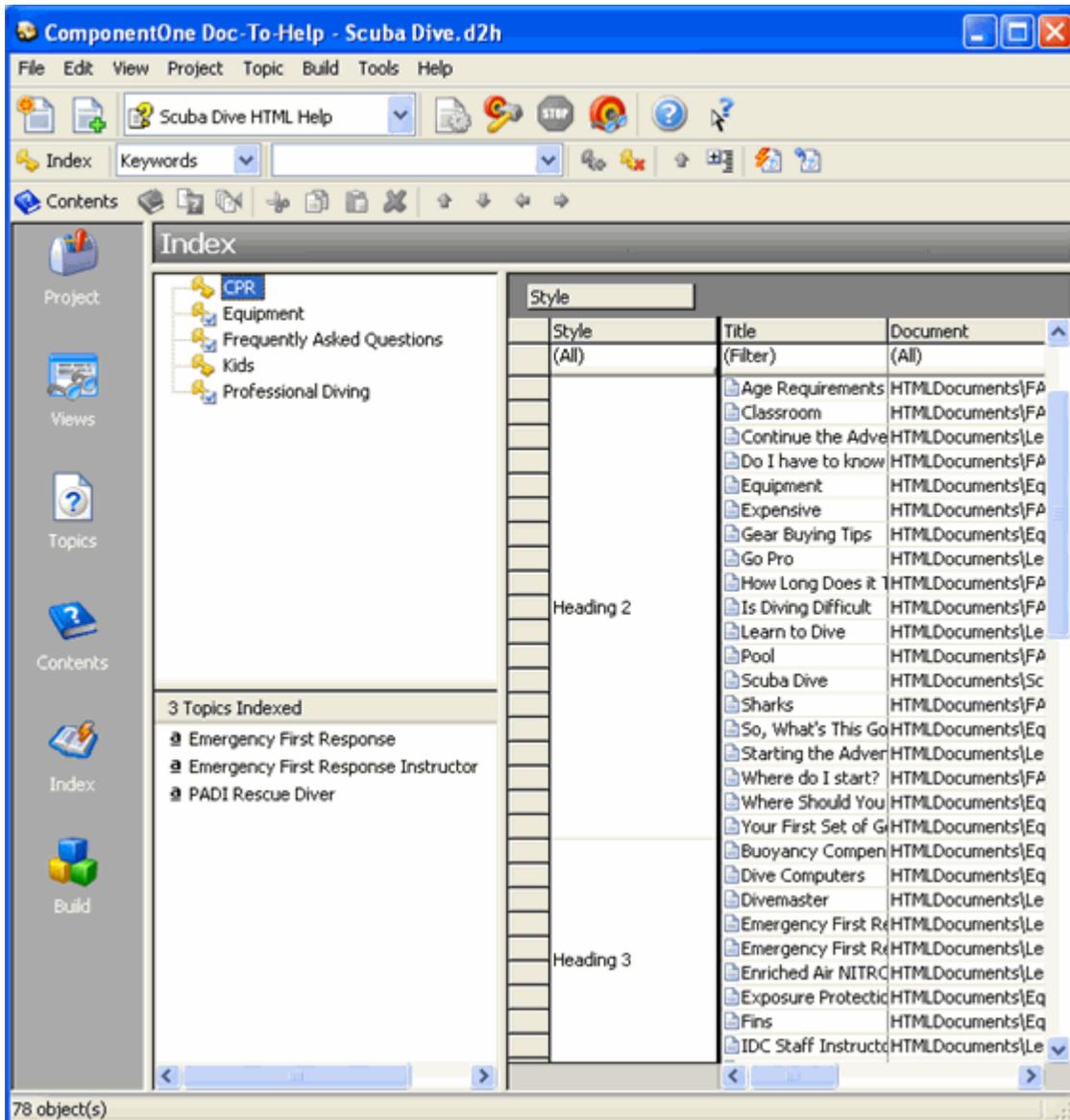
An index is a searchable list of keywords, each of which corresponds to one or more topics. When building an index, you should anticipate which words and phrases the reader is likely to look for and associate them with matching Help topics.

In an online Help system, selecting a keyword displays a list of the associated topics. In a printed manual, the index at the end of the book shows the page numbers that correspond to each keyword.

If you have been following all the steps in this QuickStart, you already know the easiest way to quickly add a keyword while you are editing help content. Simply select a word in your HTML editor and click the **Keyword** icon in the **Doc-to-Help** toolbar.

This method is great when you are working on individual topics, but it is not convenient when working with the overall help system. For that, Doc-to-Help has an **Index** view that shows all the keywords and their topics, and allows you to edit the keywords using drag and drop.

The image below shows Doc-to-Help's Expert mode Index view.



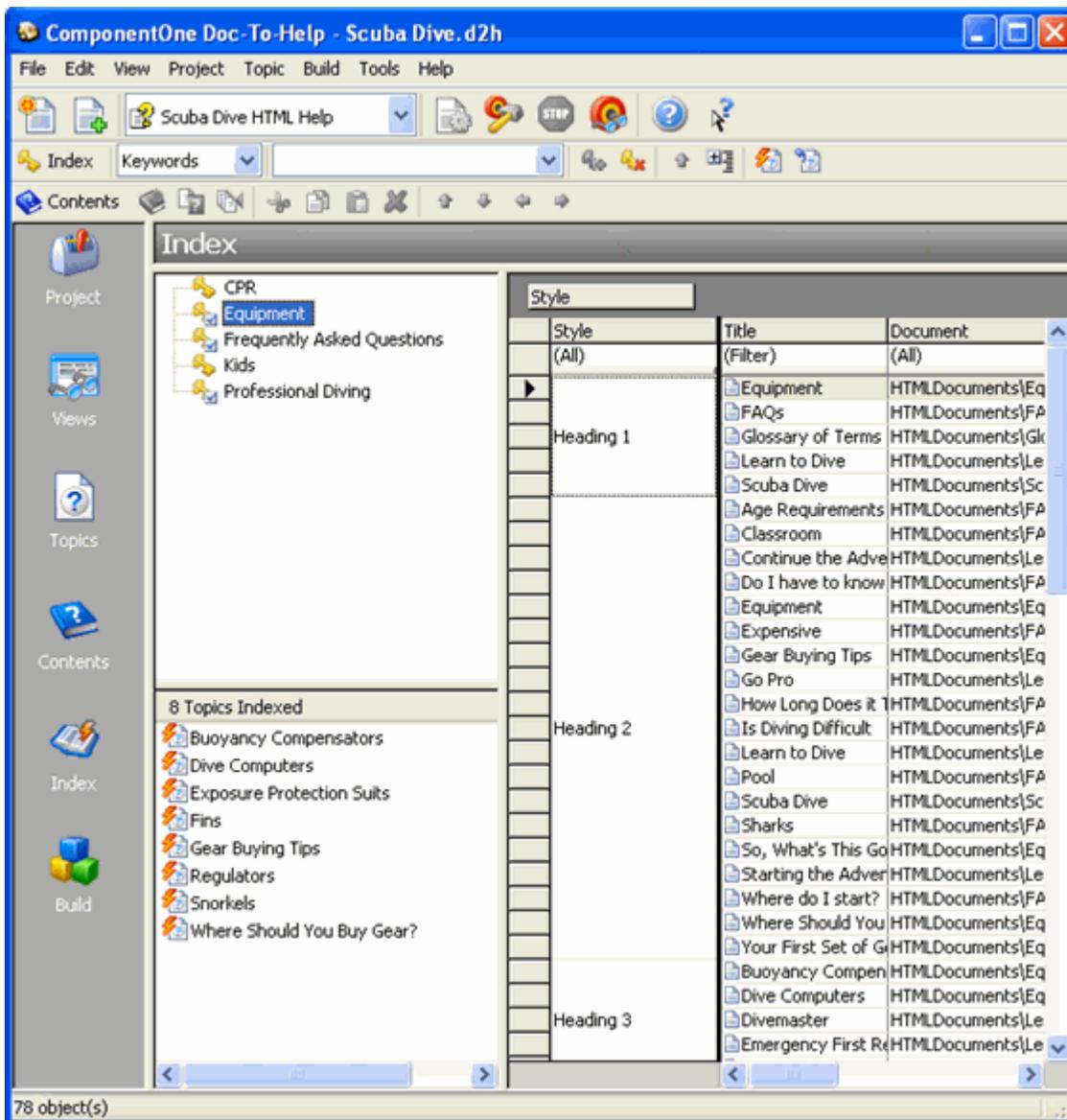
The index view shows a list with all keywords currently defined in the project. When you select a keyword on the list, all topics currently assigned to that keyword appear on the list below. You can add topics to the list by dragging them from the list on the right.

You can also use the context menus to add, rename, and remove keywords and topics assigned to them.

Before you start working on the index, drag the **Style** column to the top of the topic list. This will group topics by heading and will make it easier for you to find them.

Now let's add some keywords to our help system.

1. Right-click the keyword list and select **New**, then type the new keyword **Equipment** and press **Enter** to create the new keyword.
2. Select each of the topics directly related to equipment on the topic list and drag them to the list below the keywords. There are at least eight topics, from "Buoyancy Compensators" to "Where Should You Buy Your Gear?".
3. Repeat the steps above to create the keywords **Professional Diving** and **Frequently Asked Questions**.
4. Rebuild the project and select the **Index** tab to see the new index.



Doc-to-Help supports advanced indexing features such as multi-level keywords and automated index generation using scripts. These topics are useful in large help systems. See [Building an Index](#) (page 341) for additional information.

Using the Doc-To-Help Project

Open the **StyleGuide.d2h** project file that you created in the [Using Microsoft Word](#) (page 61) section of this tour. It should be located in the C:\Program Files\ComponentOne\DocToHelp\Tutorial folder. This project will be used to demonstrate working with topics, indexing and creating printed manuals.

Navigating Large Projects

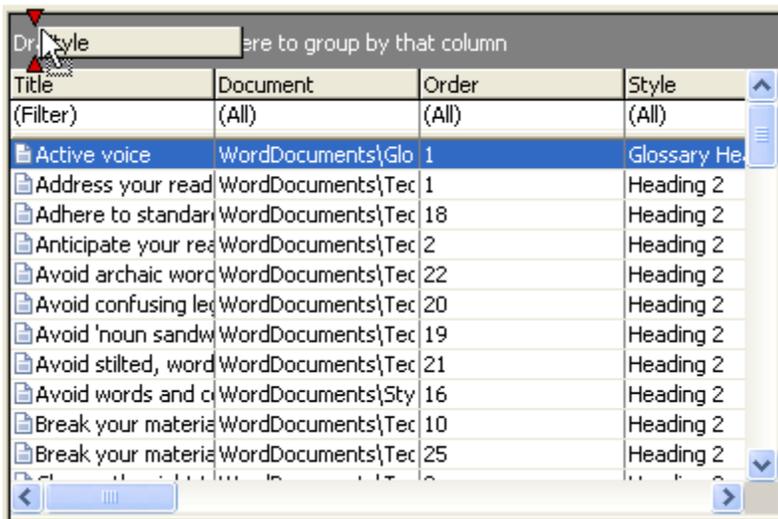
Large projects provide a special challenge when attempting to find a specific topic. Doc-To-Help's Topic view lets you find, sort, and open topics.

Sorting Topics For Easy Viewing

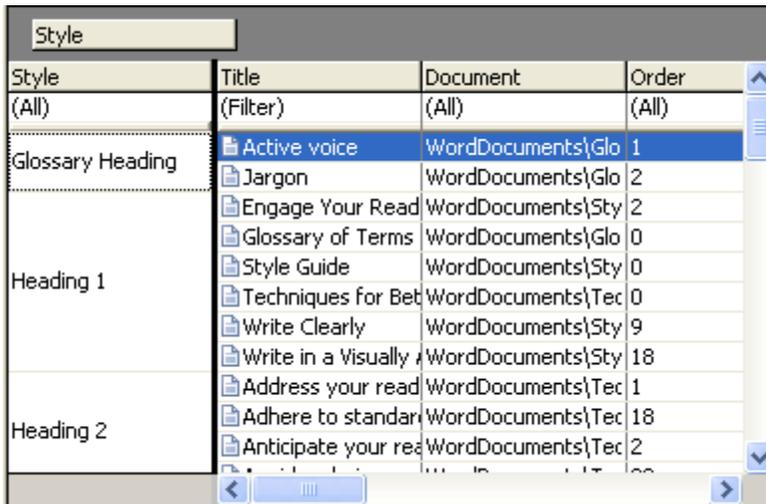
With Doc-To-Help, the Help author can modify the topic view by using the enhanced topic display and sorting capabilities in the convenient outlook style grouping feature. By simply dragging any of the topic headings (Title,

Document or Style) into the grouping area located at the top of the topic list pane, Doc-To-Help sorts automatically by that heading.

1. In the Icon Bar, click the **Topics** icon.
2. To sort by style, simply select the **Style** column and while holding the mouse button down, drag its header to the column grouping area as shown below.



3. Release the mouse button. Note the change in the Doc-To-Help topic list pane. The rows are now sorted by the Style headings.



4. Select the **Document** column and drag its header to the column group area. The topics are now sorted by Style then Document as shown below.

Style	Document	Title	Order
(All)	(All)	(Filter)	(All)
Glossary Heading	WordDocuments\Glo	Active voice	1
		Jargon	2
		Glossary of Terms	0
		Engage Your Read	2
Heading 1	WordDocuments\Sty	Style Guide	0
		Write Clearly	9
		Write in a Visually	18
	WordDocuments\Tec	Techniques for Bet	0
Heading 2	WordDocuments\Sty	Avoid words and c	16
		Identify your audie	3
		Include only one is	21

- To return to the default view, drag the headers back into the header bar.

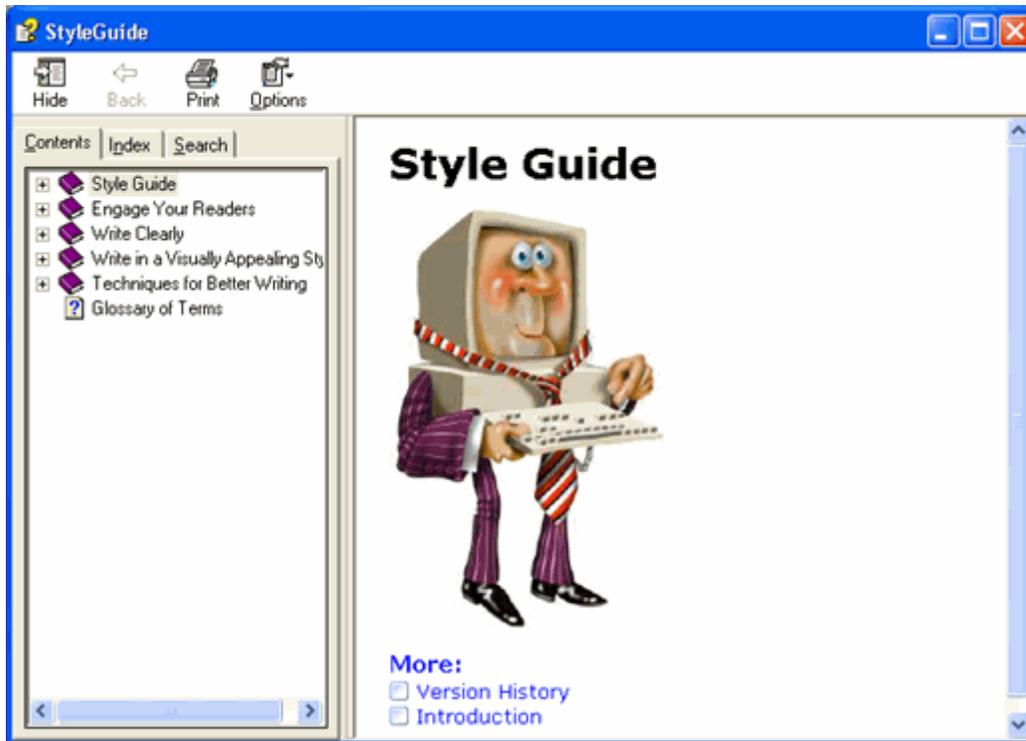
Filtering Topics

Doc-To-Help provides the author with topic filtering capabilities by using the filter boxes located beneath the headers in the topic list pane on the right. By simply choosing one or more of the drop down filtering options or by entering text in the topic textbox, you can filter the topics in any fashion you like.

- If necessary, select the **Topics** icon to switch to topics view.
- Click in the textbox under the Title header, type **Identify** and press **Enter**. The Topics list displays topics that contain the word **Identify** somewhere within the title.

Drag a column header here to group by that column				
Title	Document	Order	TopicType	Style
Identify	(All)	(All)	(All)	(All)
How do you identify pas	WordDocuments\Tec	8	Heading 2	Heading 2
Identify your audience	WordDocuments\Sty	3	Heading 2	Heading 2

- To filter by **Document**, click the arrow under the **Document** header and select **Techniques**. The Topics list now only displays the topics containing the word **Identify** in the Techniques document.



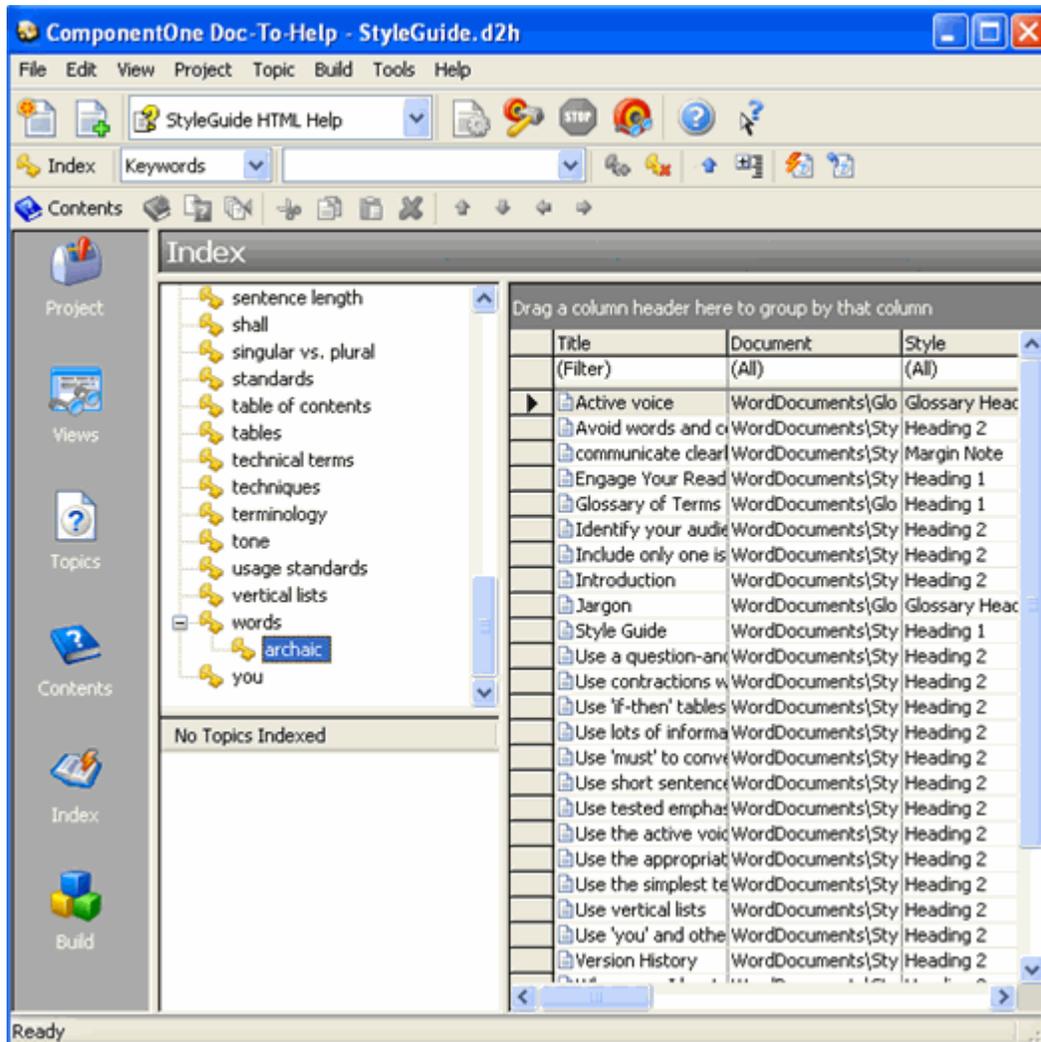
6. When you're finished, close the Help file.

Indexing in Doc-To-Help

The Doc-To-Help Project Editor provides an additional interface for associating index keywords (KLinks) or named groups (ALinks) with specific Help topics. By creating keywords and associating topics with them, **or** by choosing a topic and associating a keyword with that topic, you can develop an index for your Help file in the Project Editor. Of course, you can still use Word's {XE} fields to create index entries for print and keywords for Help targets if you prefer.

Examining the Index View

1. From the Icon Bar, select the **Index** icon.
 - The upper left pane displays the index keywords and secondary keywords.
 - The lower left pane displays the topics indexed to a selected keyword.
 - The right pane shows the list of topics generated by your styles.
2. In the left upper index keyword pane, scroll down to the keyword entry **words**.
3. To see the secondary keywords, expand the tree view by clicking the **+** next to **words**.
4. Select the secondary keyword, **archaic**. The lower left hand pane contains the topics associated with the secondary keyword.



- Place your cursor between the top of the Title column and the Document column and drag the **Document** column to the **right** until it is wide enough to read the Topic Titles.

Drag a column header here to group by that column

Title	Document
(Filter)	(All)
Active voice	WordDocuments\Glossary.doc
Address your readers and their interests	WordDocuments\Techniques.doc
Adhere to standard usage	WordDocuments\Techniques.doc
Anticipate your readers' questions	WordDocuments\Techniques.doc
Avoid archaic words and expressions	WordDocuments\Techniques.doc
Avoid confusing legal and technical jargon	WordDocuments\Techniques.doc
Avoid 'noun sandwiches'	WordDocuments\Techniques.doc
Avoid stilted, wordy language	WordDocuments\Techniques.doc
Avoid words and constructions that cause confusion	WordDocuments\StyleGuide.doc
Break your material into short sentences	WordDocuments\Techniques.doc

Adding Primary and Secondary Keywords

Now you will add the index keyword **expressions** and the secondary keyword **archaic** to the index keyword list.

- Select any primary keyword in the upper left pane.

2. Select **Project | New | Keyword**.
3. Replace <new> with **expressions**.



4. Press **Enter** to insert "expressions" into the list of keywords alphabetically.
5. While the keyword "expressions" is still highlighted in the upper left pane, click the **Show Secondary** button



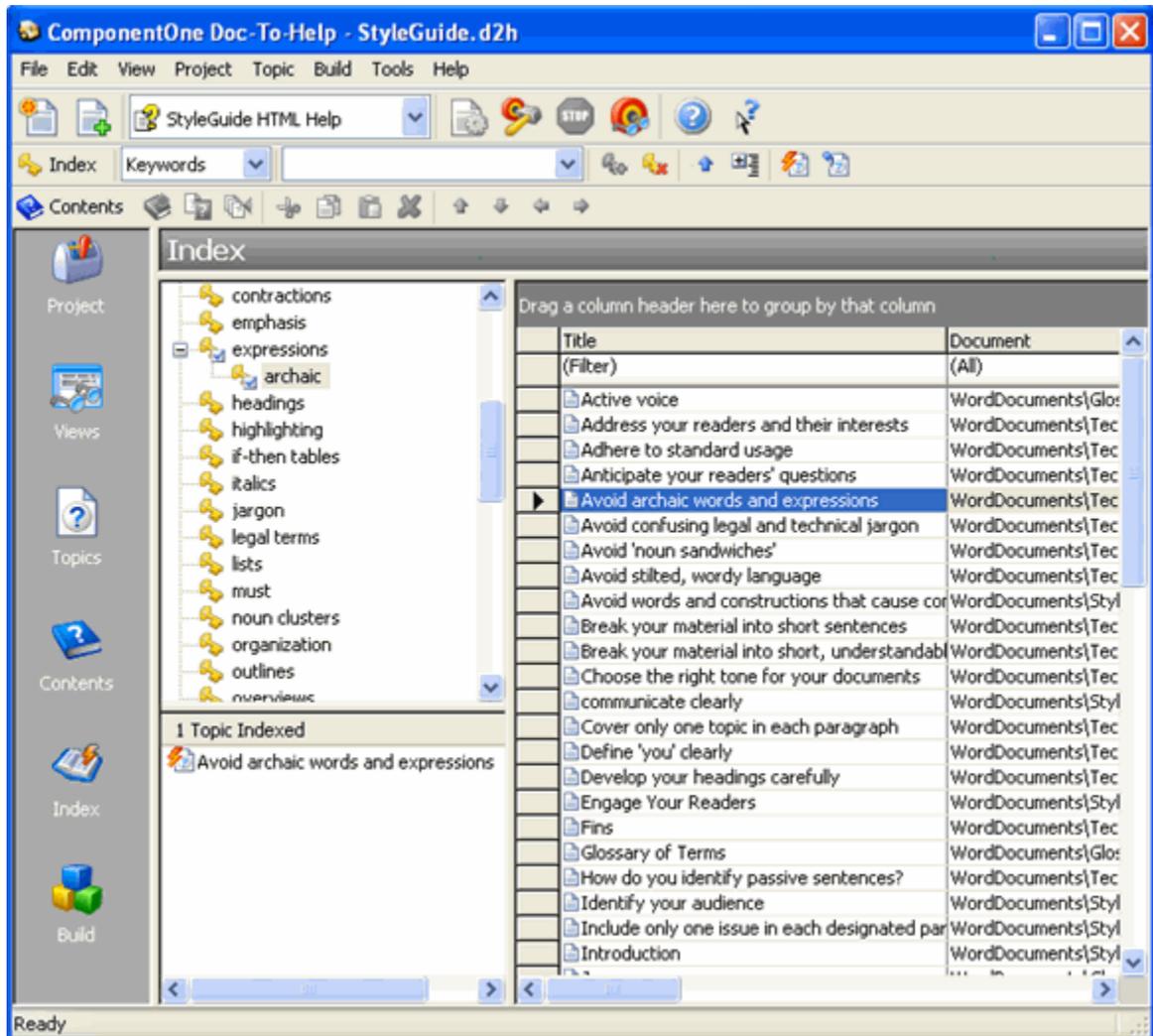
Note: Clicking the **Show Secondary** button changes the options on the **Index** toolbar, disabling some options and enabling others. It also activates the **Index Elements** text box.

6. Click anywhere in the **Index Elements** drop-down.
7. Type **archaic** and press **Enter**. Doc-To-Help adds the secondary keyword to the keyword list.



Associating Topics with a Keyword

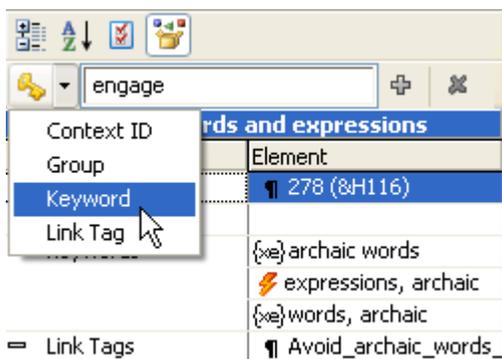
1. Select the "expressions" keyword.
2. In the right pane, select the topic title **Avoid archaic words and expressions**.
3. Right-click the topic and select **Index** from the shortcut menu.
4. Select the secondary keyword **archaic**.
5. In the right pane, select the topic title **Avoid archaic words and expressions**.
6. Right-click the topic and select **Index** from the shortcut menu. The topic has been associated with the both the primary and secondary keywords.



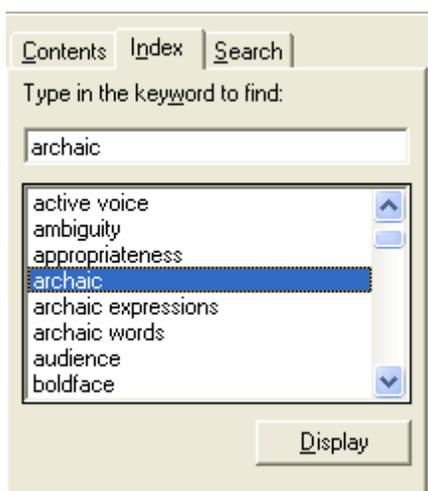
Associating Keywords with Topics

In the previous steps, you learned how to create an index keyword and associate it with a topic. Another method of indexing is to select a topic and associate keywords with that topic. This "topic to keyword" workflow makes it easier to create index keywords based on the existing topic titles.

1. Click the **Topics** icon.
2. Make the **Title** column as wide as necessary to view the list.
3. In the right pane, highlight **Avoid archaic words and expressions**.
4. In the property pane, click the **Collections** button .
5. Choose **Keyword** from the collections drop-down.



6. Type **archaic** into the keyword textbox and press **Enter**. Doc-To-Help adds archaic to the list of keywords for this topic.
7. Type **archaic expressions** into the keyword textbox and press **Enter**.
8. Select StyleGuide HTML Help from the Help Target drop-down on the **Project** toolbar.
9. Click the **Make Target** button.
10. After the target is compiled, click the **View Target** button.
11. Choose the **Index** tab to view your new index entries. The secondary keyword archaic and the primary keywords expressions and archaic expressions have been added to the index and are associated with the correct topic.



12. When you're finished, close the Help file.

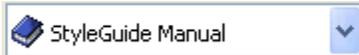
For more information, see [Building an Index](#) (page 341).

Creating Printed Manuals

One of the features included in Doc-To-Help is the ability to produce a "Printed Manual" in the same manner as you produce any other Help target. The compiled Printed Manual Help target includes all of the source document content, a table of contents, an index and customizable "front page". By setting the [ByLine](#), [SuperTitle](#) and [Title](#) properties you can customize the front page of your manual with any text you like. Additionally, Doc-To-Help supports **live links** in your manual Word document; **live links** are working links in the PDF generated from your manual document.

Examining the Printed Manual

1. Select **StyleGuide Manual** from the **Project** toolbar drop-down.



2. Click the **Make Target** button.
3. Click the **View Target** button. The front page of your printed manual contains the default [ByLine](#), [SuperTitle](#) and [Title](#).
4. Scroll down to see that Doc-To-Help has automatically included a table of contents.
5. Scroll down to the "Introduction" heading. The graphic and the margin note are included in the printed manual. Note that the bulleted items under the heading "How can we be better writers?" have page numbers in place of the hyperlinks in the online Help file and the "printed manual" conditional text is included.

How can we be better writers?

We believe that the most important goals are these--

- → "Engage Your Readers" on page 3.
- → "Write Clearly" on page 5.
- → "Write in a Visually Appealing Style" on page 7.

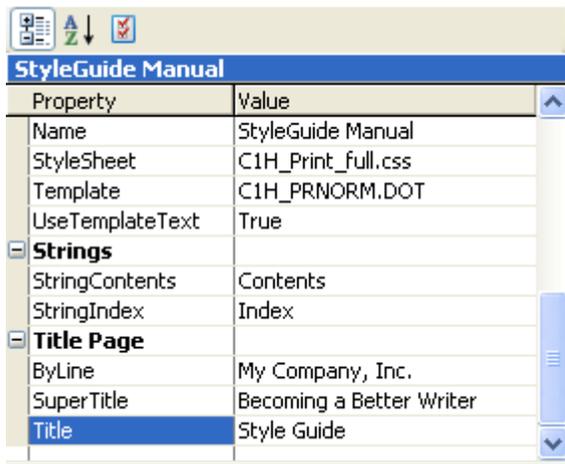
The next several pages summarize some of the best techniques to achieve these three goals. Following the summary, you'll find detailed suggestions for applying these techniques to your writing.

6. Scroll to the end of the printed manual. Doc-To-Help has automatically included an index at the end of the printed manual.
7. When you're finished, close the printed manual.

Modifying the Front Page of the Printed Manual

The next step is to change the defaults of the [ByLine](#), [SuperTitle](#) and [Title](#) properties to suit the StyleGuide document. By default the [ByLine](#) property is set to "By ComponentOne LLC", the [SuperTitle](#) property is set to "Doc-To-Help", and the [Title](#) property is set to "Standard Template".

1. Select the **Project** icon.
2. In the left pane, click **Help Targets**.
3. Select **StyleGuide Manual** from the right pane.
4. In the properties pane, select the [ByLine](#) Property.
5. Highlight the *By ComponentOne LLC* text and type *My Company, Inc.*
6. Select the [SuperTitle](#) property.
7. Highlight the Doc-To-Help text and type *Becoming a Better Writer*.
8. Select the [Title](#) property.
9. Highlight the **Standard Template** text and type *Style Guide*.



10. Click the **Make Target** button.
11. Click the **View** target button. Doc-To-Help has converted your title page to match the changes you made to the properties.
12. When you're finished, close the printed manual.

Creating Live Links

Doc-To-Help creates links in the Word document created when you build a Manual target. If you convert the resulting document to a PDF, these links become live, or working, links in the PDF. Simply set the [LiveLinks](#) property to **True** for the Manual target. All topic links, except pop-up links, regardless of the way they are specified in the source document, become Word hyperlinks in the manual document. When the document is converted to a PDF, the hyperlinks are live PDF links.

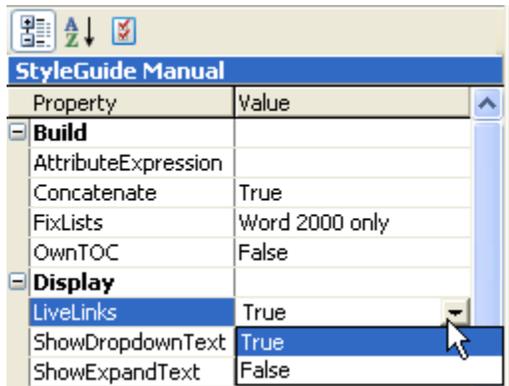
Note: Pop-up links do not become live links in the Manual target and PDF because neither Word nor Acrobat PDF has a pop-up feature.

The [Concatenate](#) property must be set to **True** in order for the **LiveLinks** property to take effect.

There is an issue with the Adobe Acrobat PDF converter when *Enable accessibility and reflow with Tagged PDF* is checked in Adobe Acrobat, including the latest version 7. When the PDF is produced and the Word hyperlinks are converted to PDF links, the links jump to the top of the page containing the destination topic instead of jumping to the desired topic location. To fix this problem, uncheck the *Enable accessibility and reflow with Tagged PDF* checkbox in Adobe Acrobat before creating a PDF or use a PDF converter other than Adobe Acrobat.

To Create Live Links:

1. Select the **Project** icon.
2. In the left pane, click **Help Targets**.
3. Select **StyleGuide Manual** from the right pane.
4. In the properties pane, click the drop-down arrow next to the [LiveLinks](#) property and select **True**.



- Click the **Make Target** button.
- Click the **View Target** button.
- Scroll down to the "Introduction" heading. Note that the bulleted items under the heading "How can we be better writers?" are now links in addition to having page reference numbers.

How can we be better writers?

We believe that the most important goals are these--

- ["Engage Your Readers"](#) on page 3.
- ["Write Clearly"](#) on page 5.
- ["Write in a Visually Appealing Style"](#) on page 6.

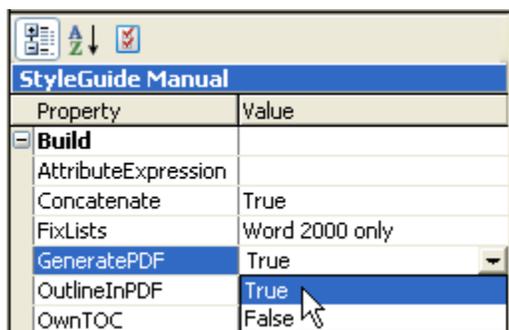
If the links are clicked, they jump to the appropriate topic. If the master document is converted to a PDF, the links jump to the appropriate topic in the PDF.

- When you are finished, close the printed manual.

Creating a PDF

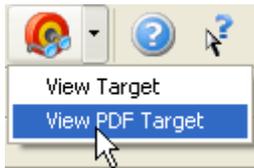
When you build a Printed Manual target, you have the option of creating a Word document or printing directly to a PDF. You can also choose to create an outline, or bookmarks which appear in the table of contents pane on the left side of the PDF viewer, which link to topics in the PDF.

- Select the **Project** icon.
- In the left pane, click **Help Targets**.
- Select **StyleGuide Manual** from the right pane.
- In the properties pane, click the drop-down arrow next to the [GeneratePDF](#) property and select **True**.



- Click the drop-down arrow next to the [OutlineInPDF](#) property and select **True**.

6. Click the **Make Target** button.
7. Click the drop-down arrow next to the **View Target** button and select **View PDF Target**. This option is also available on the **Build** menu when a manual target is selected in the **Help Target** drop-down list.



Note: Although Doc-To-Help now supports PDF generation, if you need more control over advanced PDF features, you may want to continue using a third-party PDF generator to convert the Word document generated for the Manual target to PDF.

Congratulations, you have completed the Doc-To-Help Guided Tour!

Working with Projects

A ComponentOne Doc-To-Help 2006 project file is a database that stores information about a Help project. Doc-To-Help maintains everything except the actual content of the source documents in a project file. Specifically, Doc-To-Help project files contain the following elements:

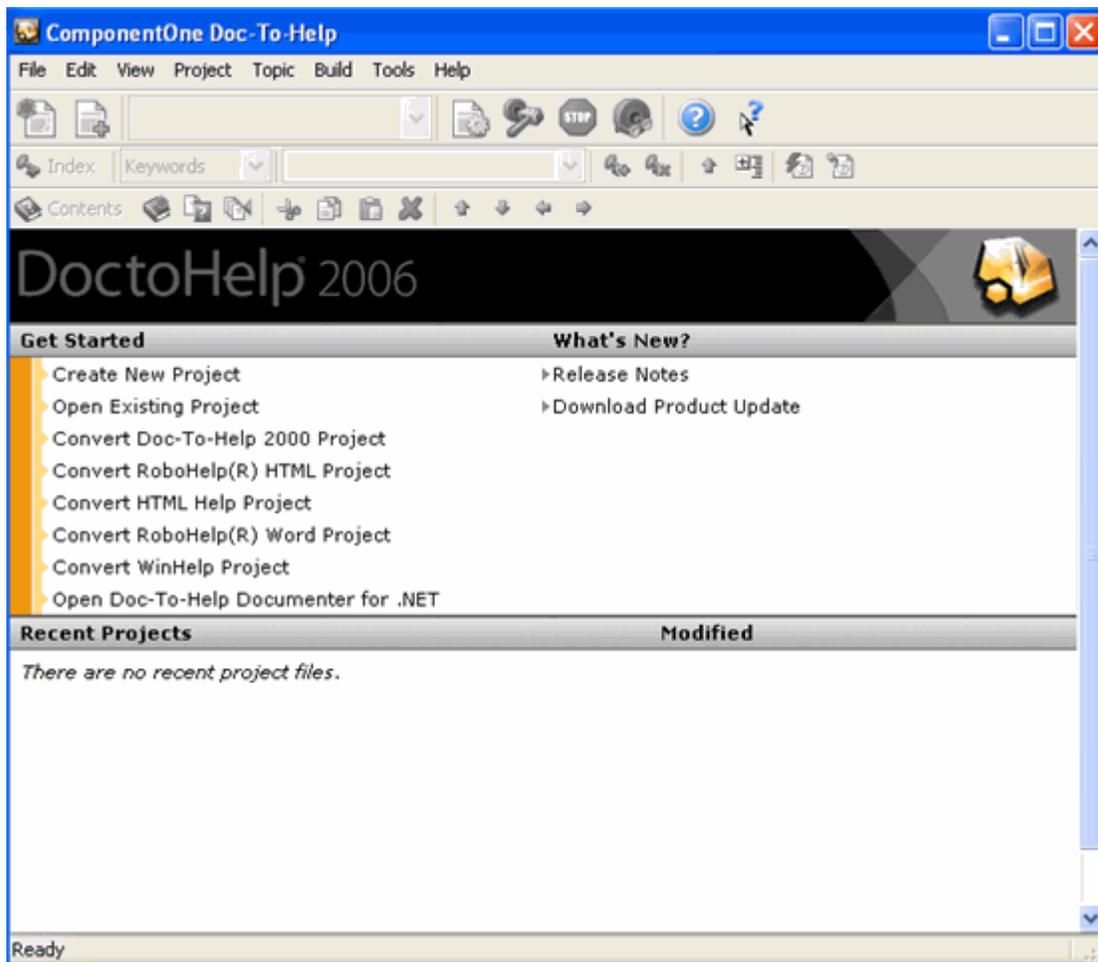
-  Source document paths
-  Document template paths
-  Topic titles derived during compilation
-  Paragraph style behaviors
-  Character style behaviors
-  Script code modules
-  Help window definitions
-  Index keywords visible to the reader
-  Group names hidden from the reader

The icons shown here are used throughout Doc-To-Help to identify elements of a specific type and to indicate relationships between disparate elements.

Since a Doc-To-Help project file is a database, it must exist on your hard disk or a network drive; it cannot reside in memory like a new document in Microsoft Word or FrontPage. By default, Doc-To-Help project files have a *.d2h* extension.

The Doc-To-Help Start Page

When you run Doc-To-Help from the Start menu, or by double-clicking its icon in Windows Desktop, you are presented with the Doc-To-Help Start Page.

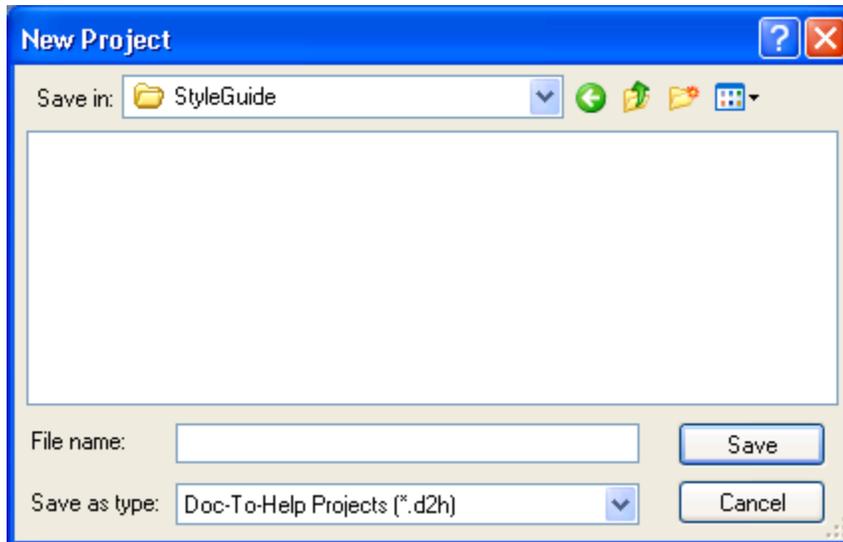


From here you can choose one of the following options to begin working with Doc-To-Help:

- [Create New Project](#)
- [Open Existing Project](#)
- [Convert Doc-To-Help 2000 Project](#)
- [Convert RoboHelp\(R\) HTML Project](#)
- [Convert HTML Help Project](#)
- [Convert a RoboHelp\(R\) Word Project](#)
- [Convert WinHelp Project](#)
- [Open Doc-To-Help Documenter for .NET](#)
- [Release Notes](#)
- [Download Product Updates](#)
- [Recent Projects](#)

Creating a New Project

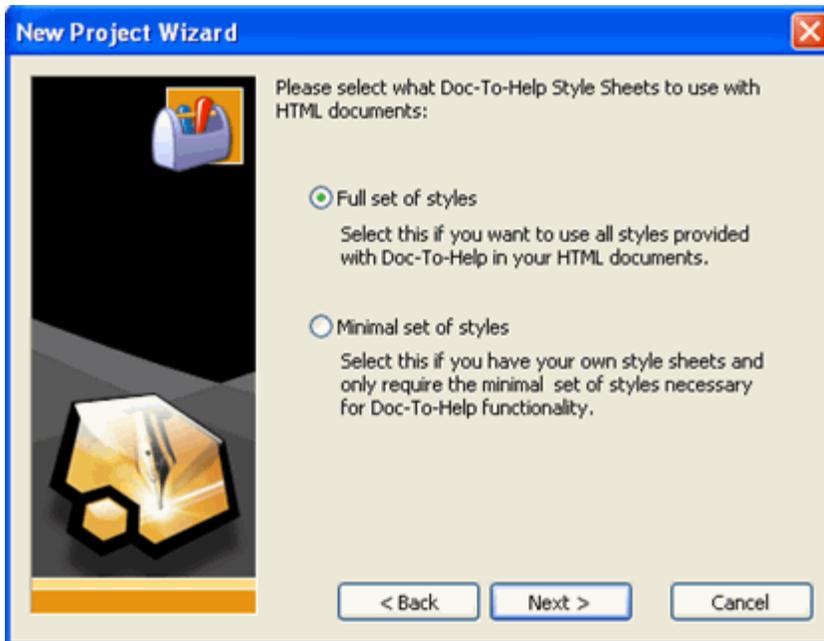
1. Click the **Create New Project** hyperlink on the Doc-To-Help Start page or select **File | New** within the Doc-To-Help project editor. The first screen of the **New Project Wizard** opens.
2. Using the **Browse** button to open the **New Project** dialog box, create or choose a directory and enter a project name in the **File name** textbox.



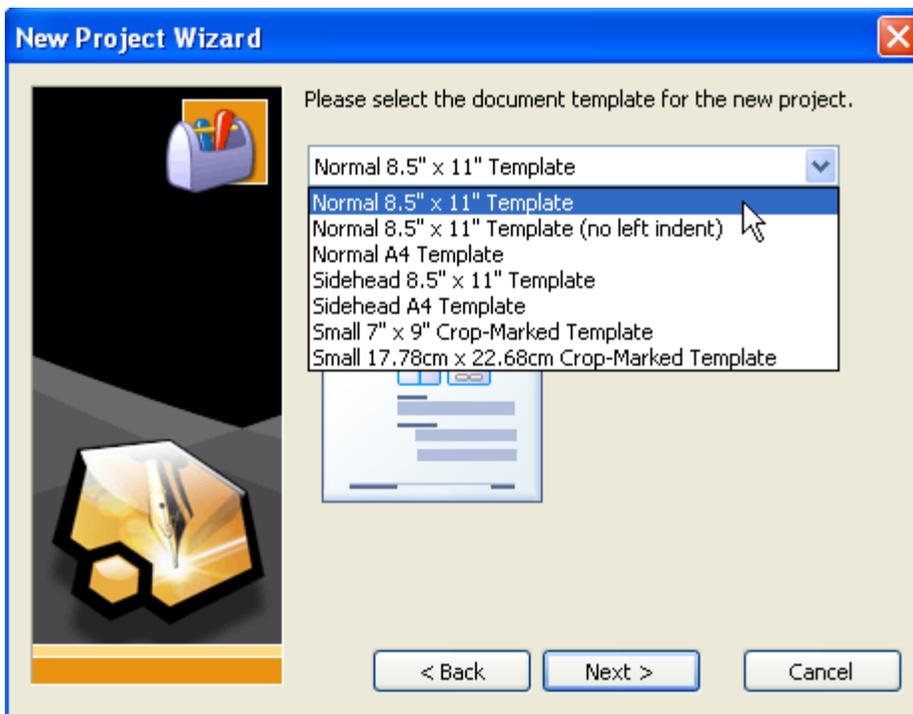
3. After you have created the new directory and entered the project name click **Save**, then **Next**.
4. Select the type of source documents you want to use in your project. You have three options:
 - Microsoft Word documents
 - Mixed documents
 - HTML documents

See [Doc-To-Help 2006 Source Documents](#) (page 181) for more information on the source documents that can be used with Doc-To-Help.

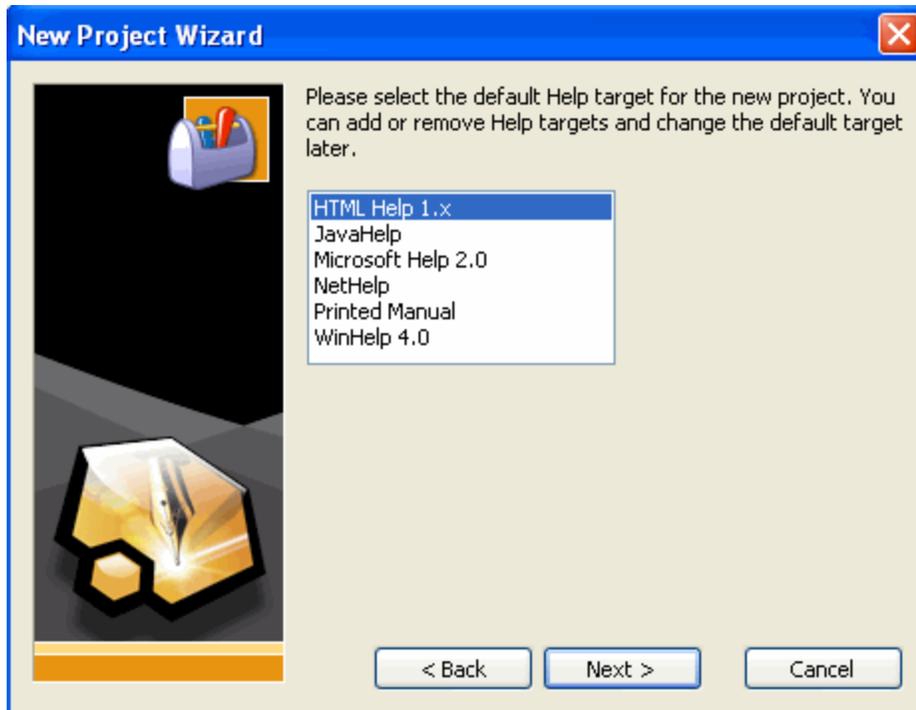
5. Click **Next**.
6. If you selected *Microsoft Word documents*, you can skip this step. If you selected *Mixed documents* or *HTML documents*, you are prompted to select the Doc-To-Help Style Sheets you want to use with the HTML documents:
 - Full set of styles should be used if you want to use some or all of the styles provided with Doc-To-Help in your HTML documents.
 - Minimal set of styles should be used if you want to use your own style sheets and the minimal set of styles necessary for Doc-To-Help functionality.



7. Choose one of the options and click **Next**.
8. From the drop-down list, choose the template that you wish to attach to your new project and click **Next**.

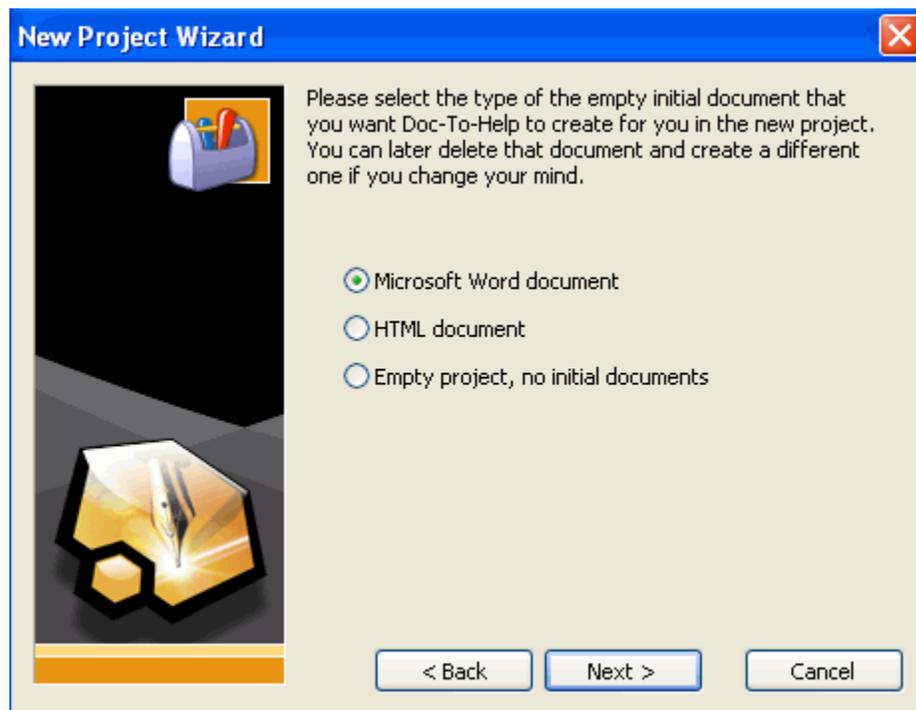


9. From the list of Help targets, choose your default Help target and click **Next**. Though you can build any Help target for your project, this determines the target your project compiles initially.



10. If **Mixed documents** was selected in step 5, the wizard asks the type of empty initial document you want Doc-To-Help to create. Select one of the following options and click **Next**:

- Microsoft Word document
- HTML document
- Empty project, no initial documents. If you choose not to create an initial document, you can always add documents to your project later.



Note: A glossary document of the same type as the empty initial document is also created and added to your project. If *Empty project* is selected, no glossary document is created.

11. Review your directory location and file name and click **Finish**.

The Doc-To-Help Wizard produces a new Help file and opens the project editor. If you specified Microsoft Word documents for your project or the type of initial document, a “start-up” document is opened in Microsoft Word and you can begin composing your documentation.

Note: The Word document is placed in the **WordDocuments** folder within your project folder. It is recommended that you keep all Word source documents for the project in this folder.

If you specified HTML documents for your project or the type of initial document, a **New Document** dialog box appears. Enter a name for your document and click **Save**. The **Adding New HTML Document** dialog appears:

- Select **Single topic** or **Multiple topics**.
- Enter a title for the first topic in the **Title** text box.
- If **Single topic** was selected, choose a Document Style from the drop-down box.
- Click **OK**.

An HTML page is opened in your default HTML editor and you can begin composing your documentation.

Note: The HTML document is placed in the **HTMLDocuments** folder within your project folder. It is recommended that you keep all HTML source documents for the project in this folder.

Opening an Existing Project

1. Click the **Open Existing Project** link in the Doc-To-Help Start Page or click **File | Open** within the Doc-To-Help project editor. The **Open** dialog box opens.
2. Specify a folder and the name of an existing project file, then click **Open**.

Converting Doc-To-Help 2000 Projects

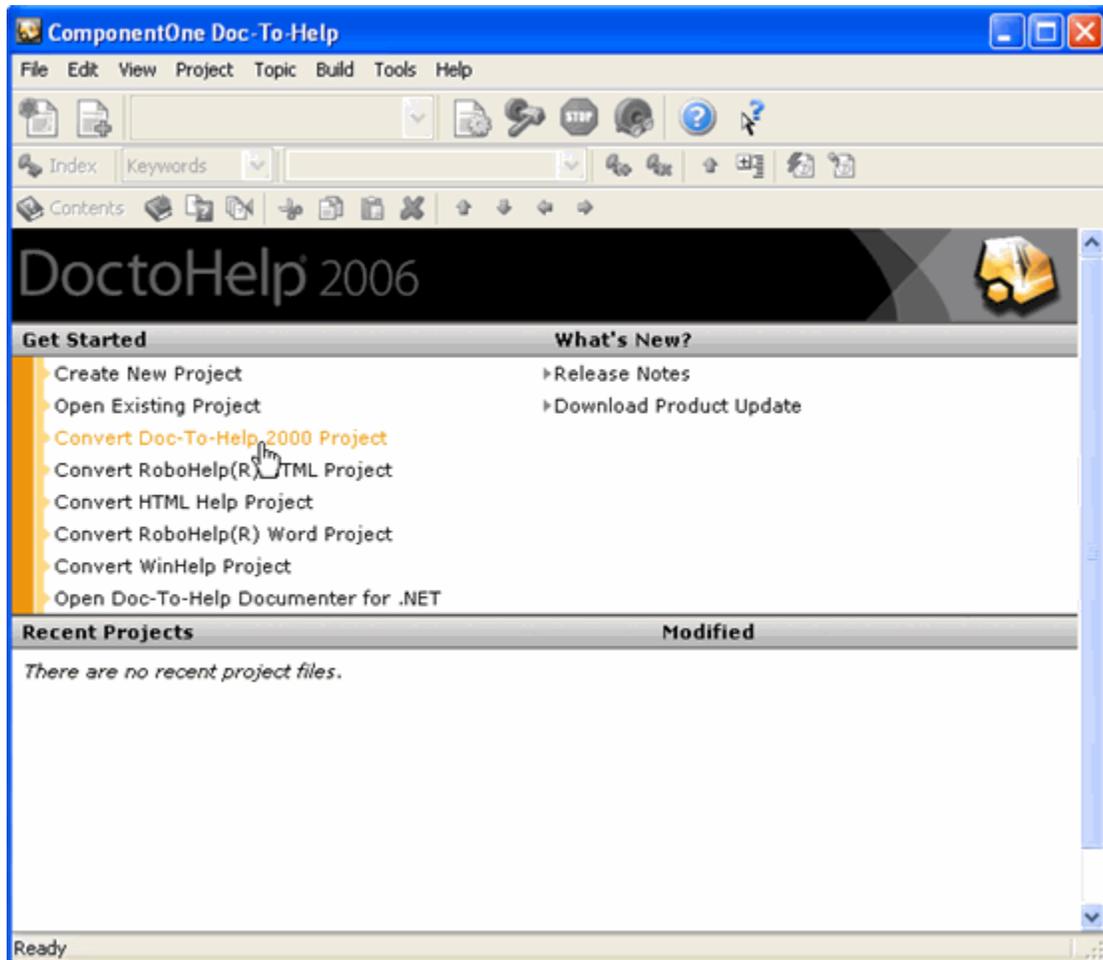
If you have projects that were produced in Doc-To-Help 2000, you can convert them to Doc-To-Help 2006 by using the Doc-To-Help 2000 conversion wizard.

Before Converting a Doc-To-Help 2000 Project

The project must be in Doc-To-Help 2000 format and the project must have been “built” (output directories must exist). You should also run the Doc-To-Help 2000 Diagnostics, particularly those options that deal with bookmarks and cross-references.

Converting a Doc-To-Help 2000 Project

1. Open Doc-To-Help 2006.
2. Click **Convert Doc-To-Help 2000 Project**.



The Doc-To-Help conversion dialog box opens.

3. Click **Next** to continue.
4. Using the browse button, locate the project (.INI file) you wish to convert and click **Open**.
5. In the **New Doc-To-Help project** textbox, type the name of the new file and click **Next**.
6. Review your selections and click **Finish**. The Doc-To-Help Converter status dialog box opens to display the conversion progress.
7. Once the conversion is complete you are prompted with “Project converted successfully.
8. Click **OK** to close this window, then press **F5** to build the current Help target”.

Note: A back-up of your existing Doc-To-Help 2000 project is saved in the D2HconverterBackup directory.

Accessing the Converted Project

By default, Doc-to-Help 2006 places your converted Help project files and folders in the parent directory where your Doc-To-Help 2000 files resided.

1. The source documents are placed directly in the parent directory.
2. The related folders are also placed in the parent directory. The Doc-To-Help conversion duplicates the original folder names used in your Doc-To-Help 2000 project.

Additional Doc-To-Help 2000 Conversion Items

The list below includes some known items that you should check after converting your Doc-To-Help 2000 project to Doc-To-Help 2006.

1. If there are topics that are excluded from the default Help target via conditional text, then any hyperlinks to those topics will need to be recreated manually.
2. Tables containing conditional text may require the conditional text be reapplied, particularly if multiple targets were involved. If the original table contained a conditional column, you must make two tables and mark them accordingly.
3. If you have a list of related topics at the end of a parent topic, you may have to reinsert the related topics heading, or the heading that precedes the list of topics and their buttons. Doc-To-Help uses the Help target [ButtonLabel](#) property to add a related topics heading.
4. If a {bmc} reference or HTML passthrough code cannot locate a file, check the value of the [CopyFolder](#) property. Either move the missing file to one of the referenced folders or append a new folder (preceded by a semi-colon) to the string value.

Doc-To-Help RoboHelp, HTML Help and WinHelp Converters



Doc-To-Help 2006 can convert a RoboHelp® Word or HTML project or any WinHelp (.hlp) or HTML Help project (.hhp) produced by another help-authoring tool to Doc-To-Help. There are four **Convert** options, in addition to the **Convert Doc-To-Help 2000 Project** option, when you open Doc-To-Help 2006: **Convert RoboHelp(R) Project**, **Convert HTML Help Project**, **Convert RoboHelp(R) Word Project** and **Convert WinHelp Project**.

The RoboHelp converter differs from the standard converters only in that it converts some RoboHelp-specific features that are not part of the generic HTML Help and WinHelp specifications, such as RoboHelp conditional build tags, RoboHelp pop-ups, and the glossary.

Both the RoboHelp HTML Project and HTML Help Project converters need an HTML Help project (.hhp) to begin; therefore, it is necessary to build an HTML Help layout, or target, for your RoboHelp project before converting it to Doc-To-Help.

The converters walk you through a conversion wizard, making the process as simple as a few clicks.

RoboHelp HTML Project Conversion

This section describes the particulars of the RoboHelp HTML to Doc-To-Help conversion.

Files:

All files in the RoboHelp project folder and its subfolders are copied to the Doc-To-Help project folder except those that are used by RoboHelp for its internal purposes. This is done to ensure that external files that may be referenced in the project source files are present in the converted folder and do not cause broken links. If you see files that you know are not needed, you can delete them manually.

Topics:

Converted topic files form the Doc-To-Help document tree. The tree hierarchy is based on the table of contents (TOC) defined in the RoboHelp project (not on the topic hierarchy that may be different from TOC). For each TOC book that does not have a corresponding topic, an empty topic file is created, with name prefix “_D2H_”.

Topic templates, headers and footers:

Doc-To-Help does not have a concept of topic templates. However, no information is lost in conversion. Template headers and footers become a permanent part of topic content. So you will see headers and footers in converted topics

exactly as they were in the source document, the only difference is that changes you make to headers and footers in the Doc-To-Help project apply only to the topic to which it belongs, not to other topics that were based on the same topic template in the RoboHelp project.

Style Sheets:

Style sheets that you use in the RoboHelp are preserved and continue to define the appearance of your topics. The Doc-To-Help target style sheet **C1H_Source_short.css** is added to each topic. It only defines Doc-To-Help Markup Language (D2HML) styles; it does not alter the appearance of your CSS styles.

TOC:

The RoboHelp TOC is converted to a Doc-To-Help TOC, which is automatically generated from topics (from the document tree hierarchy) if possible, customized if necessary.

Index keywords:

Index keywords are converted to Doc-To-Help index keywords. They appear on the Index tab in Doc-To-Help. Keywords are defined by D2HML hot spots, using the [C1HIndexInvisible](#) (page 263) style, in topic files, so they can be modified in Doc-To-Help either on the **Index** tab of the Project Editor or in topic HTML text using D2HML.

See Also keywords:

These keywords are converted to Doc-To-Help groups. They appear on the Index tab in Doc-To-Help. Groups are defined by D2HML hot spots, using the [C1HGroupInvisible](#) (page 265) style, so they can be modified in Doc-To-Help either on the Index tab of the Project Editor or in topic HTML text using D2HML.

Glossary:

The glossary is converted to a Doc-To-Help glossary, a multiple topic document, **Glossary.htm**.

Browse Sequence:

Browse sequence defined in RoboHelp in the Browse Sequence Editor is not converted, because the Doc-To-Help browse sequence is always determined by the topic hierarchy, that is, the document tree hierarchy. If you need to change the default browse sequence, rearrange topic files in the document tree in Doc-To-Help. Be aware that this changes your TOC when you build; you may need to customize your TOC before or after you rearrange the document tree.

Single source layouts:

RoboHelp single source layouts are converted to Doc-To-Help help targets. A property specified in a RoboHelp layout is converted to a Doc-To-Help target property only if Doc-To-Help supports that property. The following properties are converted:

Help Target	RoboHelp Property	Doc-To-Help Property
<i>HTMLHelp:</i>	Conditional Build Expression	HelpTarget.AttributeExpression
	Default Topic	Default Topic
	Default Window	HelpTarget.DefaultWindow
	Binary TOC	HelpTarget.BinaryTOC

Help Target	RoboHelp Property	Doc-To-Help Property
<i>WebHelp (converted to the Doc-To-Help NetHelp target):</i>		
	Conditional Build Expression	HelpTarget.AttributeExpression
	Default Topic	Default Topic
<i>JavaHelp:</i>		
	Conditional Build Expression	HelpTarget.AttributeExpression
	Default Topic	Default Topic
	Enable TOC	HelpTarget.ContentsTab
	Enable Index	HelpTarget.IndexTab
	Enable Search	HelpTarget.SearchTab
	Enable Favorites	HelpTarget.FavoritesTab
	TOC Label	HelpTarget.StringContents
	Index Label	HelpTarget.StringIndex
	Search Label	HelpTarget.StringSearch
	Favorites Label	HelpTarget.StringFavorites
<i>Printed Documentation (converted to the Doc-To-Help Manual target):</i>		
	Conditional Build Expression	HelpTarget.AttributeExpression
	Name of the printed documentation	HelpTarget.Title
	Include expanding text	HelpTarget.ShowExpandText
	Include drop-down text	HelpTarget.ShowDropdownText
	Chapter layout	TOC for the Manual target
<i>Project Settings Language Advanced LNG File:</i>		
	[Common]	
	Contents	StringContents
	Index	StringIndex
	Search	StringSearch
	SyncToc	StringSyncToc
	[BrowseSequence]	
	PreCaption	StringPrevious
	NextCaption	StringNext

Help Target	RoboHelp Property	Doc-To-Help Property
	[WebHelp]	
	IndexInputPromt	StringIndexCaption
	FtsInputPromt	StringSearchCaption
	TopicNotFound	StringFoundZero
	FtsBtnText	StringSearchGo
	[PrintedDoc]	
	TableOfContents	HTMLTarget(Manual).StringContents

Printed documentation chapter layout:

testIn RoboHelp, you can customize the contents of your printed documentation, the order and hierarchy of topics, and, in essence, the TOC for the Printed Documentation target. Customized chapter layout is converted to the Manual target TOC in Doc-To-Help. You can see the Manual TOC on the **Contents** tab of the Doc-To-Help Project Editor. If it differs from the main TOC of the project; the converted TOC is customized separately from the main TOC, that is, the Manual target has its [OwnTOC](#) property set to **True**. If there are more than one Print Documentation layouts in the RoboHelp project that have customized chapter layout different from the main project TOC, additional Manual targets are created in the converted Doc-To-Help project, each with its own customized TOC.

Printed documentation section layout:

This layout is not converted to Doc-To-Help, because Doc-To-Help does not support custom section breaks in the Manual target. Doc-To-Help creates default sections, separate sections for the title, contents and index and for each top-level chapter.

Printed documentation style mapping:

In RoboHelp, you can map HTML styles to Word styles for printed documentation. This mapping is not converted to Doc-To-Help, because Doc-To-Help provides a different, more versatile, mechanism of determining the target appearance of styles. If you have a custom style mapping for printed documentation in your RoboHelp project, use the `HelpTarget(Manual).StyleSheet` property to achieve the same effect in the converted Doc-To-Help project. Copy the default style sheet (for the Manual target, it is **C1H_Print_short.css**) to your project directory and modify it; define the appearance for styles or tags whose appearance you want to change in the Manual target document. Note that you only need to do it for styles whose appearance needs to be different in the printed document than it is in the online targets. If you want a style or tag to look the same in Manual as it looks in HTML-based targets, that will be done automatically, without the need to modify the Manual target style sheet.

Conditional build tags:

Conditional build targets are converted to Doc-To-Help attributes. You can see the attributes in the **Attributes** node of the Doc-To-Help Project Editor tree view. In addition to the attributes created for conditional build tags, there is a *HeadersAndFooters* attribute created to control the inclusion of headers and footers in help targets. Headers and footers are included in online targets and excluded from the Manual target.

Conditional build expressions in layouts:

Conditional Build Expressions in RoboHelp layouts are converted to Doc-To-Help property `HelpTarget.AttributeExpression`.

Topic-level conditional build tags:

Conditional build tags specified in the properties of a RoboHelp topic are converted to a D2HML topic properties hot spot, using the [CIHTopicProperties](#) (page 270) style, in the Doc-To-Help topic text.

Conditional build tags in topic text:

Conditional text in a RoboHelp topic is converted to D2HML conditional text hot spots, using the [CIHConditional](#) (page 273) style, in the Doc-To-Help topic text.

Hyperlinks:

Hyperlinks are converted to D2HML hot spots, using the [CIHJump](#) (page 252) style, in topic text.

Popups:

Pop-ups are converted to D2HML hot spots, using the [CIHPopup](#) (page 252) style, in topic text.

Keyword Links:

Keyword links are converted to D2HML hot spots, using the [CIHKeywordLink](#) (page 256) style, in topic text.

See Also Control:

These are converted to D2HML hot spots, using the [CIHGroupLink](#) (page 258) style, in topic text.

Related Topics Control:

These are converted to D2HML hot spots, using the [CIHGroupLink](#) (page 258) style, in topic text. For each Related topic control, a special group is created in the Doc-To-Help project, with the name prefix “RelatedTopics_”. These groups can be seen on the **Index** tab of the Doc-To-Help Project Editor, select **Groups** in the **Index** toolbar combo box to see the group.

Text-only Popups:

Text-only pop-ups are converted to D2HML inline pop-up hot spots using the [CIHInlinePopup](#) (page 262) style.

Expanding Text:

Expanding text is converted to D2HML inline expand hot spots using the [CIHInlineExpand](#) (page 260) style.

Drop-down Text:

Drop-down text is converted to D2HML inline drop-down hot spots using the [CIHInlineDropdown](#) (page 261) style.

Glossary hotspots:

RoboHelp and Doc-To-Help use the glossary in different ways. The RoboHelp glossary is a tool for creating expanding text hotspots; the user creates glossary hotspots in topic text explicitly, there is no automatic detection of glossary terms in topic text. In Doc-To-Help, glossary terms are detected in topic text automatically. Since the RoboHelp glossary is converted to a Doc-To-Help glossary document, glossary hotspots in topic text are detected and marked as pop-up links when Doc-To-Help builds a help target, although they are not converted to a hot spot in Doc-To-Help.

Note: Doc-To-Help displays glossary terms as pop-ups in help targets, although RoboHelp displays them as expanding text.

Dynamic HTML Effects:

Dynamic HTML effects such as Blur, Fly in, etc. are removed from HTML topic files, because they are based on proprietary RoboHelp scripts.

Note: RoboHelp scripts are removed to make HTML source clean and vendor-independent. After the conversion, you can add effects supported by your HTML editor.

Image maps:

Although image maps are not converted to Doc-To-Help-specific constructs, they are left intact in topic files and will work in online help targets as expected. Building a help target, Doc-To-Help honors links specified in AREA tags in image maps, so they point to the correct URL for the topics to which they link.

Note: The URL of a topic in the target is defined by the Doc-To-Help Topic.[URL](#) property which by default has the same value as the source topic file path.

Windows:

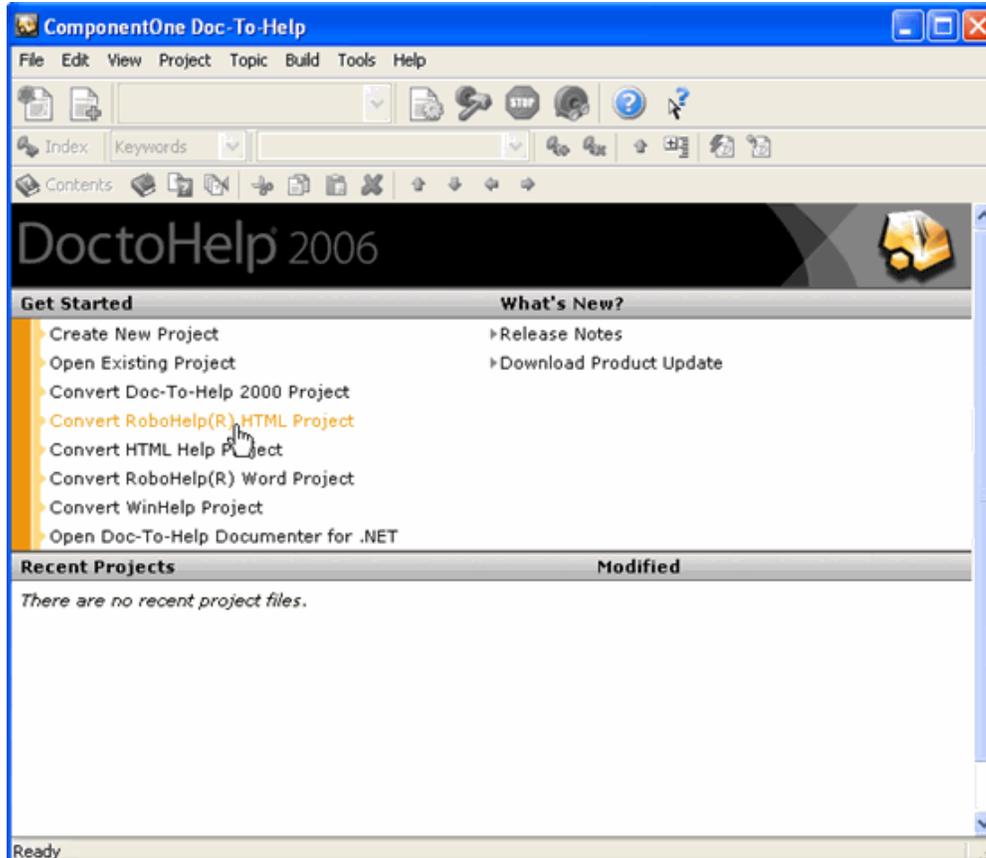
Windows specified in the RoboHelp project are converted to Doc-To-Help windows with the same names. Doc-To-Help window properties are set for HTML Help and WinHelp targets based on the values of corresponding properties in the RoboHelp project.

Map IDs:

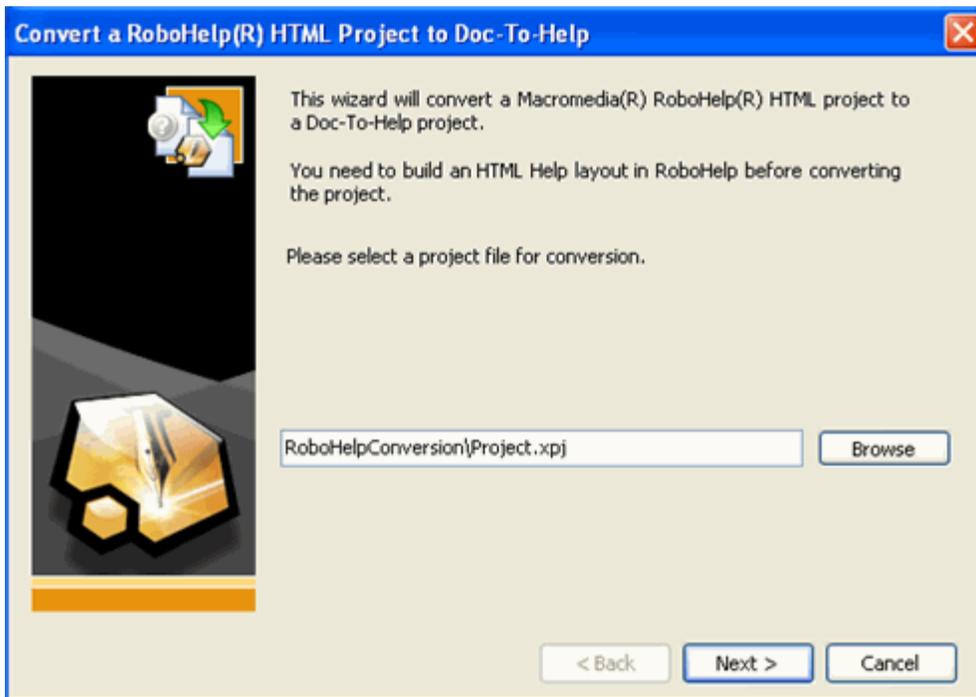
Map IDs are converted to topic context IDs in the Doc-To-Help project.

To convert a RoboHelp project:

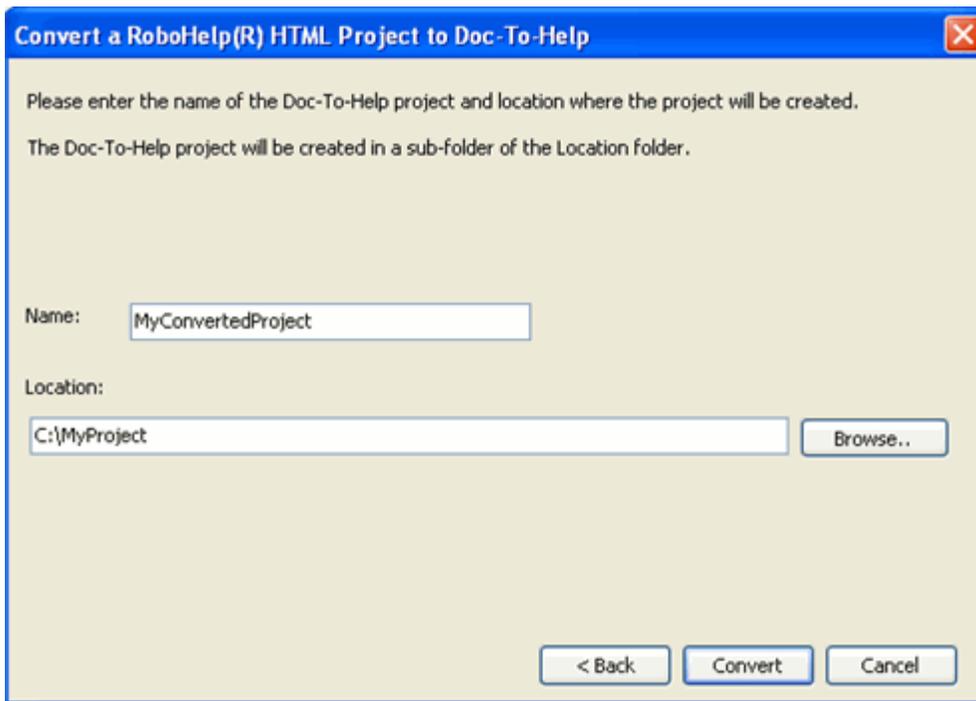
1. Generate a Microsoft HTML Help layout for your RoboHelp project. This creates the HTML Help project (.hhp) used by the Doc-To-Help RoboHelp converter.
2. Open Doc-To-Help 2006 and click **Convert RoboHelp(R) HTML Project**.



3. Browse to locate a RoboHelp project for the conversion in the **Open** dialog box.
4. Select the project file (.xpj or .mpj) and click **Open**.
5. Review the location and click **Next**.

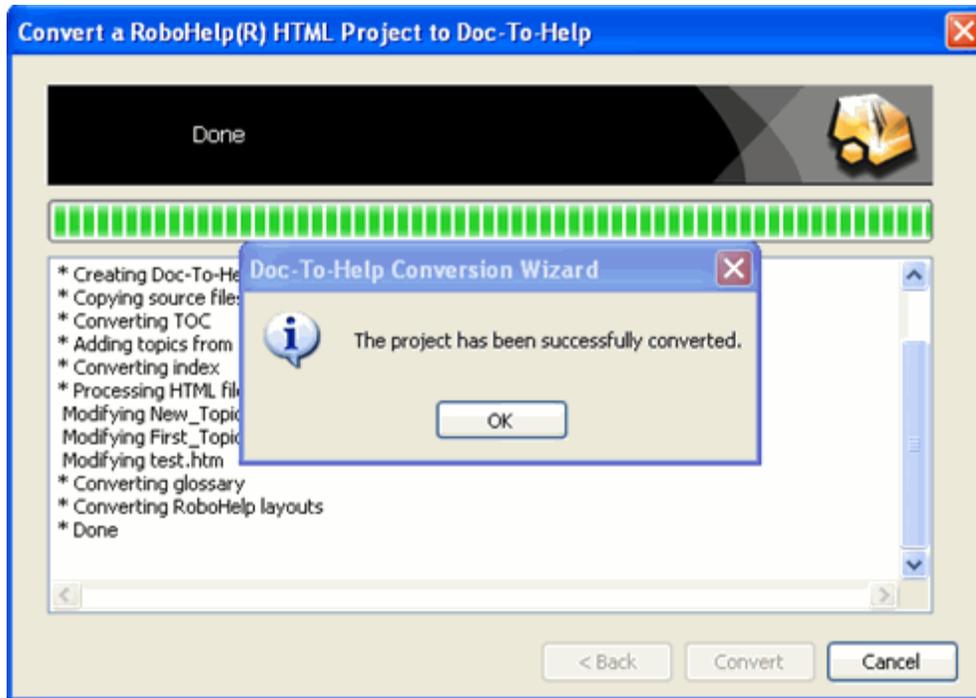


6. Enter the name of the Doc-To-Help project to be created in the **Name** text box.
7. Browse for a location for the new project.



8. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.

9. Once the conversion is complete, click **OK**.



10. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the Doc-To-Help project editor. You can begin customizing and building your help files.

Generic HTML Help Project Conversion

You can convert an HTML Help project produced with any help authoring tool to a Doc-To-Help project. You need an HTML Help project (files .HHP, .HHC, .HHK, etc) to do that. If you only have a .CHM file, you can still do it, but first you need to decompile it to an HTML Help project using either HTML Help Workshop or one of third-party CHM decompilation tools.

If your HTML Help is produced by RoboHelp and you have the source RoboHelp project, do not use the generic converter, use the RoboHelp – Doc-To-Help converter instead. Generic HTML Help converter does not convert RoboHelp-specific features.

Files:

When the generic HTML Help converter copies files from the source folder to the Doc-To-Help project folder, it only copies files listed in the [FILES] section of the HTML Help .HHP project file. This can cause broken links in the Doc-To-Help project when you build help targets if some files that are referenced in topic files are not copied. Make sure that all files you need are actually copied to the Doc-To-Help project folder. If some are missing, copy them manually.

Topics:

Converted topic files form the Doc-To-Help document tree. The tree hierarchy is based on the HTML Help TOC (.HHC file). For each TOC book that does not have a corresponding topic, an empty topic file is created, with the name prefix “_D2H_”.

TOC:

The HTML Help TOC (.HHC file) is converted to a Doc-To-Help TOC, which is automatically generated from topics (from the document tree hierarchy) if possible, customized if necessary.

Index keywords:

Keywords defined in the HTML Help index file (.HHK) and in topics are converted to Doc-To-Help index keywords. They appear on the **Index** tab in Doc-To-Help. Keywords are defined by D2HML hot spots in topic files using the [C1HIndexInvisible](#) (page 263) style, so they can be modified in Doc-To-Help either on the **Index** tab of the Project Editor or in topic HTML text using D2HML.

ALink keywords:

ALink keywords defined in topics are converted to Doc-To-Help groups. They appear on the **Index** tab in Doc-To-Help. Groups are defined by D2HML hot spots, using the [C1HGroupInvisible](#) (page 265) style, in topic files, so they can be modified in Doc-To-Help either on the **Index** tab of the Project Editor or in topic HTML text using D2HML.

Links:

Links (A-tags) to topic files are converted to D2HML hot spots, using the [C1HJump](#) (page 252) style, in topic text.

KLinks:

Keyword links (KLink HTML Help controls) are converted to D2HML hot spots, using the [C1HKeywordLink](#) (page 256) style, in topic text.

ALinks:

Associative links (ALink HTML Help controls) are converted to D2HML hot spots, using the [C1HGroupLink](#) (page 258) style, in topic text.

Related Topics Control:

These are converted to D2HML hot spots, using the [C1HGroupLink](#) (page 258) style, in topic text. For each Related topic control, a special group is created in the Doc-To-Help project, with the name prefix “RelatedTopics_”. These groups can be seen on the **Index** tab of the Doc-To-Help Project Editor. Select **Groups** in the **Index** toolbar combo box to see the group.

Image maps:

Although image maps are not converted to Doc-To-Help-specific constructs, they are left intact in topic files and will work in online help targets as expected. Building a help target, Doc-To-Help honors links specified in AREA tags in image maps, so they point to the correct URL for the topics to which they link.

Note: The URL of a topic in the target is defined by the Doc-To-Help Topic.[URL](#) property which by default has the same value as the source topic file path).

Windows:

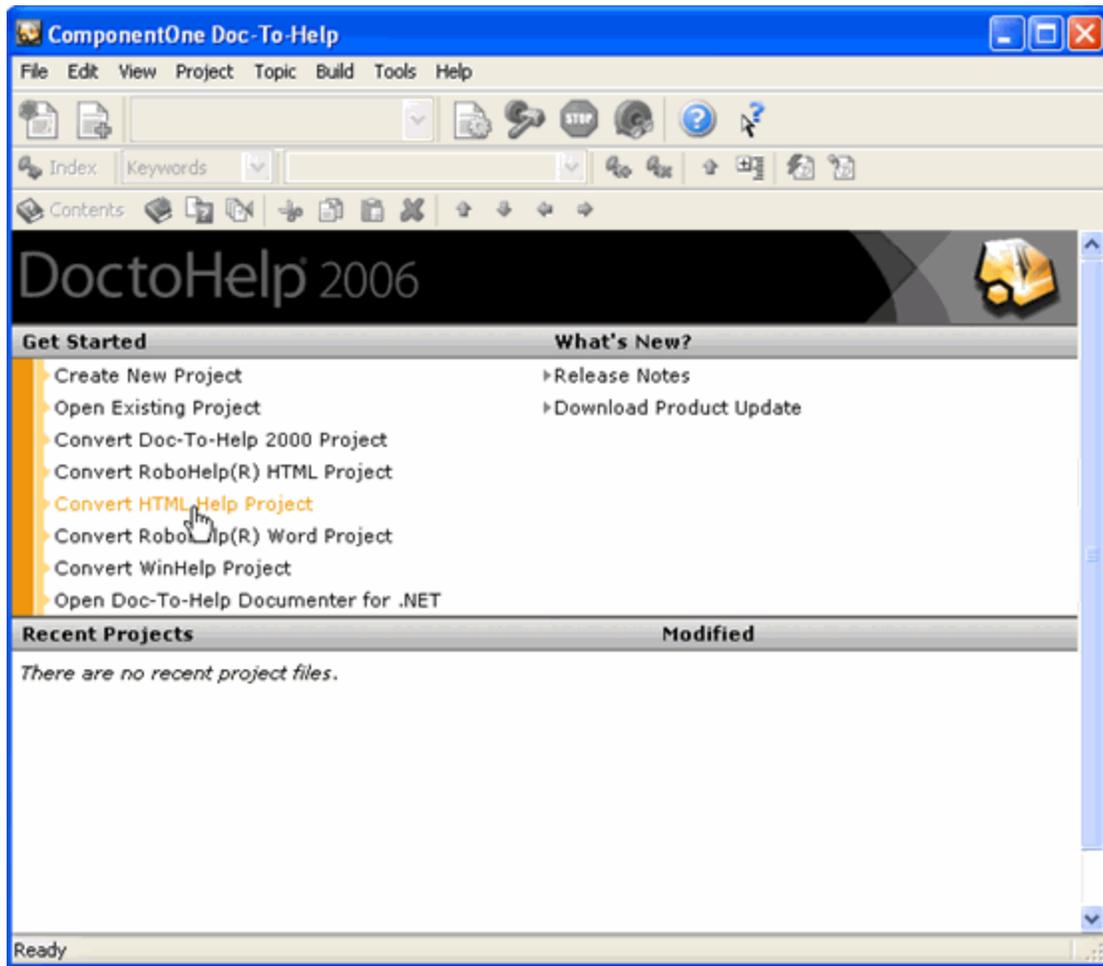
Windows specified in the HTML Help project (.HHP) are converted to Doc-To-Help windows with the same names. Doc-To-Help window properties are set for HTML Help and WinHelp targets based on the values of corresponding properties in the .HHP file.

Map IDs

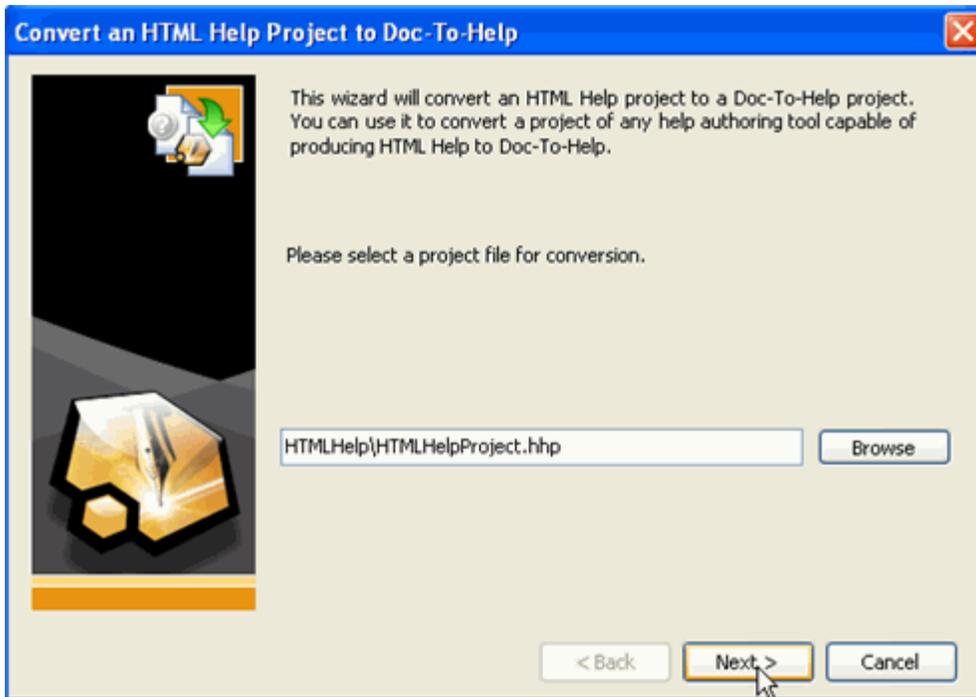
Map IDs are converted to topic context IDs in the Doc-To-Help project.

To convert an HTML Help project:

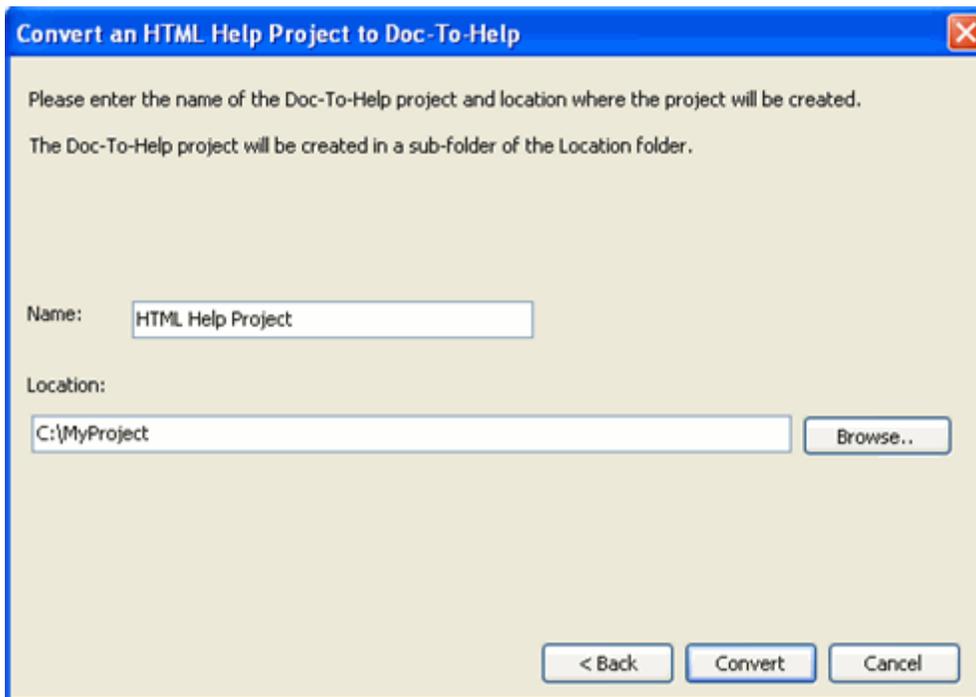
1. Open Doc-To-Help 2006 and click **Convert HTML Help Project**.



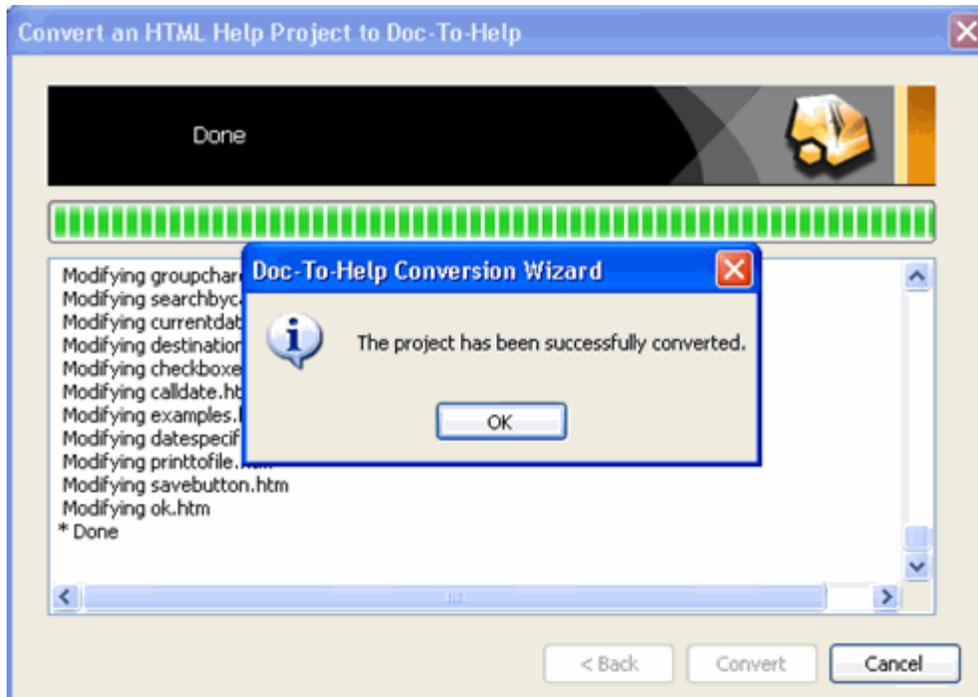
2. Browse to locate an HTML Help project (.hlp) for the conversion in the **Open** dialog box.
3. Select the file and click **Open**.
4. Review the location and click **Next**.



5. Enter the name of the Doc-To-Help project to be created in the **Name** text box.
6. Browse for a location for the new project.



7. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
8. Once the conversion is complete, click **OK**.



9. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the project editor. You can begin customizing and building your help files.

RoboHelp Word Project Conversion

To convert a RoboHelp Word (WinHelp) project to Doc-To-Help use **Convert RoboHelp Word Project** in the **File | Convert** menu.

Files:

All files in the original project folder and its subfolders are copied to the Doc-To-Help project folder except those that are used by RoboHelp for its internal purposes. This is done to ensure that external files that may be referenced in the project source files are present in the converted folder and do not cause broken links. If you see files that you know are not needed, you can delete them manually.

Source documents (Word .RTF files included in the project) are converted to Doc-To-Help format. For example, WinHelp hotspots are converted to D2HML hotspots, and so on; please see the following conversion information. Other files are copied to the destination directory unchanged.

Source documents located outside the original project directory are not copied and not converted, with a warning issued in the conversion log.

Graphic files (help images, included in statements such as {bmc}) located outside the project directory remain in their places and their paths in the documents are changed to an absolute path with a warning in the conversion log.

Files included in the BAGGAGE section of the project are not copied to the destination directory unless they are located inside the source project directory. If you need those files, copy them manually to an appropriate location in the project directory.

A special directory `_defbmp` is created in the converted project directory containing standard WinHelp bitmaps (bitmaps supplied by Help Workshop) such as `bullet.bmp`, `shortcut.bmp`, etc.

Styles and templates:

Styles and style appearance in the source documents are preserved in conversion. Converted documents have a Doc-To-Help template, `C1H_NOMARGIN.DOT`, attached to them, so the author can use Doc-To-Help styles. However, the template does not change the appearance of the styles already used in the source document, because the check box

Automatically update document styles in the converted Word document's **Tools | Options** menu is unchecked. If you check that check box, the styles appearance can change, because it will be defined by the C1H_NOMARGIN.DOT template.

The target template, which is set through the HelpTarget.[Template](#), property in the converted Doc-To-Help project is set to (None) to preserve the appearance of the source documents in the help target. You can change it to one of the standard Doc-To-Help templates or to your own customized template if you want to control target appearance by a template.

The style of the heading of each topic in Doc-To-Help must be one of the *active paragraph styles*, those styles that define a topic when Doc-To-Help compiles the document. For a topic with the first paragraph formatted with a style without an outline level (*non-active style*), its first paragraph is reformatted with a new style with the postfix (**Topic**) added in the end of the style name, the new style is inherited from the original style. For styles with outline levels (*active styles*) that are used in the original documents in a mixed way, both for formatting topic headings and for formatting paragraphs that are not topic headings, the paragraphs that are not topic headings are reformatted with a new style with (**Nontopic**) added in the end of the style name, the new style is inherited from the original style.

Topics:

A Doc-To-Help topic is created in the converted project for each WinHelp topic. A WinHelp topic ID becomes a topic link tag and the value of the topic's [AsciiName](#) property. If a topic has an alias in the project file, that alias is also added to the collection of the topic's link tags.

Topic properties and TopicType:

Some topic properties are implemented in Doc-To-Help via topic types. For example, you can't assign windows to individual topics directly, but you can set Topic.[TopicType](#) to a topic type that has a specific window in the TopicType.[Window](#) property. To enable this mechanism, topic types are created in the converted project as necessary, having the necessary property values, and these topic types are assigned to the Topic.[TopicType](#) property as needed, to specify various topic properties.

Topic properties:

Topic title (\$ footnote) is converted to the Topic.[DisplayTitle](#) property specified by a D2HML hot spot (style **C1HTopicProperties**) in topic text.

Topic title specified in the table of contents (TOC) is converted to the Topic.[ContentsTitle](#) property specified by a D2HML hot spot (style **C1HContentsTitle**) in topic text.

Topic window (> footnote) is converted to the Topic.TopicType.[Window](#) property.

Topic macro (! footnote) is converted to the Topic.[WinHelpMacro](#) property.

Topic.TopicType.[Nonscrolling](#) is set to **True** when a topic heading paragraph in the source document has its Word paragraph format setting **Keep with next = True**.

Topic.TopicType.[Midtopic](#) is set to **True** for topics that are *mid-topics*, that is, bookmarks in their parent topic rather than separate topics. According to WinHelp rules, such topics are characterized by the absence of a page break before their first paragraph.

TOC:

Table of Contents is converted to Doc-To-Help TOC.

Index keywords:

Index keywords are converted to Doc-To-Help index keywords. They appear on the Index tab in Doc-To-Help. Keywords are defined by D2HML hot spots in topic text (style **C1HIndexInvisible**), so they can be modified in Doc-To-Help either on the Index tab of the Project Editor or in the topic text using D2HML.

See Also (A-keywords):

A-keywords are converted to Doc-To-Help groups. They appear on the Index tab in Doc-To-Help. Groups are defined by D2HML hot spots in topic text (style **C1HGroupInvisible**), so they can be modified in Doc-To-Help either on the Index tab of the Project Editor or in the topic text using D2HML.

Links:

Jump and pop-up links (including those defined by macros JI, PI) are converted to D2HML hot spots (styles **C1HJump**, **C1HPopup**) in topic text.

Keyword links:

Keyword links (macros KL, JK) are converted to D2HML hot spots (style **C1HKeywordLink**) in topic text.

A-links (See Also links):

A-links (macro AL) are converted to D2HML hot spots (style **C1HGroupLink**) in topic text.

Windows:

Windows specified in the project are converted to Doc-To-Help windows with the same names. Doc-To-Help window properties are set for a WinHelp target based on the values of corresponding properties in the source project.

Map IDs:

Map IDs are converted to topic context IDs in the Doc-To-Help project. Context IDs are defined by D2HML hot spots in topic text (style **C1HContextID**), so they can be modified in Doc-To-Help either in the topic property browser of the Project Editor or in the topic text using D2HML.

Browse sequence:

Topics included in a browse sequence (having + footnotes) are included in the Doc-To-Help navigation sequence by setting Topic.TopicType.[AutoNavigate](#) = **True**. However, custom browse sequence (+ footnotes with browse code) is not converted, because Doc-To-Help browse sequence is always determined by the topic hierarchy, that is, by positioning of topics inside documents.

RoboHelp document default browse sequence:

RoboHelp document default browse sequence is not converted, because Doc-To-Help browse sequence is always determined by the topic hierarchy, that is, by positioning of topics inside documents.

Conditional build tags:

Conditional build tags (* footnotes) are converted to Doc-To-Help attributes. You can see the attributes in the Attributes node of the Doc-To-Help Project Editor tree view.

Conditional build tags in RoboHelp layouts:

Conditional Build Expressions in RoboHelp layouts are converted to Doc-To-Help property HelpTarget.AttributeExpression.

Topic-level conditional build tags:

Conditional build tags specified in the properties of a WinHelp topic (* footnotes) are converted to a D2HML topic properties hot spot (style **C1HTopicProperties**) in the Doc-To-Help topic text.

Conditional build tags in topic text:

RoboHelp conditional text (specified as special comments in the source document) is converted to Doc-To-Help conditional text comments in topic text.

Macros:

Macros defined in the project file for the entire help (CONFIG section) are converted to the [WinHelpMacro](#) property in **Project Settings**.

Macros defined in the project file for a window (CONFIG:window section) are converted to the Window.[WinHelpMacro](#) property.

Macros defined for a topic (! Footnote) are converted to the Topic.[WinHelpMacro](#) property.

Macros defined in macro hotspots in topic text are converted to Doc-To-Help WinHelp macro comments in topic text unless they are jump, pop-up or K-/A-links. In the latter case they are converted to D2HML hotspots.

RoboHelp proprietary macros are omitted, not converted. This is done to keep the converted project vendor-independent.

Notes:

WinHelp macros have effect only in the WinHelp target; they are ignored in all other targets.

Doc-To-Help has a limitation of 255 characters for macro string length. Macro strings that exceed that limit are truncated, with a warning in the conversion log.

Help images:

Help images (bitmap statements {bmc}, {bml}, {bmr}) are converted to linked pictures (Word {INCLUDEPICTURE} field).

Hotspot images:

Hotspot images (SHED images, segmented hypergraphics) are converted to Doc-To-Help image maps, images with hot spots. Image maps can then be explored and edited using the Doc-To-Help [Image Map Editor](#).

Buttons:

Help buttons ({button} statements) are converted to D2HML hot spots (styles **C1HJump**, **C1HPopup**, **C1HkeywordLink**, **C1HGroupLink**) with property **DisplayAsButton = True**.

RoboHelp graphical buttons:

RoboHelp graphical buttons are converted to images formatted with D2HML hot spot styles **C1HJump**, **C1HPopup**, **C1HkeywordLink**, **C1HGroupLink**.

Help multimedia and embedded windows:

Help multimedia ({mci} statement) is not converted because it is not supported in Doc-To-Help.

Embedded windows ({ew*} statements) are not supported with the exception of those generated by RoboHelp graphical buttons, which are converted to images formatted as Doc-To-Help hotspots.

RoboHelp glossary:

The RoboHelp glossary is converted to a Doc-To-Help glossary, a Word document Glossary.doc. Glossary topics (glossary term definitions) are assigned special link tags of the form RhGlossDef_... necessary to preserve pop-up links to glossary terms created by the RoboHelp Glossary Hotspot Wizard. You can add new glossary terms to the glossary after conversion; they will be used in Doc-To-Help according to the Doc-To-Help glossary rules. You do not need to assign link tags to glossary terms that you add after conversion.

Note that in addition to glossary hotspots converted from those existing in the original project, you may find new glossary pop-ups in the help built by Doc-To-Help. This is because the Doc-To-Help glossary is automatic in the sense that every occurrence of a glossary term automatically generates a pop-up. You can disable this automatic pop-up generation by formatting a glossary term with the **D2HNoGloss** style.

RoboHelp single source layouts:

RoboHelp single source layouts are converted to Doc-To-Help help targets. A property specified in a RoboHelp layout is converted to a Doc-To-Help target property only if Doc-To-Help supports that property. The following properties are converted:

Help Target	RoboHelp Property	Doc-To-Help Property
<i>HTMLHelp:</i>	Conditional Build Expression	HelpTarget.AttributeExpression
	Default Topic	Default Topic
	Background Watermark	HelpTarget.BackgroundImage

WebHelp (converted to the Doc-To-Help NetHelp target):

Conditional Build Expression	HelpTarget.AttributeExpression
Default Topic	Default Topic
Background Watermark	HelpTarget.BackgroundImage

<i>JavaHelp:</i>	Conditional Build Expression	HelpTarget.AttributeExpression
	Default Topic	Default Topic
	Navigation Pane TOC	HelpTarget.ContentsTab
	Navigation Pane Index	HelpTarget.IndexTab
	Navigation Pane Full-text Search	HelpTarget.SearchTab

Printed Documentation (converted to the Doc-To-Help Manual target):

Conditional Build Expression	HelpTarget.AttributeExpression
Name of the printed documentation	HelpTarget.Title
Chapter layout	TOC for the Manual target

Project Settings | Language | Advanced | LNG File:

[Common]

Contents	StringContents
Index	StringIndex
Search	StringSearch
SyncToc	StringSyncToc

[BrowseSequence]

PreCaption	StringPrevious
NextCaption	StringNext

[WebHelp]

IndexInputPromt	StringIndexCaption
FtsInputPromt	StringSearchCaption
TopicNotFound	StringFoundZero
FtsBtnText	StringSearchGo

[PrintedDoc]

TableOfContents	HTMLTarget(Manual).StringContents
-----------------	-----------------------------------

RoboHelp printed documentation chapter layout:

In RoboHelp, you can customize the contents of your printed documentation, the order and hierarchy of topics, in essence, customize the TOC for the Printed Documentation target. Customized chapter layout is converted to the Manual target TOC in Doc-To-Help. You can see the Manual TOC on the **Contents** tab of the Doc-To-Help Project Editor. If it differs from the main TOC of the project, the converted TOC is customized separately from the main TOC, that is, the Manual target has its [OwnTOC](#) property set to **True**. If there are more than one Print Documentation layouts in the RoboHelp project that have a customized chapter layout different from the main project TOC, additional Manual targets are created in the converted Doc-To-Help project, each with its own customized TOC.

RoboHelp printed documentation section layout:

RoboHelp section layout for printed documentation is not converted to Doc-To-Help, because Doc-To-Help does not support custom section breaks in the Manual target. Doc-To-Help creates default sections, separate sections for the title, contents and index and for each top-level chapter.

RoboHelp printed documentation style mapping:

In RoboHelp, you can map styles in your source documents to styles in your template for printed documentation. This mapping is not converted to Doc-To-Help, because Doc-To-Help provides a different, in fact, more versatile, mechanism of determining the target appearance of styles. If you have a custom style mapping for printed documentation in your RoboHelp project, use the HelpTarget(Manual).[Template](#) property to achieve the same effect in the converted Doc-To-Help project. Note that you only need to do this if you want style appearance in printed documentation to be different from that in the online targets.

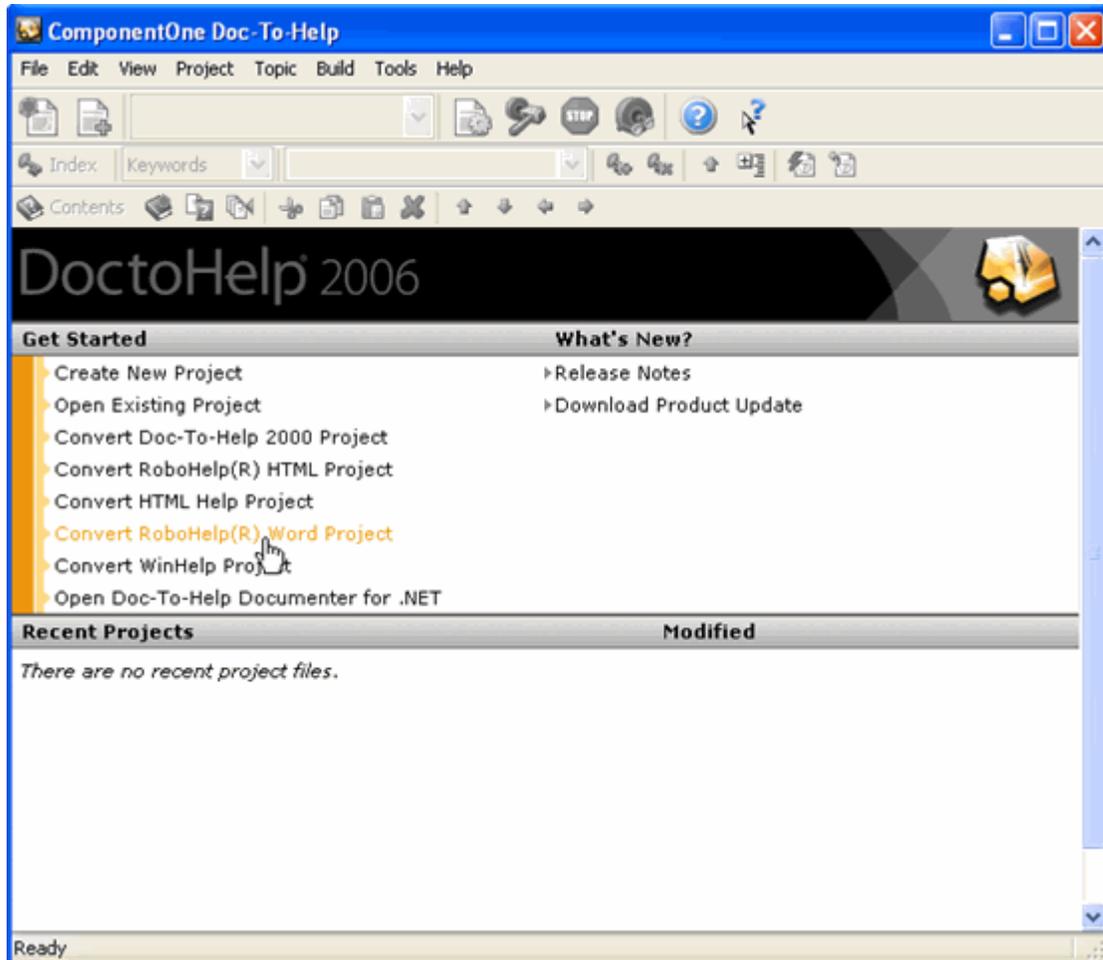
Unsupported RoboHelp topic properties:

Background and nonscrolling watermark and alignment are not converted, because they are not supported by Doc-To-Help

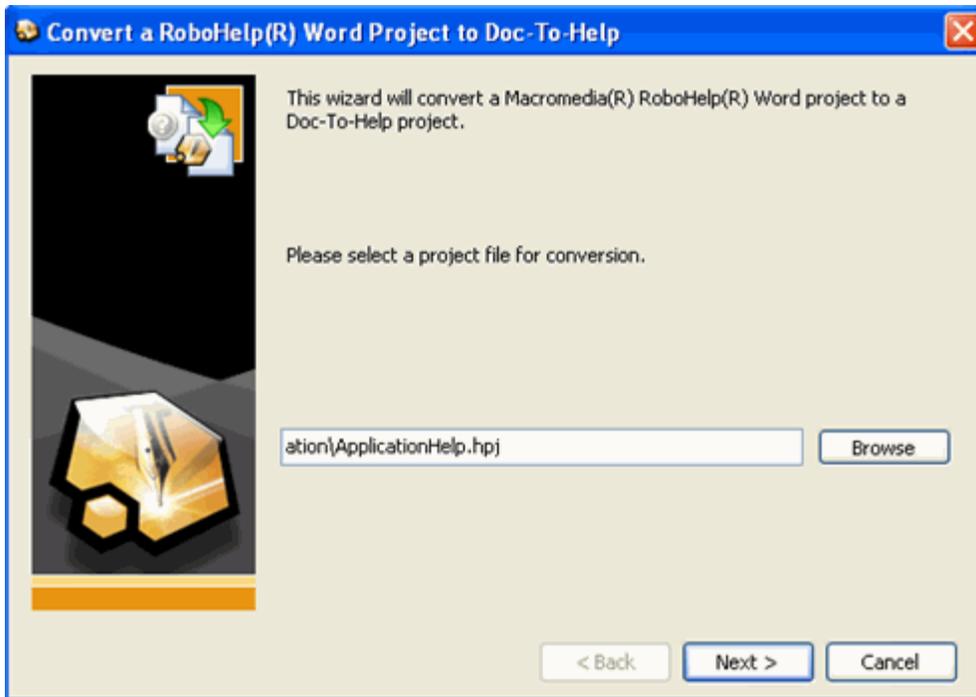
Custom topic browse sequence is not converted because browse sequence in Doc-To-Help is always determined by the positioning of topics inside documents. However, topics included in a RoboHelp browse sequence are included in Doc-To-Help navigation sequence by setting `Topic.TopicType.AutoNavigate = True`.

To Convert a RoboHelp Word project:

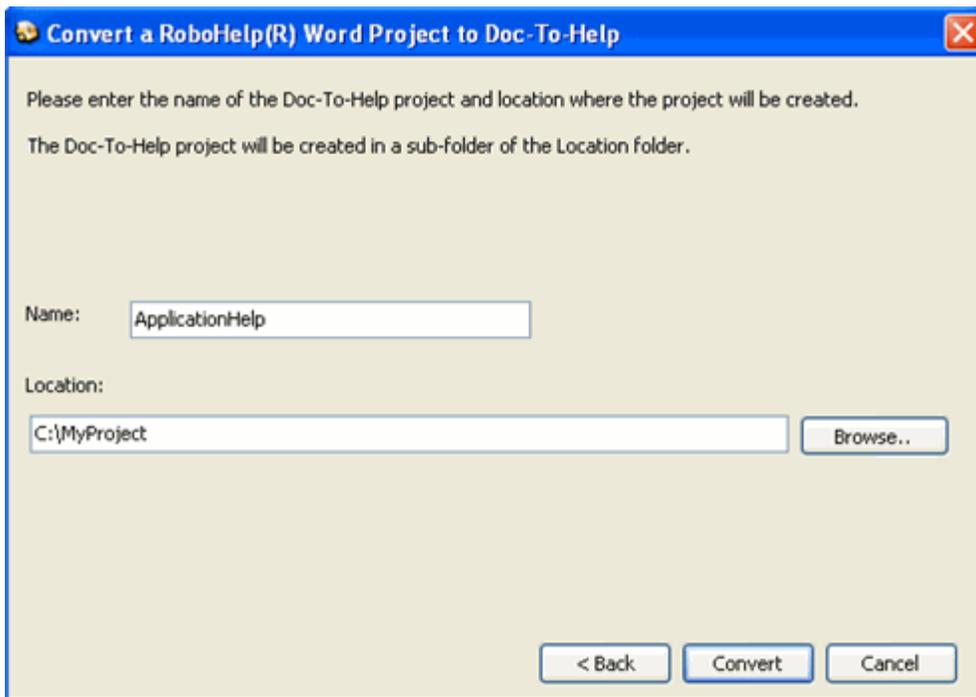
1. Open Doc-To-Help 2006 and click **Convert RoboHelp(R) Word Project**.



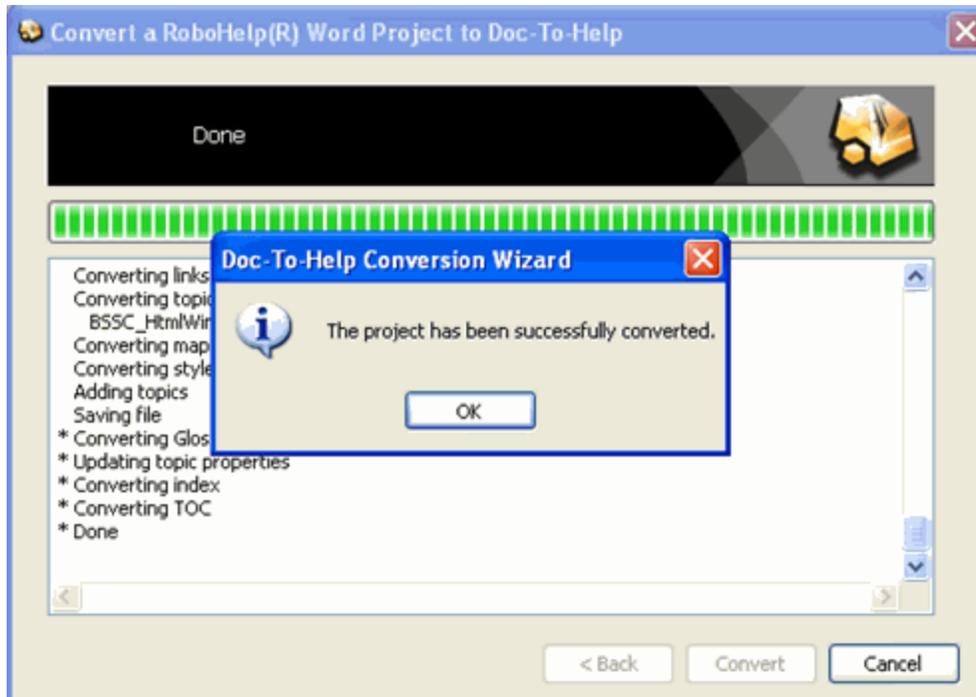
2. Browse, select an **.hpl** project file and click **Open**.
3. Review the location and click **Next**.



4. Enter the name of the Doc-To-Help project to be created in the **Name** text box and browse for a location for the new project.



5. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
6. Once the conversion is complete, click **OK**.



7. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the Doc-To-Help project editor. You can begin customizing and building your help files.

WinHelp Project Conversion

You can convert a WinHelp project produced with any help authoring tool other than RoboHelp to a Doc-To-Help project. You need a WinHelp project (files .HPJ, .CNT, etc.) to do that. If you only have an .HLP file, you can still do it, but first you need to decompile it to a WinHelp project using, for example, the freeware HelpDeco utility.

If your WinHelp is produced by RoboHelp and you have the source RoboHelp project, do not use the generic converter, use the RoboHelp Word – Doc-To-Help converter instead. The generic WinHelp converter does not convert RoboHelp-specific features.

Files:

All files in the original project folder and its subfolders are copied to the Doc-To-Help project folder. This is done to ensure that external files that may be referenced in the project source files are present in the converted folder and don't cause broken links. If you see files that you know are not needed, you can delete them manually.

Source documents (Word .RTF files included in the project) are converted to Doc-To-Help format. For example, WinHelp hotspots are converted to D2HML hotspots, and so on; please see the following conversion information. Other files are copied to the destination directory unchanged.

Source documents located outside the original project directory are not copied and not converted, with a warning issued in the conversion log.

Graphic files (help images, included in statements such as {bmc}) located outside the project directory remain in their places and their paths in the documents are changed to an absolute path with a warning in the conversion log.

Files included in the BAGGAGE section of the project are not copied to the destination directory unless they are located inside the source project directory. If you need those files, copy them manually to an appropriate location in the project directory.

A special directory _defbmp is created in the converted project directory containing standard WinHelp bitmaps (bitmaps supplied by Help Workshop) such as bullet.bmp, shortcut.bmp, etc.

Styles and templates:

Styles and style appearance in the source documents are preserved in conversion. Converted documents have a Doc-To-Help template, C1H_NOMARGIN.DOT, attached to them, so the author can use Doc-To-Help styles. However, the template does not change the appearance of the styles already used in the source document, because the check box *Automatically update document styles* in the converted Word document's **Tools | Options** menu is unchecked. If you check that check box, the styles appearance can change, because it will be defined by the C1H_NOMARGIN.DOT template.

The target template, which is set using the HelpTarget.[Template](#) property in the converted Doc-To-Help project, is set to **(None)** to preserve the appearance of the source documents in the help target. You can change it to one of the standard Doc-To-Help templates or to your own customized template if you want to control target appearance by a template.

The style of the heading of each topic in Doc-To-Help must be one of the active paragraph styles, those styles that define a topic when Doc-To-Help compiles the document. For a topic with the first paragraph formatted with a style without an outline level (non-active style), its first paragraph is reformatted with a new style with the postfix **(Topic)** added in the end of the style name; the new style is inherited from the original style. For styles with outline levels (active styles) that are used in the original documents in a mixed way, both for formatting topic headings and for formatting paragraphs that are not topic headings, the paragraphs that are not topic headings are reformatted with a new style with **(Nontopic)** added in the end of the style name; the new style is inherited from the original style.

Topics:

A Doc-To-Help topic is created in the converted project for each WinHelp topic. A WinHelp topic ID becomes a topic link tag and the value of the topic's [AsciiName](#) property. If a topic has an alias in the project file, that alias is also added to the collection of the topic's link tags.

Topic properties and TopicType:

Some topic properties are implemented in Doc-To-Help via topic types. For example, you can't assign windows to individual topics directly, but you can set Topic.[TopicType](#) to a topic type that has a specific window in the TopicType.[Window](#) property. To enable this mechanism, topic types are created in the converted project as necessary, having the necessary property values, and these topic types are assigned to the Topic.[TopicType](#) property, as needed, to specify various topic properties.

Topic properties:

Topic title (\$ footnote) is converted to the Topic.[DisplayTitle](#) property specified by a D2HML hot spot (style **C1HTopicProperties**) in topic text.

Topic title specified in table of contents (TOC) is converted to the Topic.[ContentsTitle](#) property specified by a D2HML hot spot (style **C1HContentsTitle**) in topic text.

Topic window (> footnote) is converted to the Topic.TopicType.[Window](#) property.

Topic macro (! footnote) is converted to the Topic.[WinHelpMacro](#) property.

Topic.TopicType.[Nonscrolling](#) is set to **True** when a topic heading paragraph in the source document has its Word paragraph format setting **Keep with next = True**.

Topic.TopicType.[Midtopic](#) is set to **True** for topics that are mid-topics, that is, bookmarks in their parent topic rather than separate topics. According to WinHelp rules, such topics are characterized by the absence of a page break before their first paragraph.

TOC:

Table of Contents is converted to Doc-To-Help TOC.

Index keywords:

Index keywords are converted to Doc-To-Help index keywords. They appear on the **Index** tab in Doc-To-Help. Keywords are defined by D2HML hot spots in topic text (style **C1HIndexInvisible**), so they can be modified in Doc-To-Help either on the **Index** tab of the Project Editor or in the topic text using D2HML.

See Also (A-keywords):

A-keywords are converted to Doc-To-Help groups. They appear on the **Index** tab in Doc-To-Help. Groups are defined by D2HML hot spots in topic text (style **C1HGroupInvisible**), so they can be modified in Doc-To-Help either on the **Index** tab of the Project Editor or in the topic text using D2HML.

Links:

Jump and pop-up links (including those defined by macros JI, PI) are converted to D2HML hot spots (styles **C1HJump**, **C1HPopup**) in topic text.

Keyword links:

Keyword links (macros KL, JK) are converted to D2HML hot spots (style **C1HKeywordLink**) in topic text.

A-links (See Also links):

A-links (macro AL) are converted to D2HML hot spots (style **C1HGroupLink**) in topic text.

Windows:

Windows specified in the project are converted to Doc-To-Help windows with the same names. Doc-To-Help window properties are set for a WinHelp target based on the values of corresponding properties in the source project.

Map IDs:

Map IDs are converted to topic context IDs in the Doc-To-Help project. Context IDs are defined by D2HML hot spots in topic text (style **C1HContextID**), so they can be modified in Doc-To-Help either in the topic property browser of the Project Editor or in the topic text using D2HML.

Browse sequence:

Topics included in a browse sequence (having + footnotes) are included in the Doc-To-Help navigation sequence by setting Topic.TopicType.[AutoNavigate](#) = **True**. However, custom browse sequence (+ footnotes with browse code) is not converted, because Doc-To-Help browse sequence is always determined by the topic hierarchy, that is, by positioning of topics inside documents.

Conditional build tags:

Conditional build tags (* footnotes) are converted to Doc-To-Help attributes. You can see the attributes in the **Attributes** node of the Doc-To-Help Project Editor tree view.

Conditional build tags in project sections INCLUDE, EXCLUDE:

In the generic WinHelp converter, attributes created from conditional build tags are checked or unchecked in the help targets in the Doc-To-Help project depending on whether they appear in [INCLUDE] or in the [EXCLUDE] section of the project file.

Topic-level conditional build tags:

Conditional build tags specified in the properties of a WinHelp topic (* footnotes) are converted to a D2HML topic properties hot spot (style **C1HTopicProperties**) in the Doc-To-Help topic text.

Macros:

Macros defined in the project file for the entire help (CONFIG section) are converted to the [WinHelpMacro](#) property in **Project Settings**.

Macros defined in the project file for a window (CONFIG:window section) are converted to the Window.[WinHelpMacro](#) property.

Macros defined for a topic (! Footnote) are converted to the Topic.[WinHelpMacro](#) property.

Macros defined in macro hotspots in topic text are converted to Doc-To-Help WinHelp macro comments in topic text unless they are jump, pop-up or K-/A-links. In the latter case they are converted to D2HML hotspots.

Notes:

WinHelp macros have effect only in the WinHelp target; they are ignored in all other targets.

Doc-To-Help has a limitation of 255 characters for macro string length. Macro strings that exceed that limit are truncated, with a warning in the conversion log.

Help images:

Help images (bitmap statements {bmc}, {bml}, {bmr}) are converted to linked pictures (Word {INCLUDEPICTURE} field).

Hotspot images:

Hotspot images (SHED images, segmented hypergraphics) are converted to Doc-To-Help image maps, images with hot spots. Image maps can then be explored and edited using the Doc-To-Help [Image Map Editor](#).

Buttons:

Help buttons ({button} statements) are converted to D2HML hot spots (styles **C1HJump**, **C1HPopup**, **C1HkeywordLink**, **C1HGroupLink**) with property **DisplayAsButton = True**.

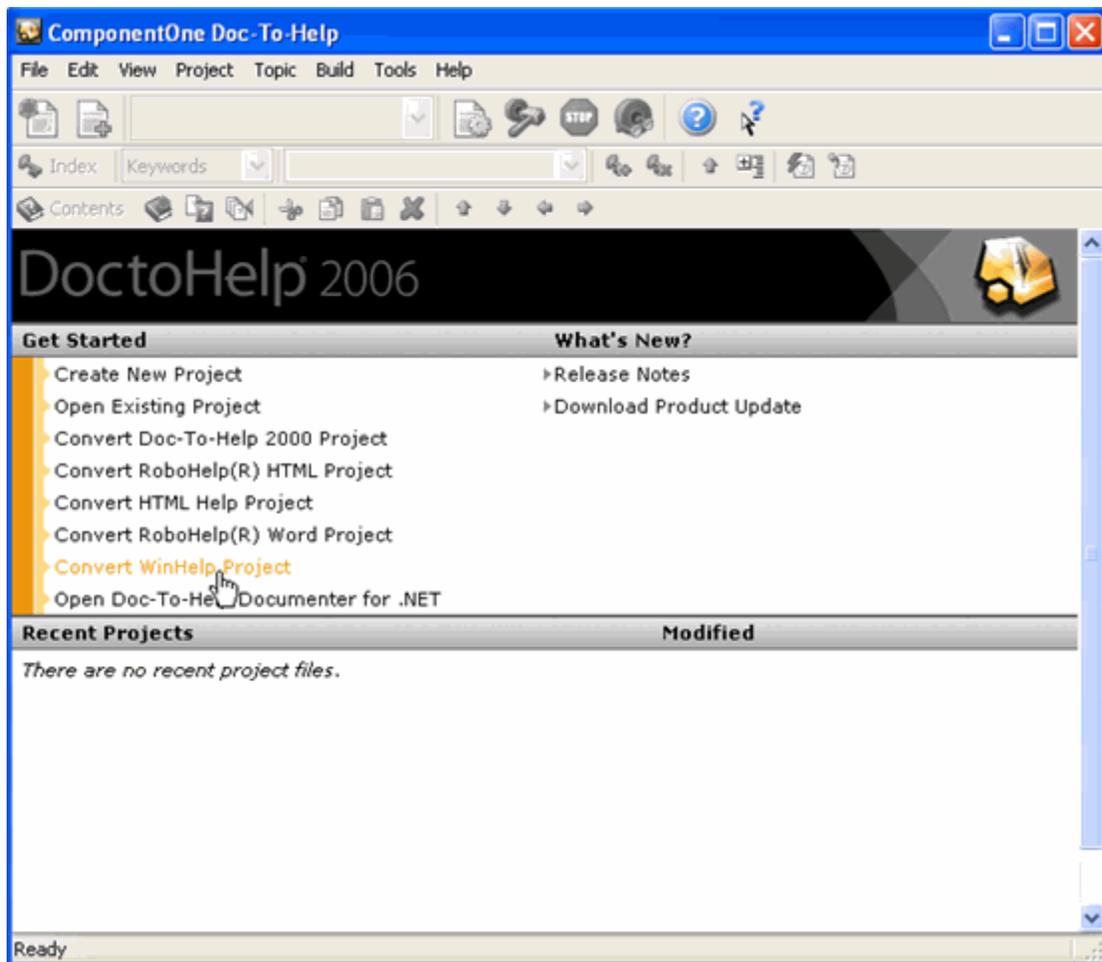
Help multimedia and embedded windows:

Help multimedia ({mci} statement) is not converted because it is not supported in Doc-To-Help.

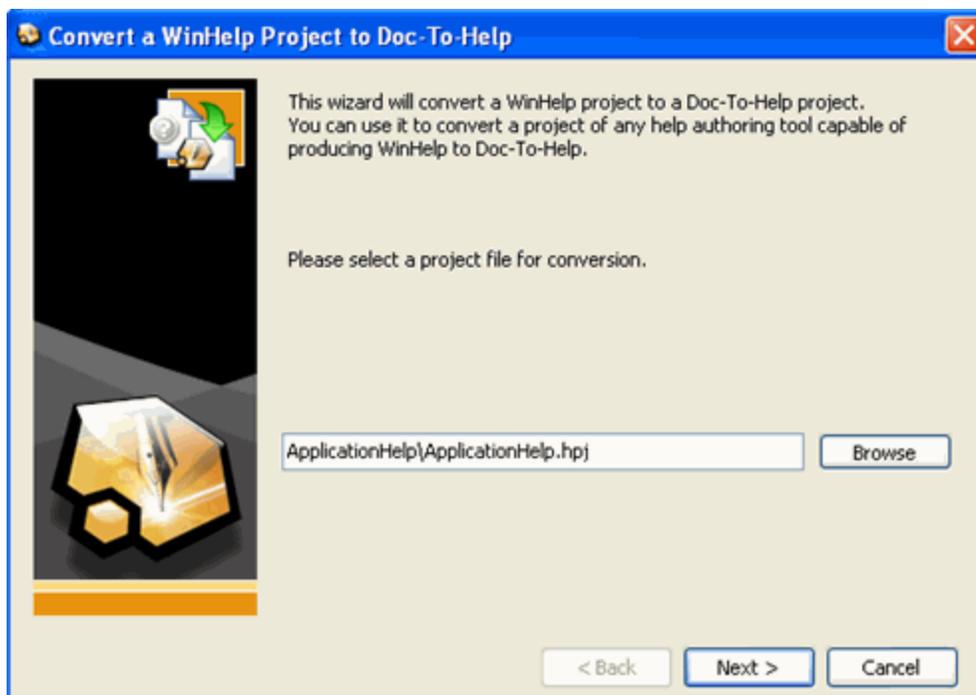
Embedded windows ({ew*} statements) are not supported, with the exception of those generated by RoboHelp graphical buttons, which are converted to images formatted as Doc-To-Help hotspots (RoboHelp Word converter only).

To convert a WinHelp project:

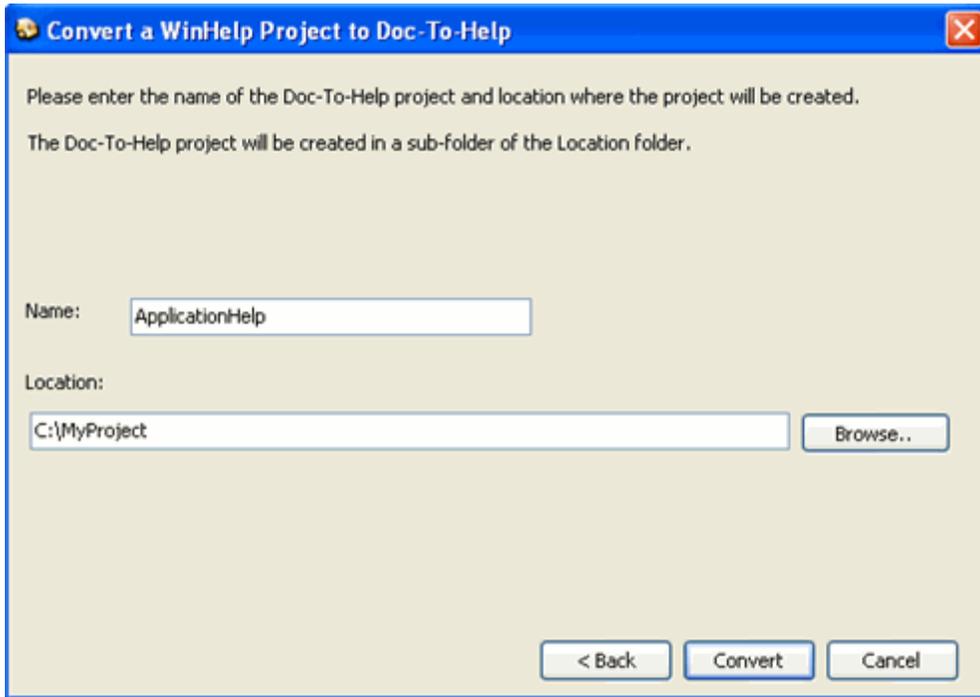
1. Open Doc-To-Help 2006 and click **Convert WinHelp Project**.



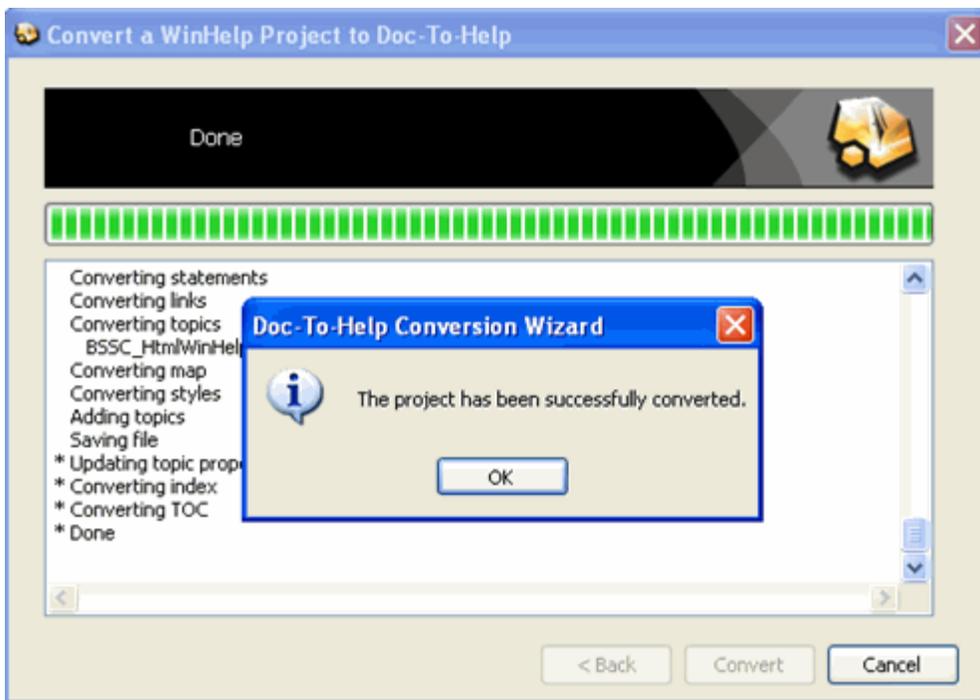
2. Browse, select an **.hjp** project file and click **Open**.
3. Review the location and click **Next**.



4. Enter the name of the Doc-To-Help project to be created in the **Name** text box.
5. Browse for a location for the new project.



6. Click the **Convert** button to begin the conversion. A window appears showing the conversion process.
7. Once the conversion is complete, click **OK**.



8. Click the **Close** button to close the Convert dialog box. Your new Doc-To-Help project opens in the project editor. You can begin customizing and building your help files.

Opening Documenter for .NET

Click the **Open Doc-To-Help Documenter for .NET** link in the Doc-To-Help Start Page. Documenter opens and you have the option to open a recent project, create a new project or open a project. For additional information, see [Documenter for .NET](#) (page 449).

Opening the Release Notes

Click the **Release Notes** link in the Doc-To-Help Start Page. The Doc-To-Help 2006 readme.txt document opens, listing all of the new features in the current version of Doc-To-Help.

Downloading Product Updates

Click the **Download Product Update** link in the Doc-To-Help Start Page. Your default HTML browser opens to the ComponentOne Download Center. Here you can select and download a product for the latest features and updates.

Opening a Recent Project

From the Doc-To-Help Start Page:

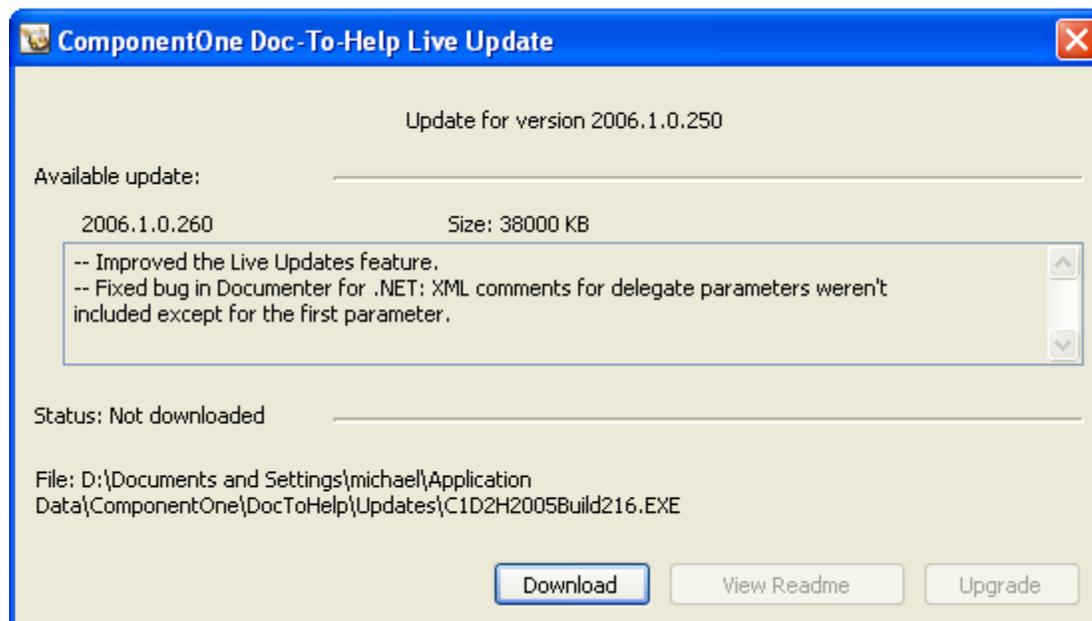
Select any file under the *Recent Projects* in the Doc-To-Help Start Page.

From the Doc-To-Help Project Editor:

Doc-To-Help keeps track of the last nine project files that you opened. You can open one of these project files directly by clicking one of the numbered commands on the **File** menu in the Doc-To-Help project editor.

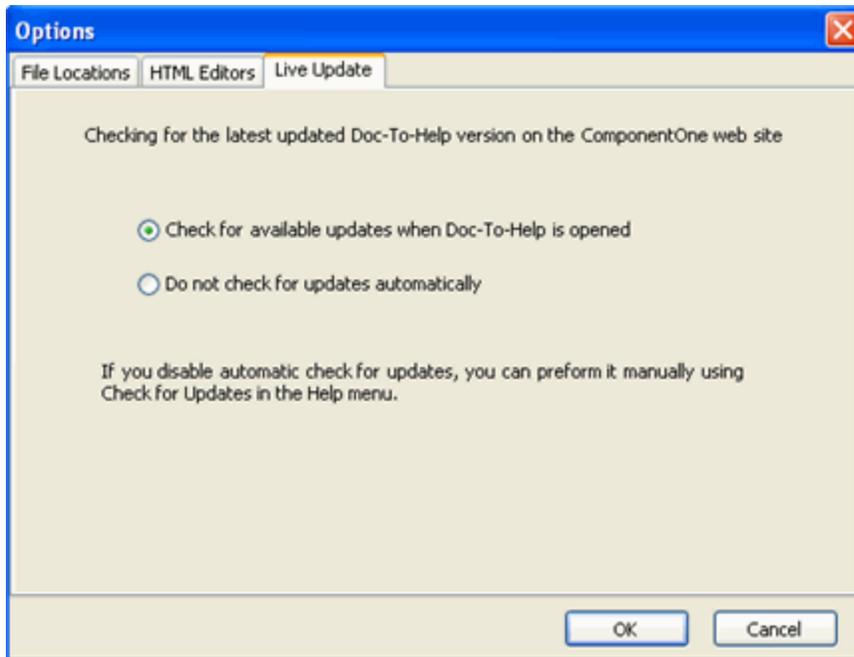
Using Live Update

Live Update is a utility that automatically checks for new versions of Doc-To-Help and downloads them to your computer. Live Update is enabled by default, so if there is a newer version of Doc-To-Help available when you open Doc-To-Help, it prompts you to download it.



To disable live update:

1. Open Doc-To-Help.
2. Click **Tools | Options**. The **Options** dialog box appears.
3. Select the **Live Update** tab.



4. Select **Do not check for updates automatically** and click **OK**. Live Update is disabled. You can always enable Live Update at a later time by selecting **Check for available updates when Doc-To-Help is opened** from the **Live Update** tab in the **Options** dialog box.

If Live Update is disabled, you may choose to manually check for updates. To do this:

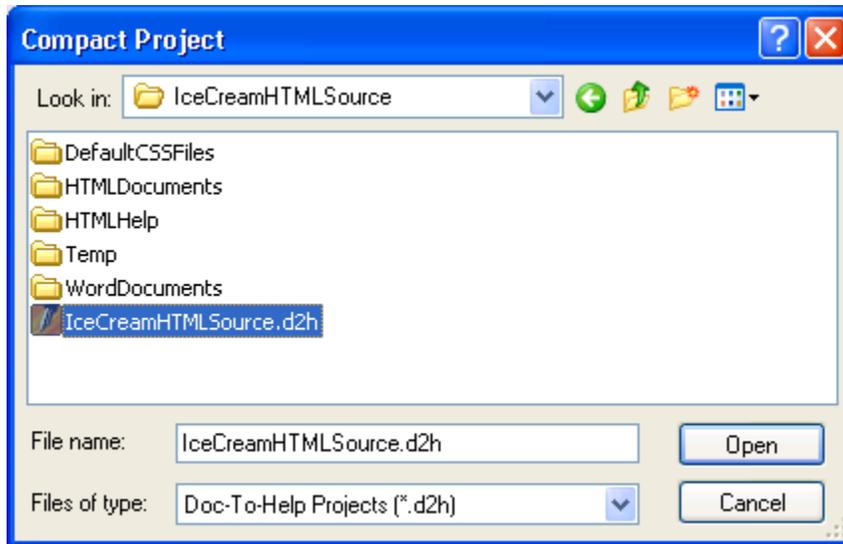
1. Click **Help | Check for Updates**. The ComponentOne Doc-To-Help Live Update dialog box appears with the latest version of Doc-To-Help.
2. Click the **Download** button to download the new version to your computer.

Compacting a Project File

Doc-To-Help includes a project compacting utility that can reduce the size of your project file. This utility checks the project file and, if it is fragmented, rearranges how the project file is stored on disk. When completed, the compacted project has reclaimed wasted space, and is usually smaller than the original. Doc-To-Help automatically checks and compacts your project, if necessary, when it is opened or any time a noticeable increase in size is detected.

To run the compacting utility manually:

1. Open the Doc-To-Help program. You are presented with the Doc-To-Help Start Page.
2. Click **Project | Compact Project**. The **Compact Project** dialog opens.



3. From the dialog, select the Doc-To-Help project you would like to compact and click **Open**.
4. The utility performs the compacting process automatically.

Note: The compact utility will not work if the Doc-To-Help project you wish to compact is open.

Project Settings Properties

By selecting the **Project Settings** item on the project editor tree list, you are presented with the Make and View Project commands. The lower right property pane contains the default project properties. These properties are generally set globally as defaults for the entire project. You can modify most properties by simply clicking in the box next to the property.

Document Default

The document properties are set globally for all documents, but can be overridden on a document-by-document basis in the document properties.

AdjustLeftIndent	Controls whether paragraph indentation is adjusted to account for wide margins when building online help.
KeepOutlineNumbers	When enabled, outline numbers are included as part of topic titles and are present in help targets.
KeepPageBreaks	When True, this property retains the page break characters in the source documents during compilation of a Printed Manual Help target. Set this property to False to discard page break characters.
SplitFile	Controls whether a document is split into separate topic files for HTML-based platforms.
SplitFileWithHHCtrl	When True, a single HTML file is generated using intrinsic HTML Help commands to give the appearance of separate HTML files for each topic in the document. When False, individual HTML files are generated for each topic in the document.
TextPopups	If True, generates a plain text only version of the Help file for context sensitive Help topics.

Localization

OnPage	Specifies the default text used when completing a cross-reference.
------------------------	--

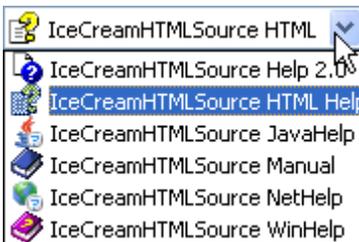
Miscellaneous Properties

DefaultTemplate	Specifies the document template that will be attached to new documents created in the Project Editor.
DocumentType	Specifies the types of source documents that can be included in the project. Possible values for this property are MS Word, HTML and Mixed.
Glossary	Specifies the document that contains glossary terms for the project.
HTMLDefaultCSS	Specifies the default cascading style sheet used in the project.
HTMLSourceFolder	Defines the folder name where the HTML source documents will reside.
MapNumberOffset	The value of this property is added to the automatically generated map numbers to prevent numbering conflicts in modular Help systems.
ModularHub	If True, the generated Help file can dynamically load the contents of other Help files, if present.
URLMode	Specifies the rules used for naming the .htm files generated for topics.
WordSourceFolder	Defines the folder name where the Word source documents will reside.

Changing the Current Help Target

A Doc-To-Help project always has a current Help target that specifies the kind of Help system to be built or viewed. To change the current Help target, choose the **Help Targets** item in the left pane of the project editor, then do either one of the following:

1. On the **Project** toolbar, select one of the available items in the **Help Target** list.



2. In the right pane of the project editor, select one of the available items in the Help target list.

Name	Type	Folder
(New)		
IceCreamHTMLSource Help 2.0	Microsoft Help 2.0	MSHelp
IceCreamHTMLSource HTML Help	HTML Help 1.x	HTMLHelp
IceCreamHTMLSource JavaHelp	JavaHelp	JavaHelp
IceCreamHTMLSource Manual	Printed Manual	Manual
IceCreamHTMLSource NetHelp	NetHelp	NetHelp
IceCreamHTMLSource WinHelp	WinHelp 4.0	Help

Building the Current Help Target

When you update the current Help target, Doc-To-Help selectively incorporates any substantive changes you have made to the project since it was last built, and performs only those actions that are absolutely necessary. When you rebuild the current Help target, Doc-To-Help empties its output directory and recompiles the entire project, even if you haven't changed anything.

To update the current Help target:

On the **Build** menu, click **Make Target**.

Alternatively, you can click the **Make Target** button on the **Project** toolbar.

To rebuild the current Help target in its entirety:

On the **Build** menu, click **Rebuild Target**.

Building Help Targets in Batch Mode

You may also choose to build a Help target in batch mode using a command line through a special executable program, **C1D2HBatch.exe**, installed with Doc-To-Help. The file is located in C:\Program Files\ComponentOne\DocToHelp, by default.

The commands that can be used correspond with the commands in the **Build** menu of the Doc-To-Help project editor and include:

Command	Description	Example
<code>c1d2hbatch -build project target flags</code>	Rebuilds the current Help target entirely.	<code>c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp" -p</code>
<code>c1d2hbatch -make project target flags</code>	Updates the current Help target.	<code>c1d2hbatch -make "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual"</code>
<code>c1d2hbatch -compileall project target flags</code>	Compiles all project files.	<code>c1d2hbatch -compileall "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp"</code>
<code>c1d2hbatch -compile project target document_name flags</code>	Compiles the specified document.	<code>c1d2hbatch -compile "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide HTML Help" -p</code>

Note: There is currently only one flag allowed in 'flags': -p, enabling the output of progress messages.

All error and log messages are output to the console standard output and have the following format:

- For errors aborting compilation:

```
D2H: fatal error: <error code>: <message text>
```

The <error code> is a numeric error code.

- For logged user errors, compilation continues and every log error is output as three lines in the order as shown in the following example. The three lines correspond to the three columns shown in the [Errors](#) (page 49) list in the project editor:

```
D2H: log error document: <document name>
D2H: log error topic: <topic title>
D2H: log error message: <severity>: <message text>
```

The <document name> and <topic title> can be empty, depending on the nature of the error.

The <severity> will be listed as one of the following three strings: "note", "warning", or "error".

- A log message, for information only, shows what document is currently compiling, what action is being performed, and so on:

```
D2H: log: <message text>
```

- A progress message appears only if the **-p** flag is present:

```
D2H: progress: <done>/<total >
```

Here <done> is an integer number indicating how many steps of the currently performed action have been completed so far.

The <total> is an integer number indicating the total number of steps in the currently performed action.

- A progress completion message indicates that the current action has been completed, and it appears only if the **-p** flag is present:

```
D2H: progress: done
```

- An unresolved links list is the output at the end of a successful build when the **-build** or **-make** commands are used. Every unresolved link is output as four lines in the order as shown in the following example. The four lines correspond to the four columns shown in the [Unresolved Links](#) (page 49) list in the project editor:

```
D2H: unresolved link text: <link text>
D2H: unresolved link style: <style name>
D2H: unresolved link topic: <topic title>
D2H: unresolved link document: <document name>
```

- For a successful build:

```
D2H: succeeded
```

- For a failed build:

```
D2H: failed
```

- The last message before Doc-To-Help batch mode exits:

```
D2H: exit
```

Viewing the Current Help Target

On the **Build** menu, click **View Target**.

Alternatively, you can click the **View Target** button on the **Project** toolbar.

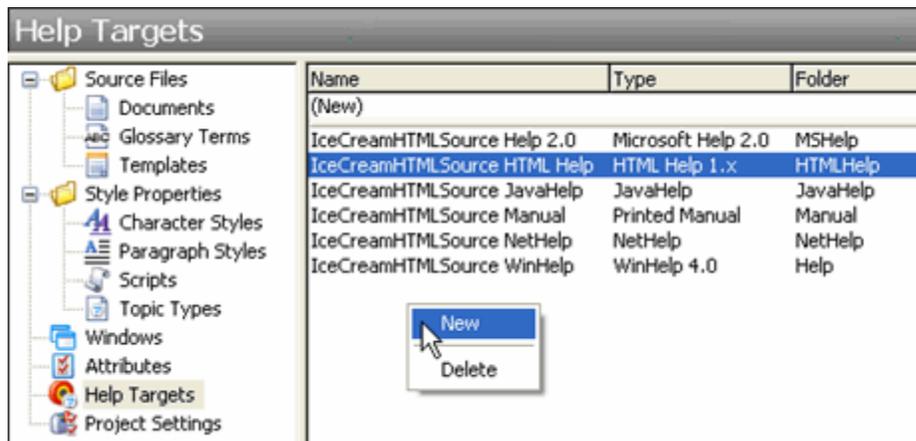
Note: If the current Help target has not been built or needs to be updated, you are asked if you want to build it. If you choose *Yes*, Doc-To-Help updates the Help target first by executing the **Make Target** command.

Adding a New Help Target

Doc-To-Help provides six default Help targets. In some situations, you may wish to have multiple Help targets of the same type, utilizing different file names. To add a new Help target to the default list:

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Right-click anywhere in the Help target list in the right pane and choose **New** from the shortcut menu.

Note that the cursor has moved to the top of the Help target list window.



4. Type the name of the new Help target.
5. Choose the type of Help target from the **Type** drop-down menu and press **Enter**.

Note that the new Help target has been added to the list. This new Help target contains the default properties of the **Type** you chose.

Alternatively, you can add a new Help target by clicking the **Project** menu, choosing **New**, then clicking **Help Target**.

Renaming a Help Target

When you create a new project, the default name for all help targets begins with the name of the project. For example, if you create a project named IceCream.d2h, the default name for the HTML Help target is IceCream HTML Help. For Help 2.0 it is IceCream Help 2.0, and so on. You can rename your help target at any time using the Help Target's [Name](#) property.

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Choose the help target for which you would like to change the name.
4. In the Help target property pane, locate the [Name](#) property. In the text box to the right of the property, highlight the existing name and change it as desired.

Changing the Help Target Directory

By default, Doc-To-Help stores the Help target files in an output directory whose name corresponds to the type of the Help target, or, in the case of a newly created Help target, the default output directory matches the name of the Help target. You can easily change the name of your output directory as needed:

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Choose the Help target you wish to edit from the right pane of the project editor.

Note that the property pane changes to reflect the Help target chosen.

4. In the Help target property pane, locate the [Folder](#) property. In the text box to the right of the property, highlight the existing name and change it as desired.

Help Target Properties

By selecting the **Help Targets** item on the project editor tree list, you are presented with a list of the currently defined Help targets in the right pane. By choosing any one of the Help targets, the lower left pane displays the property settings for that target.

Microsoft Help 2.0

This target produces the standard Microsoft Help 2.0.

Appearance Properties

BackColor	Displays the current background color setting for the Help pane.
BackgroundImage	Displays the image file to be used as a background in the Help pane.
BackgroundRepeat	Defines how the background image is arranged in the Help pane.
Caption	Represents the string to be displayed in the caption bar of the help window.
Theme	Determines the appearance of the Help pane and framesets for HTML-based targets.

Build Properties

AttributeExpression	Specifies an advanced conditional build expression for a help target.
GenerateCSS	When set to False, the existing cascading stylesheet will not be overwritten when building the Help target.
OwnTOC	Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties

ButtonLabel	Specifies the text that precedes the subtopic buttons.
ButtonLabelStyle	Lists the name of the style used to format the label that precedes the subtopic buttons.
ButtonSeparator	When enabled, a dividing line separates the topic text from the subtopic buttons.
ButtonStyle	Lists the name of the style used to format automatically generated subtopic buttons.
ColorReduction	Determines the color threshold for images that are saved with source documents.

Value	Description
None	Images are not converted.
16 Colors	Images with more than 16 colors are reduced to 16.
256 Colors	Images with more than 256 colors are reduced to 256.

GraphicsScaleWithFonts	When set to True, all graphics in Word source documents are resized proportionally when system font size is changed.
ImageFormat	Determines graphic format in which images originated from Word source documents are stored in image files in the online help target.
ScaleGraphics	Determines what scaling is applied to the graphics in Word source document.

Localization Properties

Charset	Specifies one of the character set aliases listed in HTML Character Sets and Named Entities.
CodePage	Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.
Locale	This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hlp).

Miscellaneous Properties

BaseName	Base file name to be used for generating Help target files. By default, this name will be the same as the file name for the Help project itself.
CopyFolder	Defines the folder name where ancillary build files will reside.
Folder	Defines the folder name where the Help files will reside.
Name	Returns a string containing the name of a specified Help target (read-only).
StyleSheet	Specifies the name of the cascading style sheet applied to intermediate files during compilation.
Template	The document template to apply to the intermediate files during compilation.

Strings Properties

Namespace	Specifies the Help 2.0 namespace that is used to register the Help file after a successful build.
ParentNamespace	Specifies the namespace for the built help collection to plug into. Usually, that will be the namespace of the Visual Studio .NET Combined Collection. Set this property if you want the built Help to be automatically registered as a plug-in for Visual Studio help. For example, for Visual Studio .NET 2003, set ParentNamespace to "MS.VSCC.2003".

NETHelp

This target produces browser neutral HTML in accordance with version 4.0 of the HTML specification published by W3C.

Appearance Properties

BackgroundColor	Displays the current background color setting for the Help pane.
BackgroundImage	Displays the image file to be used as a background in the Help pane.
BackgroundRepeat	Defines how the background image is arranged in the Help pane.
Caption	Represents the string to be displayed in the caption bar of the help window.
Theme	Determines the appearance of the Help pane and framesets for HTML-based targets.

Build Properties

AttributeExpression	Specifies an advanced conditional build expression for a help target.
GenerateCSS	When set to False, the existing cascading stylesheet will not be overwritten when building the Help target.
OwnTOC	Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties

ButtonLabel	Specifies the text that precedes the subtopic buttons.								
ButtonLabelStyle	Lists the name of the style used to format the label that precedes the subtopic buttons.								
ButtonSeparator	When enabled, a dividing line separates the topic text from the subtopic buttons.								
ButtonStyle	Lists the name of the style used to format automatically generated subtopic buttons.								
ColorReduction	Determines the color threshold for images that are saved with source documents. <table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>None</td><td>Images are not converted.</td></tr><tr><td>16 Colors</td><td>Images with more than 16 colors are reduced to 16.</td></tr><tr><td>256 Colors</td><td>Images with more than 256 colors are reduced to 256.</td></tr></tbody></table>	Value	Description	None	Images are not converted.	16 Colors	Images with more than 16 colors are reduced to 16.	256 Colors	Images with more than 256 colors are reduced to 256.
Value	Description								
None	Images are not converted.								
16 Colors	Images with more than 16 colors are reduced to 16.								
256 Colors	Images with more than 256 colors are reduced to 256.								
GraphicsScaleWithFonts	When set to True, all graphics in Word source documents are resized proportionally when system font size is changed.								
ImageFormat	Determines graphic format in which images originated from Word source documents are stored in image files in the online help target.								
ScaleGraphics	When set to False, graphics contained with the source documents will not be scaled.								

HTML Properties

DefaultFile	File name to be used as part of the base URL for the HTML target. Typically, the filename will be <i>default.htm</i> or <i>index.html</i> .
DynamicTOC	Generates a Dynamic HTML version of the contents page with an expandable outline.
Frameset	Generates a frameset version of the HTML Help project with the left frame displaying the contents and the right frame displaying the Help topics.

Localization Properties

Charset	Specifies one of the character set aliases listed in HTML Character Sets and Named Entities.
CodePage	Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.
Locale	This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hhp).

Miscellaneous Properties

BaseName	Base file name to be used for generating HTML target files. By default, this name will be the same as the file name for the Help project itself.
CopyFolder	Defines the folder name where ancillary build files will reside.
Folder	Defines the folder name where the Help files will reside.
MapNumbers	Determines whether a unique Context ID (map number) is generated for each topic.
Name	Returns a string containing the name of a specified Help target (read-only).
SearchStopList	Specifies the path to a text file containing noise words.

[StyleSheet](#) Specifies the name of the cascading style sheet applied to intermediate files during compilation.

[Template](#) The document template to apply to the intermediate files during compilation.

Natural Search Properties

[AnswerCount](#) Determines the maximum number of answers to be returned when a natural language search is performed.

[ConfidenceLevel](#) Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed.

[SearchEnabled](#) Controls whether ComponentOne Natural Search is enabled.

[StemPercent](#) Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.

[StemSize](#) Determines the maximum number of characters used as a root word when building natural language search indexes.

[StringNoMatch](#) Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Strings Properties

[StringContents](#) Allows modification of the **contents** string value.

[StringFoundZero](#) Allows modification of the **No topics found.** string value.

[StringIndex](#) Allows modification of the **index** string value.

[StringIndexCaption](#) Represents the string to be displayed in the Index window above the index text box.

[StringNext](#) Allows modification of the **next** string value.

[StringPrevious](#) Allows modification of the **previous** string value.

[StringSearch](#) Allows modification of the **Search** string value.

[StringSearchCaption](#) Represents the string to be displayed in the Search window above the search text box.

[StringSearchGo](#) Represents the string to be displayed in the search button within the Search window.

[StringSyncTOC](#) Allows modification of the **Sync TOC** string value.

Tabs Properties

[ContentsTab](#) When enabled, the Contents tab is added to the help window.

[IndexTab](#) When enabled, the Index tab is added to the help window.

[SearchTab](#) When enabled, the Search tab is added to the help window.

HTML Help

This target produces Microsoft HTML Help, which requires Internet Explorer to be installed on the client machine. Internet Explorer version 3.0 is acceptable, but 4.0 or greater is preferred.

Appearance Properties

[BackgroundColor](#) Displays the current background color setting for the Help pane.

[BackgroundImage](#) Displays the image file to be used as a background in the Help pane.

[BackGroundRepeat](#)

Defines how the background image is arranged in the Help pane.

[Theme](#)

Determines the appearance of the Help pane and framesets for HTML-based targets.

Build Properties[AttributeExpression](#)

Specifies an advanced conditional build expression for a help target.

[BinaryIndex](#)

When True, generates a binary index instead of a site map index.

[BinaryTOC](#)

When True, generates a binary table of contents instead of a site map table of contents.

[GenerateCSS](#)

When set to False, the existing cascading stylesheet will not be overwritten when building the Help target.

[GenerateProjectFile](#)

When set to False, the existing .hpi or .hpl files will not be overwritten when building the Help target.

[OwnTOC](#)

Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties[ButtonLabel](#)

Specifies the text that precedes the subtopic buttons.

[ButtonLabelStyle](#)

Lists the name of the style used to format the label that precedes the subtopic buttons.

[ButtonSeparator](#)

When enabled, a dividing line separates the topic text from the subtopic buttons.

[ButtonStyle](#)

Lists the name of the style used to format automatically generated subtopic buttons.

[ColorReduction](#)

Determines the color threshold for images that are saved with source documents.

Value	Description
None	Images are not converted.
16 Colors	Images with more than 16 colors are reduced to 16.
256 Colors	Images with more than 256 colors are reduced to 256.

[DefaultWindow](#)

Specifies the default window definition for the compiled help file.

[GraphicsScaleWithFonts](#)

When set to True, all graphics in Word source documents are resized proportionally when system font size is changed.

[ImageFormat](#)

Determines graphic format in which images originated from Word source documents are stored in image files in the online help target.

[ScaleGraphics](#)

When set to False, graphics contained with the source documents will not be scaled.

[TopicsFound](#)

Determines whether Alink/Klink topic lists are displayed in a dialog box (default) or a pop-up menu.

Localization Properties[Charset](#)

Specifies one of the character set aliases listed in HTML Character Sets and Named Entities.

[CodePage](#)

Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.

[Locale](#) This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hhp).

Miscellaneous Properties

[BaseName](#) Base file name to be used for generating HTML Help target files. By default, this name will be the same as the file name for the Help project itself.

[CopyFolder](#) Defines the folder name where ancillary build files will reside.

[Folder](#) Defines the folder name where the Help files will reside.

[MapFileForVB](#) Creates Visual Basic header files for context sensitive Help.

[MapNumbers](#) Determines whether a unique Context ID (map number) is generated for each topic.

[Name](#) Returns a string containing the name of a specified Help target (read-only).

[SkipGlossary](#) When True, omits the Glossary topic from the generated Help contents.

[StyleSheet](#) Specifies the name of the cascading style sheet applied to intermediate files during compilation.

[Template](#) The document template to apply to the intermediate files during compilation.

Natural Search Properties

[AnswerCount](#) Determines the maximum number of answers to be returned when a natural language search is performed.

[ConfidenceLevel](#) Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed.

[SearchCaption](#) Specifies the text added to the tab caption (HTML Help) or button caption (WinHelp) when ComponentOne Natural Search is enabled.

[SearchEnabled](#) Controls whether ComponentOne Natural Search is enabled.

[StemPercent](#) Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.

[StemSize](#) Determines the maximum number of characters used as a root word when building natural language search indexes.

[StringNoMatch](#) Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Strings Properties

[StringNext](#) Allows modification of the **next** string value.

[StringPrevious](#) Allows modification of the **previous** string value.

Printed Manual

This target produces a Microsoft Word document that includes a table of contents, an index and the individual subdocuments generated from your source documents.

Build Properties

[AttributeExpression](#) Specifies an advanced conditional build expression for a help target.

[Concatenate](#) When True, subdocuments are concatenated into a single master document in printed manual builds.

[FixLists](#) Controls how numbered lists are corrected in the printed manual output when the Concatenate property is set to True.

Value	Description
Never	Never corrects the numbered lists.
Always	Always corrects the numbered lists.
Word 2000 Only (Default)	Corrects lists with Word 2000, but not with Word 2002 (XP).

[OwnTOC](#) Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties

[LiveLinks](#) Determines whether links are live hyperlinks in the manual Word document.

[ShowDropdownText](#) If set to True, drop-down text is shown in the help target, otherwise they are removed.

[ShowExpandText](#) If set to True, drop-down text is shown in the help target, otherwise they are removed.

Localization Properties

[CodePage](#) Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.

[Locale](#) This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hhp).

Miscellaneous Properties

[BaseName](#) Base file name to be used for generating the target file. By default, this name will be the same as the file name for the Help project itself.

[CopyFolder](#) Defines the folder name where ancillary build files will reside.

[Folder](#) Defines the folder name where the Help files will reside.

[Name](#) Returns a string containing the name of a specified Help target (read-only).

[StyleSheet](#) Specifies the name of the cascading style sheet applied to intermediate files during compilation.

[Template](#) The document template to apply to the intermediate files during compilation.

[UseTemplateText](#) Determines whether the text of the template associated with the Manual help target is included in the resulting document.

Strings Properties

[StringContents](#) Allows modification of the **contents** string value.

[StringIndex](#) Allows modification of the **index** string value.

Title Page Properties

ByLine	Specifies the byline text used on the title page of printed manuals.
SuperTitle	Specifies the text that appears above the title string on the title page of printed manuals.
Title	Represents the title string of an object.

WinHelp 4.0

This target produces the standard 32-bit Windows Help 4.0.

Build

AttributeExpression	Specifies an advanced conditional build expression for a help target.
GenerateProjectFile	When set to False, the existing .hpl or .hpl files will not be overwritten when building the Help target.
OwnTOC	Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties

ButtonLabel	Specifies the text that precedes the subtopic buttons.
ButtonLabelStyle	Lists the name of the style used to format the label that precedes the subtopic buttons.
ButtonSeparator	When enabled, a dividing line separates the topic text from the subtopic buttons.
ButtonStyle	Lists the name of the style used to format automatically generated subtopic buttons.
ColorReduction	Determines the color threshold for images that are saved with source documents. This property has no effect on linked images that are not saved with the document.

Value	Description
None	Images are not converted.
16 Colors	Images with more than 16 colors are reduced to 16.
256 Colors	Images with more than 256 colors are reduced to 256.

ScaleGraphics	When set to False, graphics contained with the source documents will not be scaled.
ShowDropdownText	If set to True, drop-down text is shown in the help target, otherwise they are removed.
ShowExpandText	If set to True, drop-down text is shown in the help target, otherwise they are removed.

Localization Properties

CodePage	Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.
Locale	This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hpl).

Miscellaneous Properties

BaseName	Base file name to be used for generating WinHelp target files. By default, this name will be the same as the file name for the Help project itself.																
Citation	Appends a citation to the end of any information (except in a context-sensitive pop-up window) that is copied from the Help file.																
Compression	Determines the level of file compression used by the WinHelp compiler. <table><thead><tr><th>Value</th><th>Description</th></tr></thead><tbody><tr><td>None</td><td>No compression.</td></tr><tr><td>Chose best</td><td>The WinHelp compiler determines the best algorithm to use.</td></tr><tr><td>Phrase</td><td>For Help files under 100K.</td></tr><tr><td>Hall</td><td>For Help files that will be compressed again by another utility.</td></tr><tr><td>Zeck</td><td>Minimal compression for quick builds.</td></tr><tr><td>Zeck + Phrase</td><td>Minimal compression for Help files under 100K.</td></tr><tr><td>Zeck + Hall</td><td>Maximum compression.</td></tr></tbody></table>	Value	Description	None	No compression.	Chose best	The WinHelp compiler determines the best algorithm to use.	Phrase	For Help files under 100K.	Hall	For Help files that will be compressed again by another utility.	Zeck	Minimal compression for quick builds.	Zeck + Phrase	Minimal compression for Help files under 100K.	Zeck + Hall	Maximum compression.
Value	Description																
None	No compression.																
Chose best	The WinHelp compiler determines the best algorithm to use.																
Phrase	For Help files under 100K.																
Hall	For Help files that will be compressed again by another utility.																
Zeck	Minimal compression for quick builds.																
Zeck + Phrase	Minimal compression for Help files under 100K.																
Zeck + Hall	Maximum compression.																
CopyFolder	Defines the folder name where ancillary build files will reside.																
Copyright	Places a custom copyright notice in the Version dialog box of WinHelp.																
Folder	Defines the folder name where the Help files will reside.																
HubContents	Displays the name of the contents (.cnt) file to associate with the Help file.																
MapFileForC	Creates C or C++ header files for context sensitive Help.																
MapFileForVB	Creates Visual Basic header files for context sensitive Help.																
MapNumbers	Determines whether a unique Context ID (map number) is generated for each topic.																
Name	Returns a string containing the name of a specified Help target (read-only).																
SkipGlossary	When True, omits the Glossary topic from the generated Help contents.																
StyleSheet	Specifies the name of the cascading style sheet applied to intermediate files during compilation.																
Template	The document template to apply to the intermediate files during compilation.																
Title	Represents the title string of an object.																
WinHelpMacro	Specifies a macro to run when a Help target, a topic, or secondary window is opened.																

Natural Search Properties

AnswerCount	Determines the maximum number of answers to be returned when a natural language search is performed.
ConfidenceLevel	Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed.
SearchCaption	Specifies the text added to the tab caption (HTML Help) or button caption (WinHelp) when ComponentOne Natural Search is enabled.
SearchEnabled	Controls whether ComponentOne Natural Search is enabled.

StemPercent	Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.
StemSize	Determines the maximum number of characters used as a root word when building natural language search indexes.
StringNoMatch	Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Navigation Properties

OneBrowseSequence	When True, this property provides a continuous browse sequence that spans multiple source documents.
-----------------------------------	--

JavaHelp

This target produces Java Help 1.1.3. For more information, see [JavaHelp Target](#) (page 30).

Appearance Properties

BackgroundColor	Displays the current background color setting for the Help pane.
BackgroundImage	Displays the image file to be used as a background in the Help pane.
BackgroundRepeat	Defines how the background image is arranged in the Help pane.
Caption	Represents the string to be displayed in the caption bar of the help window.
Theme	Determines the appearance of the Help pane and framesets for HTML-based targets.

Build Properties

AttributeExpression	Specifies an advanced conditional build expression for a help target.
GenerateCSS	When set to False, the existing cascading stylesheet will not be overwritten when building the Help target.
OwnTOC	Determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well.

Display Properties

ButtonLabel	Specifies the text that precedes the subtopic buttons.
ButtonLabelStyle	Lists the name of the style used to format the label that precedes the subtopic buttons.
ButtonSeparator	When enabled, a dividing line separates the topic text from the subtopic buttons.
ButtonStyle	Lists the name of the style used to format automatically generated subtopic buttons.
ColorReduction	Determines the color threshold for images that are saved with source documents.

Value	Description
None	Images are not converted.
16 Colors	Images with more than 16 colors are reduced to 16.
256 Colors	Images with more than 256 colors are reduced to 256.

ImageFormat	Determines graphic format in which images originated from Word source documents are stored in image files in the online help target.
-----------------------------	--

[ScaleGraphics](#) When set to False, graphics contained with the source documents will not be scaled.

[ShowDropdownText](#) If set to True, drop-down text is shown in the help target, otherwise they are removed.

[ShowExpandText](#) If set to True, drop-down text is shown in the help target, otherwise they are removed.

Fonts Properties

[IgnoreFontNames](#) Determines whether font names specified in the source document are ignored.

[IgnoreFontSizes](#) Determines whether font sizes specified in the source document are ignored.

[ScaleFontSizes](#) Determines whether all font sizes are scaled in the help target

Localization Properties

[Charset](#) Specifies one of the character set aliases listed in HTML Character Sets and Named Entities.

[CodePage](#) Defines the windows codepage number to be used by the Help file. This property is most commonly used when producing Help files in foreign languages, providing special character sets for translation on the screen or in print.

[Locale](#) This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hhp).

Miscellaneous Properties

[BaseName](#) Base file name to be used for generating HTML Help target files. By default, this name will be the same as the file name for the Help project itself.

[CopyFolder](#) Defines the folder name where ancillary build files will reside.

[Folder](#) Defines the folder name where the Help files will reside.

[Name](#) Returns a string containing the name of a specified Help target (read-only).

[StyleSheet](#) Specifies the name of the cascading style sheet applied to intermediate files during compilation.

[Template](#) The document template to apply to the intermediate files during compilation.

Strings Properties

[PrefixIndexListFile](#) Allows modification of the default prefix for topic list files corresponding to the main index.

[PrefixTopicListFile](#) Allows modification of the default prefix for topic list files corresponding to keyword hot spots.

[StringContents](#) Allows modification of the **contents** string value.

[StringFavorites](#) Allows modification to the Favorites string value.

[StringFoundMany](#) Allows modification of the **%d topics found:** string value.

[StringFoundOne](#) Allows modification of the **1 topic found:** string value.

[StringFoundZero](#) Allows modification of the **No topics found.** string value.

[StringIndex](#) Allows modification of the **index** string value.

[StringNext](#) Allows modification of the **next** string value.

[StringPrevious](#)

Allows modification of the **previous** string value.

[StringReturnToIndex](#)

Allows modification of the **return to index** string value.

[StringSearch](#)

Allows modification of the **Search** string value.

Tabs Properties

[ContentsTab](#)

When enabled, the **Contents** tab is added to the help window.

[FavoritesTab](#)

When enabled, the **Favorites** tab is added to the help window.

[IndexTab](#)

When enabled, the **Index** tab is added to the help window.

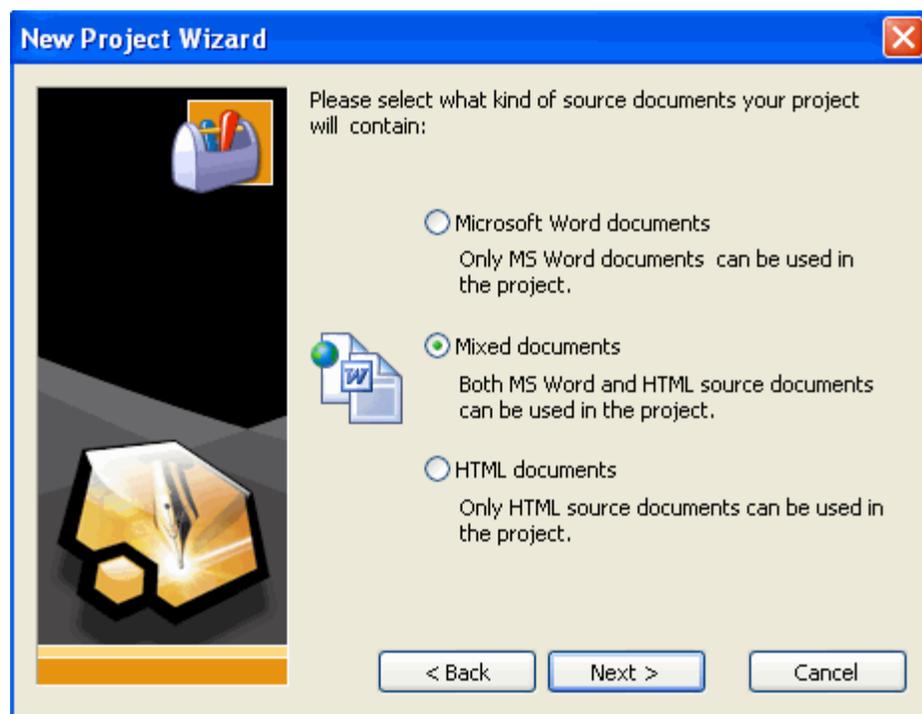
[SearchTab](#)

When enabled, the **Search** tab is added to the help window.

Doc-To-Help 2006 Source Documents

Doc-To-Help Enterprise 2006 supports two types of source documents: Word and HTML. In addition to allowing you to build help targets from Word documents, Doc-To-Help gives you the option of building a help target directly from HTML documents, without even using Word, or you can add both Word and HTML documents to the same project.

When you create a new project, you can select which type of documents you want to use.



Note: Doc-To-Help Enterprise 2006 supports HTML source documents, cascading style sheets and FrontPage and Dreamweaver integration. Doc-To-Help for Word 2006 does not support these features.

Project Types

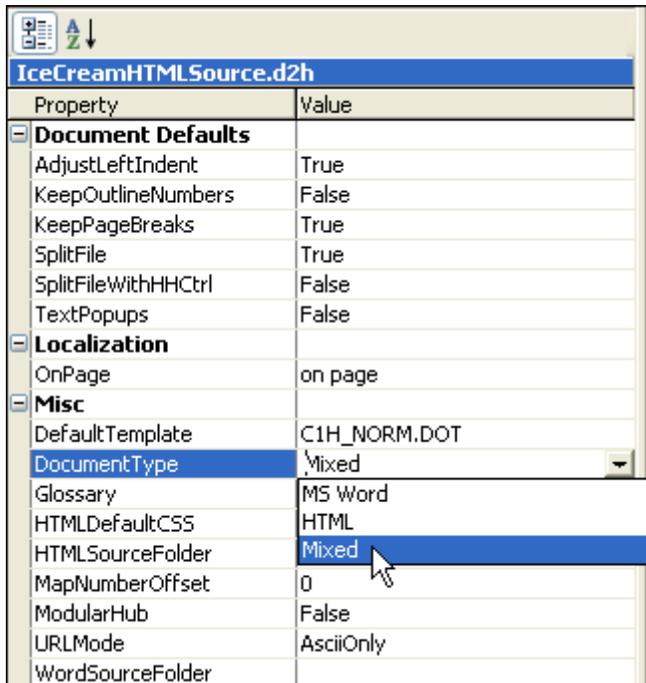
Depending on the option you choose when specifying the source documents for your project, Doc-To-Help sets the Project.[DocumentType](#) property to either MS Word, Mixed or HTML:

- *MS Word* allows your project to contain Word documents only.
- *Mixed* allows your project to contain both Word and HTML documents.
- *HTML* allows your project to contain HTML documents only.

You can change the type of documents used in your project at any time, if necessary, provided there are no “foreign”, or non-Word /non-HTML, documents in the project.

To change the Project Type:

1. In your Doc-To-Help project, select the **Project Settings** item on the project editor tree list. The lower right property pane contains the project properties.
2. Select the drop-down arrow next to the [DocumentType](#) property under the Miscellaneous properties and choose one of the options: MS Word, HTML or Mixed.

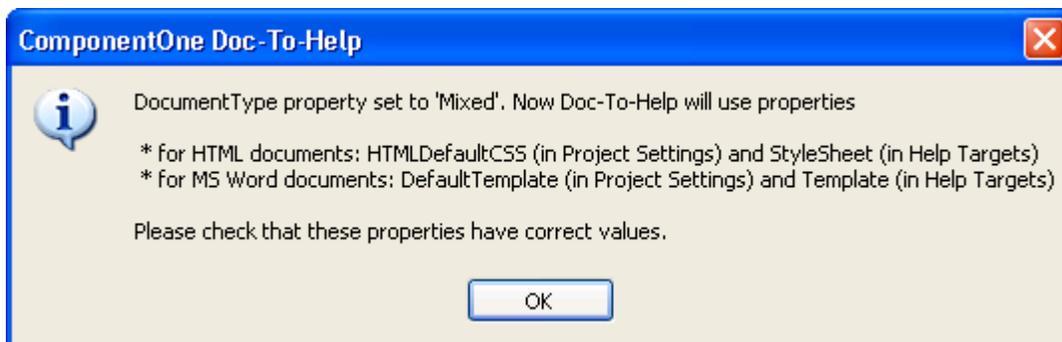


- If you select *HTML* for a project containing Word documents only, or if you select *MS Word* for a project containing HTML documents only, Doc-To-Help displays the following message:



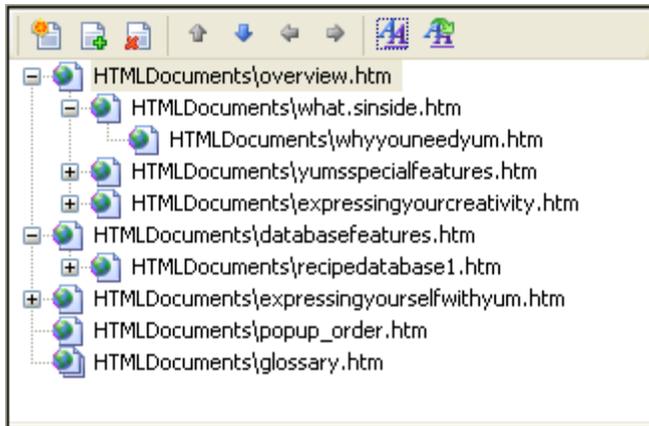
The documents within the project must be of the same type as what is specified in the [DocumentType](#) property (unless *Mixed* is specified, in which case, Word and HTML documents can be used). Remove either the Word documents or HTML documents, and then set the [DocumentType](#) property.

- If you select *Mixed*, Doc-To-Help notifies you that the project will use properties for both HTML documents and Word documents:



Document Tree

Unlike the previous Doc-To-Help versions where documents formed a plane list, Doc-To-Help 2006 uses a document tree to display the hierarchy of source documents. The tree allows you to stay organized and synchronize the source document hierarchy with the table of contents hierarchy.



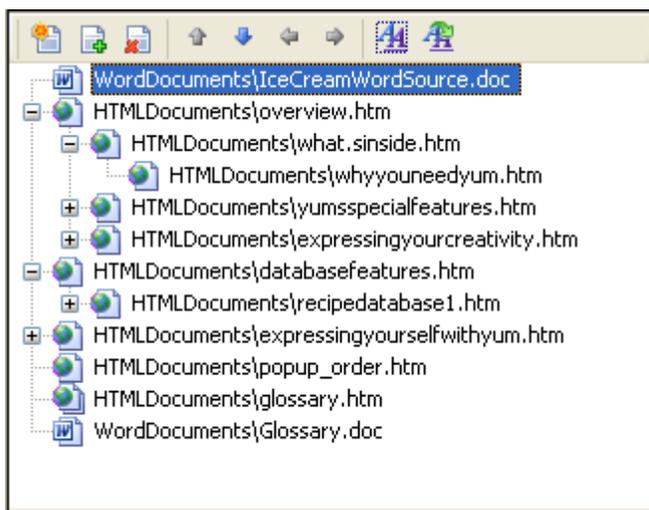
You determine the position of a document in the document tree when you add or create a new document in the project. The position can be changed later by dragging-and-dropping the document to a new position or by using the Move Up, Move Down, Move Left, and Move Right buttons on the **Documents** toolbar.

Single and Multiple Topic Documents

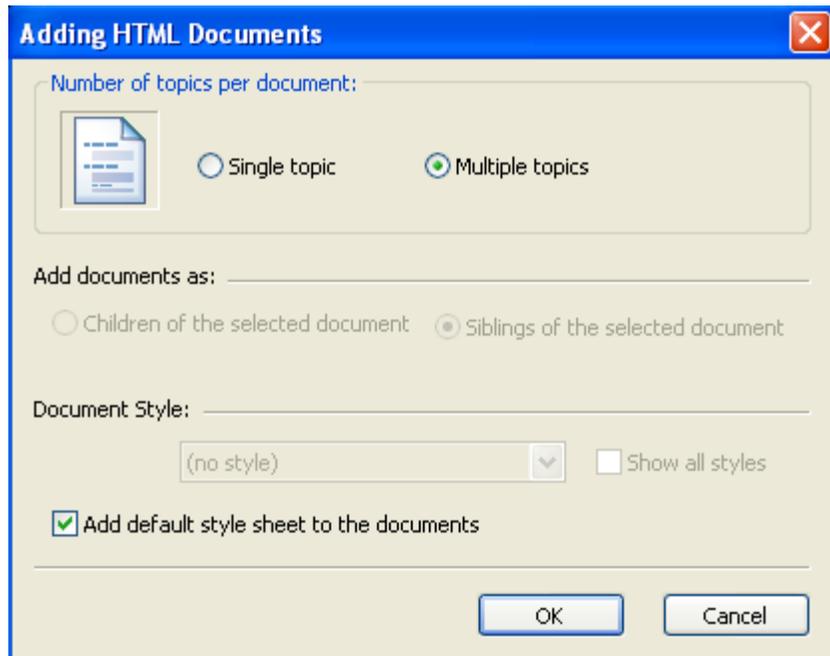
There are two kinds of source documents available, *single topic* documents and *multiple topic* documents:

- **Single topic documents** allow you to have only one topic per document. Only single-topic documents participate in the document tree hierarchy.
- **Multiple topic documents** allow you to have as many topics as you would like. Word documents are always multiple topic documents, so there is no option for single topic Word documents. Although multiple topic documents are in the same tree as single topic documents, they always appear on the first level. Only single topic documents can be placed on deeper levels.

In this sample document tree, you will notice the single topic HTML source documents are placed on multiple levels, while the multiple topic Word documents only appear on the first level:



Unlike Word documents, HTML documents can be either single or multiple topic documents. You can specify which type you would like to use by selecting either the *Single topic* or *Multiple topics* radio button when you add or create a new HTML document in your project.



A multiple topic HTML document generates topics in the same way as Word documents generate topics. Topics are defined by their headings in the source document. Headings are formatted with paragraph styles that have the `Style.Type` property set to Level 1 through 9 in the Doc-To-Help project. The style level determines the level of the topic in the automatically generated table of contents (TOC). For example, the `Style.Type` property of the Heading 1 paragraph style is set to Level 1 in Doc-To-Help. If you apply the Heading 1 style to a topic in your source document, Doc-To-Help places the topic at Level 1 in the TOC.

A single topic HTML document directly corresponds to a topic in a help target. For a single topic document, the user must specify the `Title` and `Style` of the topic, which can be done in the **Adding HTML Documents** or **New HTML Document** dialog box. This tells Doc-To-Help the title and position of the topic in the TOC hierarchy. The `Style` property directly corresponds with the paragraph styles in the project. If the `Style` property of a single topic document is set to *Heading 1*, the topic assumes all of the properties of the *Heading 1* paragraph style. The document `Style` also corresponds to a level in the tree. This help you keep the document tree hierarchy synchronized with the TOC hierarchy. If the style does not correspond with the document's level in the tree when the document is added, Doc-To-Help gives you a warning and allows you to select another style. You are not required to match the document tree level and the `Style` level; this does not prevent the target from building correctly.



For your convenience, the **Select Documents With Incorrect Style** button  on the **Documents** toolbar can be used to check the entire tree for documents that do not have the same tree and TOC levels.

The document tree hierarchy usually follows the TOC hierarchy of the topics. The tree hierarchy does not necessarily follow the source documents directory structure, although you may find it convenient to store source documents in a directory hierarchy parallel to the document tree hierarchy.

Other than helping the user to organize single-topic documents in the same hierarchy as in the TOC, the document tree hierarchy does not play an essential role in creating help targets. The help target's TOC is created based solely on the Title and Style properties of the documents for single-topic documents and on the style formatting of topic headers in multiple topic documents.

Word Source Documents

One of the keys to creating well defined Help documentation from Word source documents with ComponentOne Doc-To-Help 2006 is a working knowledge of Microsoft Word. In this section we will open a Word source document and review some of the important settings in Word.

Opening a Word Document

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click the document you wish to open and choose **Open** from the shortcut menu.

The document you have chosen opens within Microsoft Word.

Alternatively, you may choose to use Doc-To-Help's "in-place" editing feature to edit the document within the Doc-To-Help project editor. You are still editing the document with Word, but it appears in the project editor for your convenience.

To open and edit a Word document in the Doc-To-Help project editor:

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click the document you wish to open and choose **Edit** from the shortcut menu.

The document you have chosen opens within the right pane of the project editor.

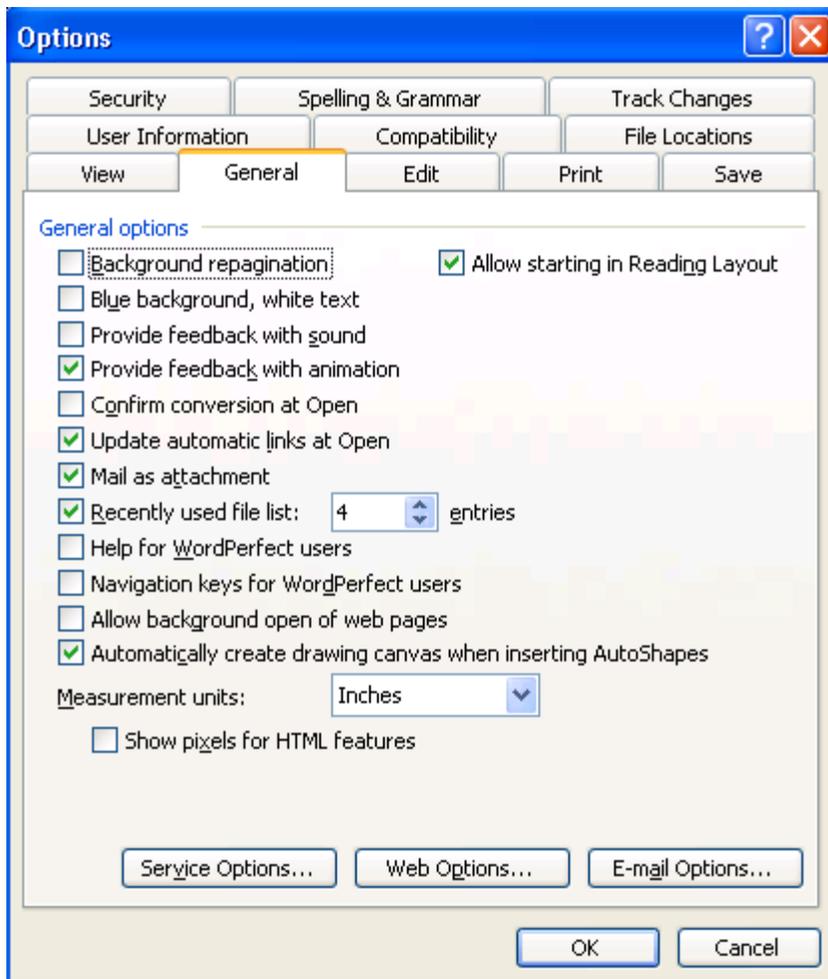
Note: Some anti-virus software options can limit the functionality of Doc-To-Help's "in place" editing features. For a list of known issues, contact [Technical Support](#) (page 12).

To close an in-place document within Doc-To-Help:

On the **File** menu, click **Close and Return to Doc-To-Help**.

The Options Dialog Box

Most of the Word options and settings that you should check are found in the **Options** dialog box. Open this dialog box by clicking the **Tools** menu, and selecting **Options**. The **Options** dialog box contains a series of tabs that control the default options for View, Print, Edit, Save and many other settings. The **General Tab** is shown below.



General Tab

Please ensure that the Help For WordPerfect Users and Navigation Keys For WordPerfect Users checkboxes are clear (not selected). They interfere with many Word command keystrokes and can also make it difficult to properly apply styles. Background Repagination should also be clear because repagination can slow down Word's performance while editing large documents.

View Tab

Please select the Screen Tips option to visibly highlight text areas that have been modified by the Doc-To-Help link features.

Save Tab

Please select the **Save** tab and make sure that the following check box is clear (not selected):

Allow Fast Saves

Microsoft added this option in order to make it possible to save your documents quickly. Fast Save accomplishes this by adding your latest changes to the end of the .DOC file and referencing them at the appropriate location. When you decide to delete text or graphics, the reference is dropped, but the material is left in the file. In large documents, this means that the size of your file can keep increasing even if you delete material. Many Word users have reported that file corruption occurs much more easily when this feature is selected.

Word and Infrequently Used Menu Items

To streamline the appearance of your menus, Word 2000 displays only the most frequently used menu items, by default. If you're using Doc-To-Help and Word 2000, you will find that certain menu items referred to in this chapter are not immediately available. Please follow the two steps below to make additional menu items available to you.

Note: The streamlining discussed above also extends to the display of toolbar icons. If certain toolbar icons do not appear at first, click the More Buttons chevron (>>) on the toolbar to display hidden, yet available icons.

Modifying Word Default Settings to Display All Menu Items

1. Open the **Tools** menu.
2. Choose **Customize**. The **Customize** dialog box opens.
3. Clear the **Menus show recently used command first** checkbox under the Options tab, then click **Close**.

Note: Should you choose not to disable this Word 2000 feature, you'll find it is still possible to display infrequently used menu items by allowing your cursor to "hover" over the menu list.

HTML Source Documents

Doc-To-Help does not limit you when editing your source documents; you can use the HTML editor of your choice. We recommend using Microsoft FrontPage or Macromedia Dreamweaver so you can take advantage of the **D2HML Styles** toolbar and dialog boxes, which simplify the formatting process, but it is not required. See [Doc-To-Help Markup in Source Documents](#) (page 197) for more information.

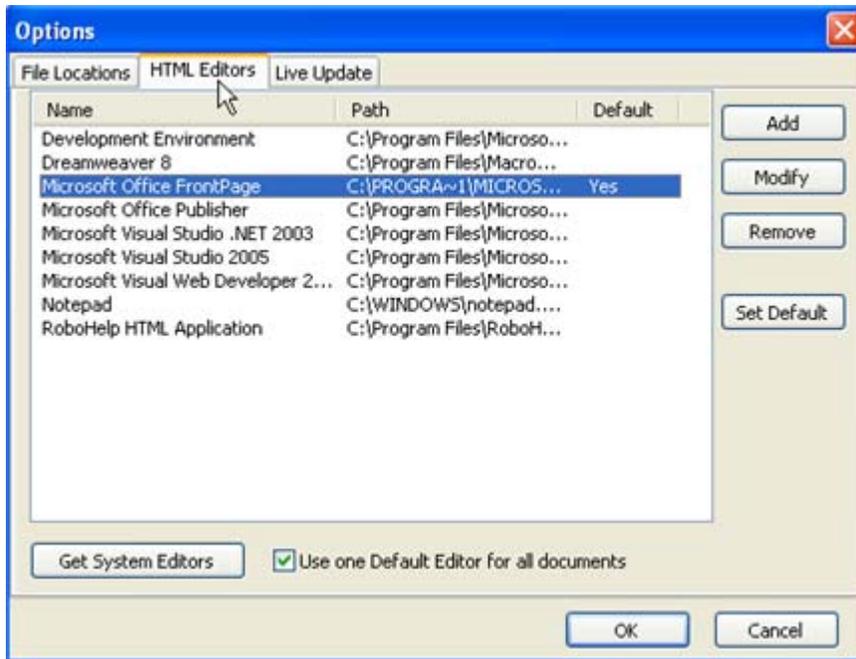
Editing HTML Source Documents

When you open an HTML document for editing the first time after installing D2H, you are shown the list of HTML editors available on your machine. You may modify this list, if necessary, by choosing the default HTML editor, adding editors if they are not automatically detected, and removing the editors you do not want to use. You can access this list at any time by selecting **Tools | Options** in the Doc-To-Help Project Editor and clicking the **HTML Editors** tab.

Specifying a Default HTML Editor

You can also access the list of HTML editors from the Doc-To-Help Project Editor.

1. Select **Tools | Options** to open the **Options** dialog box.
2. Select the **HTML Editors** tab. A list of available HTML editors appears.

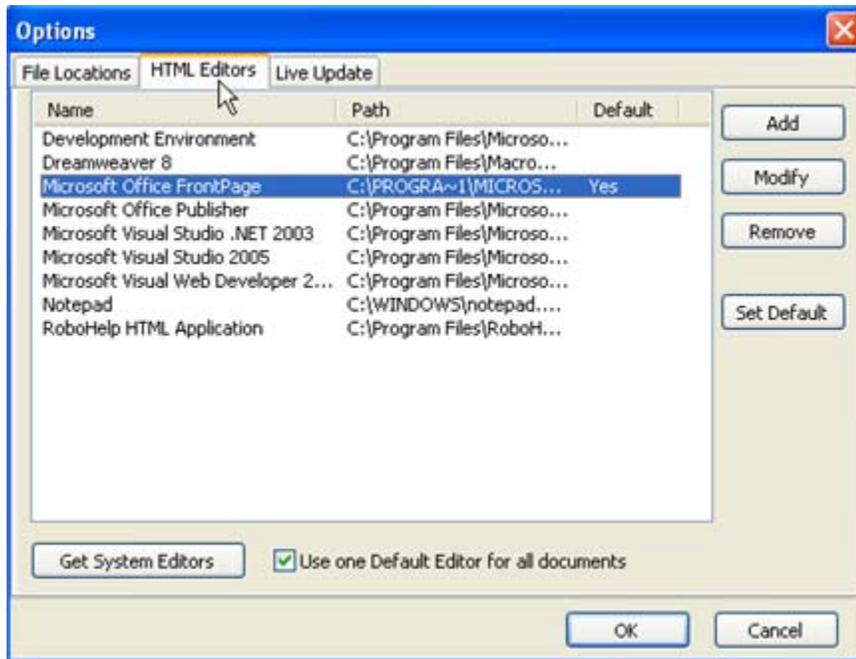


3. Choose an editor and make sure the **Use one Default Editor for all documents** checkbox is selected.
4. Click the **Set Default** button so that the HTML source documents are automatically opened with this editor. If your editor does not appear in the list, click the **Get System Editors** button or add it to the list using the **Add** button.
5. Click **OK**.

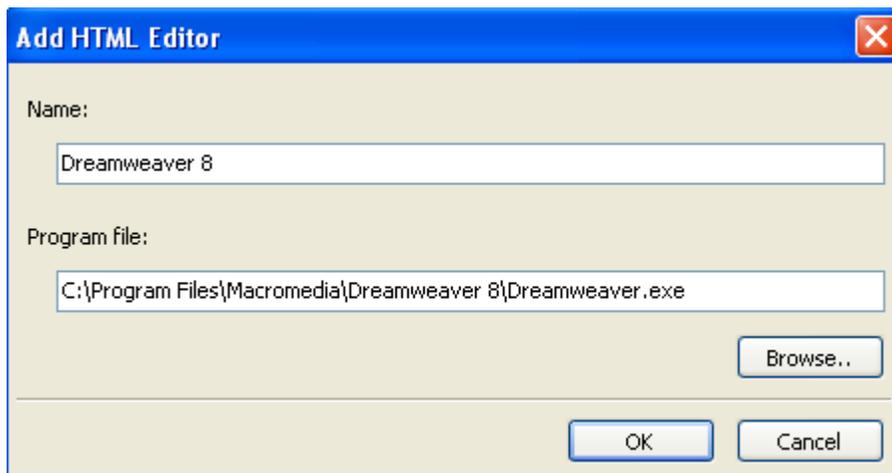
Adding an HTML Editor

In some cases, the HTML editor you would like to use may be missing from the list of HTML editors. You can search the system for available editors, or add your own manually:

1. In the Doc-To-Help Project Editor, select **Tools | Options** to open the **Options** dialog box.
2. Select the **HTML Editors** tab. A list of available HTML editors appears.



3. Click **Get System Editors** to update the list with all editors on your system. If your HTML editor is still not appearing, you must add it manually.
4. Click the **Add** button. The **Add HTML Editor** dialog box appears.



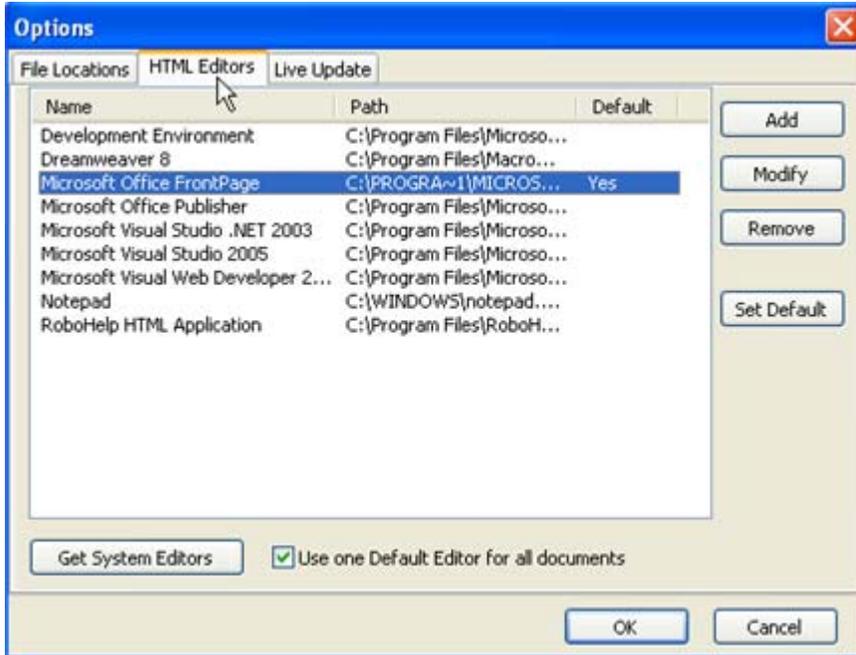
5. Enter a name for the editor in the **Name** textbox.
6. Under **Program file**, click **Browse** and locate the program in the **Open** dialog box.
7. Select the file and click **Open**.
8. Click **OK** to close the **Add HTML Editor** dialog box. The editor now appears in the list of HTML Editors.
9. Click **OK** or set the editor as your default HTML editor.

Removing an HTML Editor

You can also remove HTML editors that you do not want to use. Removing an editor from the list does not remove it from your computer.

1. In the Doc-To-Help Project Editor, select **Tools | Options** to open the **Options** dialog box.

2. Select the **HTML Editors** tab. A list of available HTML editors appears.



3. Select the editor you want to remove and click the **Remove** button. Doc-To-Help asks you to verify that you want the editor removed from the list.



4. Click **Yes**. You are returned to the list of HTML editors, and the selected editor is no longer listed.

Opening an HTML Document

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click the document or you wish to open and choose **Open** from the shortcut menu. The document you have chosen opens within your [default HTML editor](#) (page 187).

Organizing Your Project Files

When you first build a Help target, Doc-To-Help creates a subdirectory (within the directory containing the project file) for all intermediate and final output. This keeps the generated files separate from your source files. Note that you can change the name of any Help target directory by setting the Help target [Folder](#) directory.

The following table shows the mapping between the directory name and the default Help target:

MSHelp	Microsoft Help 2.0
NetHelp	NETHelp (previously HTML 4.0)
HTMLHelp	HTML Help 1.2

Manual	Printed Manual
Help	WinHelp 4.0
Help Target Name	User Created Help Target

IMPORTANT: Do not place any of your own files in these subdirectories, as Doc-To-Help may overwrite or delete them during the build process. For this reason, placing multiple Doc-To-Help projects (*.d2h* files) in a single directory is also not recommended.

Doc-To-Help stores document and template pathnames relative to the directory that contains the project file. Therefore, it is a good idea to keep your source documents (*.doc* and *.htm* files) and Doc-To-Help projects (*.d2h* files) in the same directory. That way, if you copy the entire directory to another location, Doc-To-Help will still be able to find your source documents.

If a Doc-To-Help project contains one or more Microsoft Word templates (*.dot* files) or cascading style sheets (*.css*), you can keep them in the same directory as the project file. However, if you create many Help projects, you may not want to have a duplicate set of templates for each one. In that case, you can keep one copy of the templates in a common directory, and each project can refer to them by their absolute pathnames. If you follow this approach, remember to copy the templates when transferring your source files to another machine.

Adding a Document to a Project

Many times it is advantageous to have your Help file separated into several smaller source documents. One of the advantages of using separate source documents for each topic is that it provides you with the flexibility of having several authors working on the project simultaneously. Additionally, when changes are made to topics, Doc-To-Help only needs to compile the source documents that have been modified, greatly reducing the compile time.

Document Properties

By selecting the **Documents** item on the project editor tree list, you are presented with a list of the project source documents in the right pane. By choosing any one of the documents, the lower left pane displays the property settings for that document.

Miscellaneous Properties

AdjustLeftIndent	Controls whether paragraph indentation is adjusted to account for wide margins when building online help.
KeepOutlineNumbers	When enabled, outline numbers are included as part of topic titles and are present in help targets.
Level	Returns the level of a document in the document tree.
Modified	Returns the date and time of the last saved modification in the source document.
Name	Returns a string containing the name of a specified object (read-only).
Order	Determines the order of the documents in the Help contents.
SingleTopic	Determines whether the source document can contain multiple topics or just one topic.
Size	Returns the size of the source document.
SplitFile	Controls whether a document is split into separate topic files for HTML-based platforms.
SplitFileWithHHCtrl	When True, a single HTML file is generated using intrinsic HTML Help commands to give the appearance of separate HTML files for each topic in the document. When False, individual HTML files are generated for each topic in the document.
Style	Determines the style of the topic contained in a single-topic document.

TextPopups	If True, generates a plain text only version of the Help file for context sensitive Help topics.
Title	Represents the title string of a document.
Type	Indicates whether the source document is a Word document or an HTML document.
SearchEnabled	Controls whether ComponentOne Natural Search is enabled for the document.

Adding a Source Document

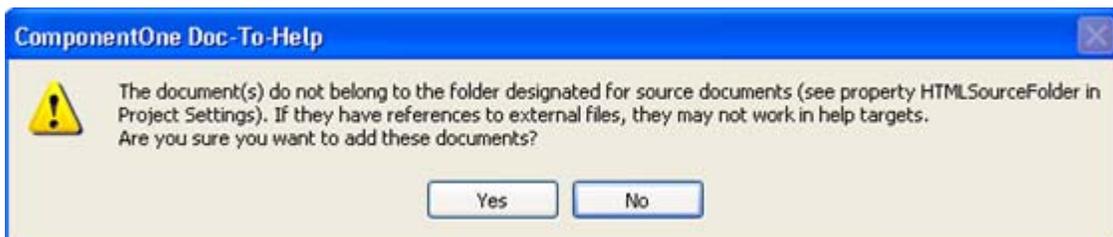
Adding a Word Document

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click anywhere in the document list in the right pane and choose **Add Documents** from the shortcut menu. The **Add Documents** dialog box opens.
4. Select *Word Documents (*.doc)* from the **Files of type** drop-down list if it is not selected already.
5. Select the document you wish to add and click **Open**.

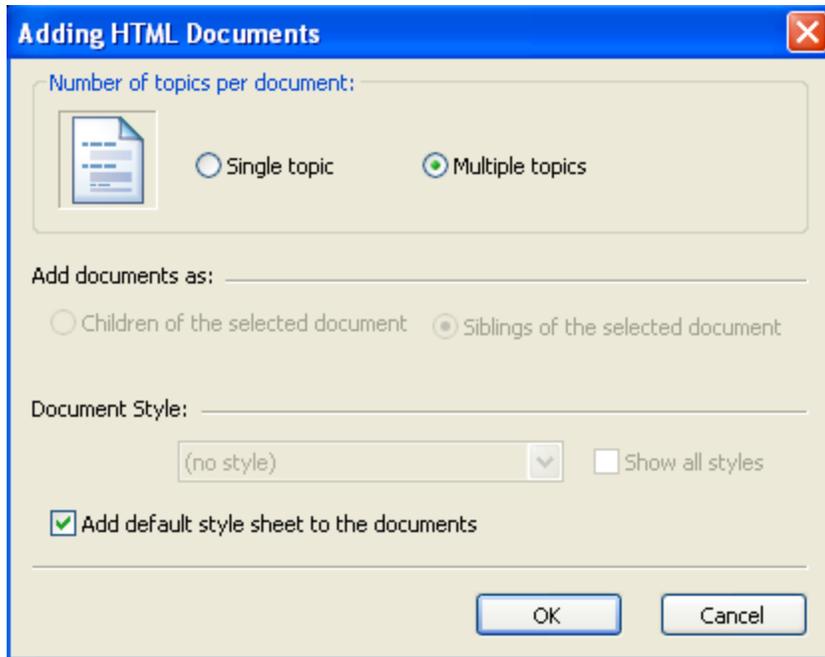
Note that the new document has been added to the list. This new document contains the default properties and can be modified as necessary.

Adding an HTML Document

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click anywhere in the document list in the right pane and choose **Add Documents** from the shortcut menu. The **Add Documents** dialog box opens.
4. Select *HTML Format (*.htm, *.html)* from the **Files of type** drop-down list if it is not selected already.
5. Select the document you wish to add and click **Open**. If the document you selected is not located within the folder specified in the project's [HTMLSourceFolder](#) property, the following warning appears:



6. Click **Yes** to continue or **No** to cancel and return to the project editor.
7. If you did not receive this warning, or if **Yes** was clicked in the previous step, the **Adding HTML Documents** window appears.

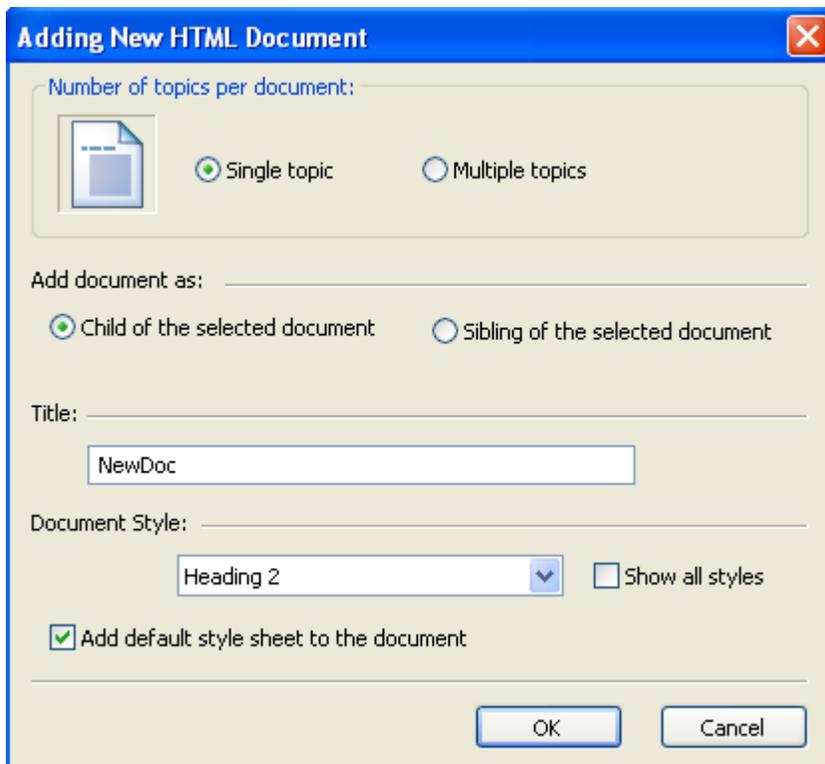


- Under **Number of topics per document**, specify *Single* or *Multiple topics*.
 - Under **Add document as**, select *Child* or *Sibling of the selected document*. Child places the document a level below the selected document. Sibling places the document on the same level as the selected document.
 - Specify a title for the new document in the **Title** text box.
 - Choose a document style from the **Document Style** drop-down list. If the style you are looking for is not listed, select Show all styles, which lists all paragraph styles in the project.
 - **Add default style sheet to the document** is checked by default. This ensures that the cascading style sheet specified in the [HTMLDefaultCSS](#) property of the project is automatically attached to the new document. If you do not want the style sheet attached, uncheck this box.
8. Click **OK** to close the **Adding HTML Documents** window. The new document is added to the list of documents. This new document contains the default properties and can be modified as necessary.

Alternatively, you can add a new document by clicking the **Project** menu, choosing **New Document**, then choosing the document.

Creating a New Document

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click anywhere in the document list in the right pane and choose **New Document** from the shortcut menu. The **New Document** dialog box appears. Make sure that the **Save as type** drop-down is set to the type of document you want to create.
4. Enter a name for the document in the **File name** field and click **Save**. If you created a Word document, it is added to the document tree and Word opens the new document. If you are creating an HTML document, the **Adding New HTML Document** window appears.



- Under **Number of topics per document**, specify *Single* or *Multiple topics*. HTML documents can contain a single topic or multiple topics. Single topic documents can have children documents, whereas multiple topic documents cannot.
- Under **Add document as**, select *Child* or *Sibling of the selected document*. Child places the document a level below the selected document. Sibling places the document on the same level as the selected document.

Note: To add a child document to the document tree, it must be attached to a single topic HTML file.

- Specify a title for the new document in the **Title** text box.
- Choose a document style from the **Document Style** drop-down. The document style level corresponds to the level of the document in the document tree. If the style you are looking for is not listed, select *Show all styles*, which lists all paragraph styles in the project.
- **Add default style sheet to the document** is checked by default. This ensures that the cascading style sheet specified in the [HTMLDefaultCSS](#) property of the project is automatically attached to the new document. If you do not want the style sheet attached, uncheck this box.

Note: Selecting a style that does not correspond to the document tree will result in a warning that the Style level does not correspond to the level of the document tree. It will ask you if you want to choose a different style level before proceeding. If you choose to continue without making changes, your help target will be built according to the Topic Style settings of each document.

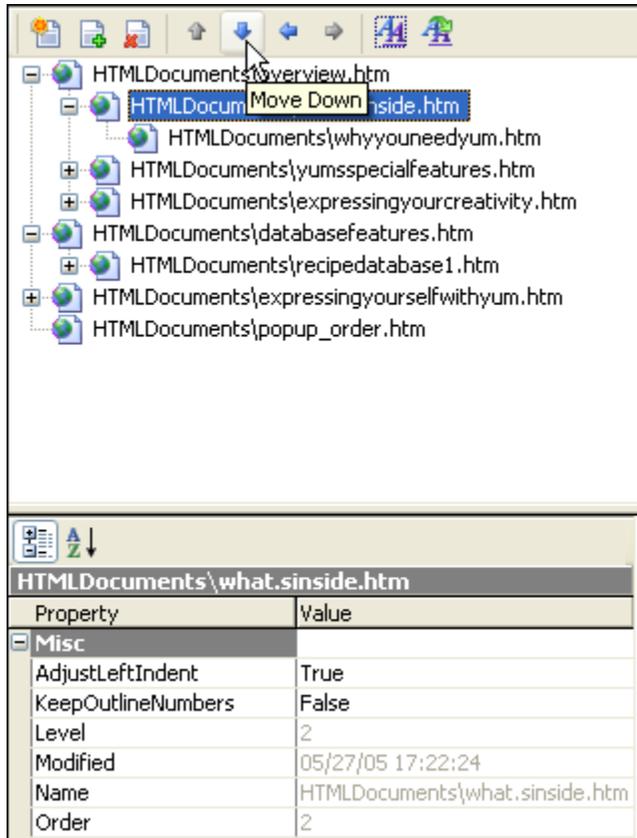
5. Click **OK** to close the **Adding New HTML Document** window. The new document is added to the document tree and your default HTML editor opens the new document.

Modifying the Order of the Source Documents

In a project that contains multiple source documents, the order in which they were added to the project determines the [Order](#) property for each document. Since this may not be the order you intended, you can rearrange it.

1. On the Icon bar, click **Project**.

2. Select the **Documents** item from the left pane.
3. Select the source document you wish to edit.
4. Change the [Order](#) property by pressing the **Move Up** or **Move Down** arrow located at the top of the documents property pane in the **Documents** toolbar.



You have two other options in addition to using the **Move Up** and **Move Down** arrows when changing the order of your documents:

Drag and Drop

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Select the source document you wish to move.
4. While holding the left mouse key, drag the document to the desired position within the document tree.

Cut and Paste

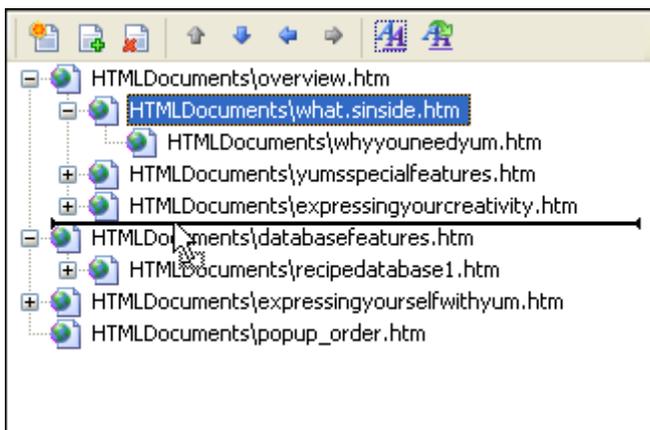
1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click the source document you wish to move and select **Cut** from the context menu.
4. Right-click the document to precede the document you are moving and select **Paste** from the context menu. The document you cut now follows this document in the tree.

Modifying the Document Hierarchy

When using multiple topic Word and HTML source documents, you can only change the document order in the document tree. With single topic HTML documents, you can also change the hierarchy.

Note: HTML documents can be single or multiple topic, while Word documents can only be multiple topic. See [Single and Multiple Topic Documents](#) (page 183) for more information.

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Select a single topic HTML source document.
4. While holding the left mouse key, drag the document to the desired position within the document tree. If you would like the document to be a child of another document, drag it to the parent document so that the parent document is highlighted, and release the mouse button. The document is placed under the parent document.



Alternatively, you can use the left and right arrow buttons to change the hierarchy of the tree:

1. Move the document you would like to become the child of another document so it appears below the parent document.
2. Click the **Move Right** arrow. The document becomes the child of the document preceding it.
3. Click the **Move Left** arrow to make the document a sibling of the document preceding it.

Removing a File From a Project

1. Based on what file you wish to remove, click **Project** on the Icon bar and choose the **Documents** or **Templates** item from the left pane.
2. Select and right-click the file you wish to remove and choose **Remove** from the shortcut menu.

Note: When you remove a document from a project, all of its topic records are deleted. However, no index keywords or groups are deleted, even if they were created during compilation of the file that was removed.

Compiling Source Documents

When you build the current Help target, Doc-To-Help compiles your source documents and combines the resulting files into a single entity. However, there are times when you may want to compile a source document individually without updating the entire target. For example, if a build fails because a topic title exceeds the 255-character limit, you can edit the topic title, and then compile that document alone to verify that the problem is fixed before proceeding with a full build.

To compile a single document:

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Select and right-click the document that you wish to compile and select **Compile** from the shortcut menu.

Alternatively, you can compile a document by highlighting the document and clicking the **Compile File** button



on the **Project** toolbar.

To compile all documents in the project:

From the **Build** menu, select **Compile All**.

To interrupt compilation of one or more documents:

Click the **Stop Build** button  on the **Project** toolbar.



Doc-To-Help Markup in Source Documents

Markup is simply information that is added to your source documents which tells Doc-To-Help what your help target should look like. You can easily insert markup into your Word and HTML documents using the toolbars provided by Doc-To-Help.

In Word documents, Doc-To-Help 2006 has two different kinds of markup:

- **D2H-proprietary markup:** D2H-dependent information is usually stored in Word comments in the document. This markup can be created only by using the **Doc-To-Help** toolbar in Word. Some examples of D2H-proprietary markup include the **Add Topic Link** button and the **Add Dynamic Link** button.
- **D2HML:** This markup can be created in Word documents simply by formatting certain text with the appropriate D2HML styles using the **D2HML Styles** toolbar. The user does not need Doc-To-Help to do this; it can even be done on a machine where Doc-To-Help is not installed or even if you do not know Doc-To-Help. The fact that D2HML is non-proprietary, pure Word or HTML, and can be used even without Doc-To-Help is the main advantage of using D2HML.

In HTML documents, only **D2HML** markup is supported; there is no D2H-proprietary markup. This markup can be created in any HTML editor by formatting text using the appropriate D2HML styles in HTML code. Doc-To-Help also provides a **D2HML Styles** toolbar for FrontPage and Dreamweaver to facilitate formatting with D2HML styles.

Note: The **D2HML Styles** toolbar is supported in FrontPage XP and 2003 versions. It is not supported in FrontPage 2000.

Please see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for additional information.

Templates and Cascading Style Sheets

When working with ComponentOne Doc-To-Help 2006, you supply the source text of a Help system by adding one or more Microsoft Word or HTML documents to a Doc-To-Help project file. Only the pathname is stored, not the document contents.

You can also supply optional Microsoft Word templates and cascading style sheets to apply target-specific formatting during compilation. As with documents, only the pathname is stored.

Select the **Templates** item in the left pane to view all of the templates and style sheets in the Help project.

Adding Templates to a Project

You can easily supply optional Microsoft Word templates to apply target-specific formatting during compilation.

Select the **Templates** item on the project editor tree list to get a list of the default templates in the right pane.

Template Properties

By selecting the **Templates** item on the project editor tree list, you are presented with a list of the default templates in the right pane. Choose one of them to display the property settings for that template in the lower left pane.

Miscellaneous Properties

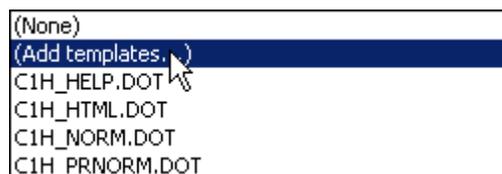
[Name](#) Returns a string containing the name of a specified object (read-only).

Adding a New Template

1. On the Icon bar, click **Project**.
2. Select the **Templates** item from the left pane.
3. Right-click anywhere in the templates list in the right pane and choose **Add Templates** from the shortcut menu. The **Add Templates** dialog box opens.
4. Select the template you wish to add and click **Open**.

The new template is added to the list. Note that you'll need to add the template to the project in order to attach it to any of the Help targets.

Alternatively, you can select the [Template](#) property for any Help target and from the template selection drop-down, pick **Add templates**. By using this method, the template is added into the project and attached as the template for the current Help target.



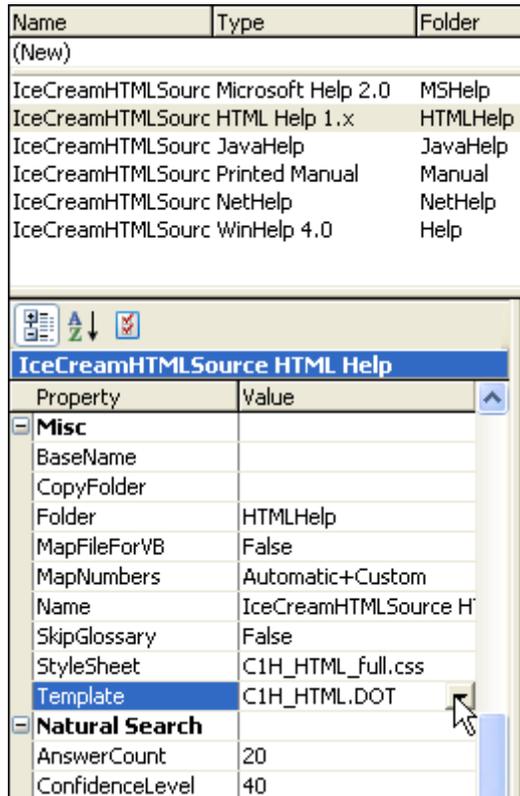
Assigning a Template to a Specific Help Target

Once you've added a template to a project, assign it to a specific Help target so that it will be applied to the source document copies during compilation. To assign a template to a Help target:

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Select the Help target you wish to edit from the right pane.

Note that the property pane changes to reflect the Help target chosen.

- In the Help target property pane, locate the [Template](#) property. In the text box to the right of the property, select the name of the desired template.



Using Custom Templates

One of the strengths of Doc-To-Help is the ability to use custom templates. For most authors, the default templates supplied with Doc-To-Help should be sufficient for most applications, but in some cases it may be necessary to use a customized template. Generally, it is a good practice to store these custom templates with the associated project. In the past, storing a template with the project could be an issue when moving the project because the template files were stored with an absolute path (full path to template directory). Now all template files are stored using a relative path, no matter where you move the project, the templates associations remain intact.

Note: If the template being used by a target is stored in both the project directory and the Word template directory, the template in the project directory will be used by default.

Opening a Template from a Project

To open a template in Microsoft Word:

- On the Icon bar, click **Project**.
- Select the **Templates** item from the left pane.
- Right-click the template you wish to open and choose **Open** from the shortcut menu.

The template you have chosen opens within Microsoft Word.

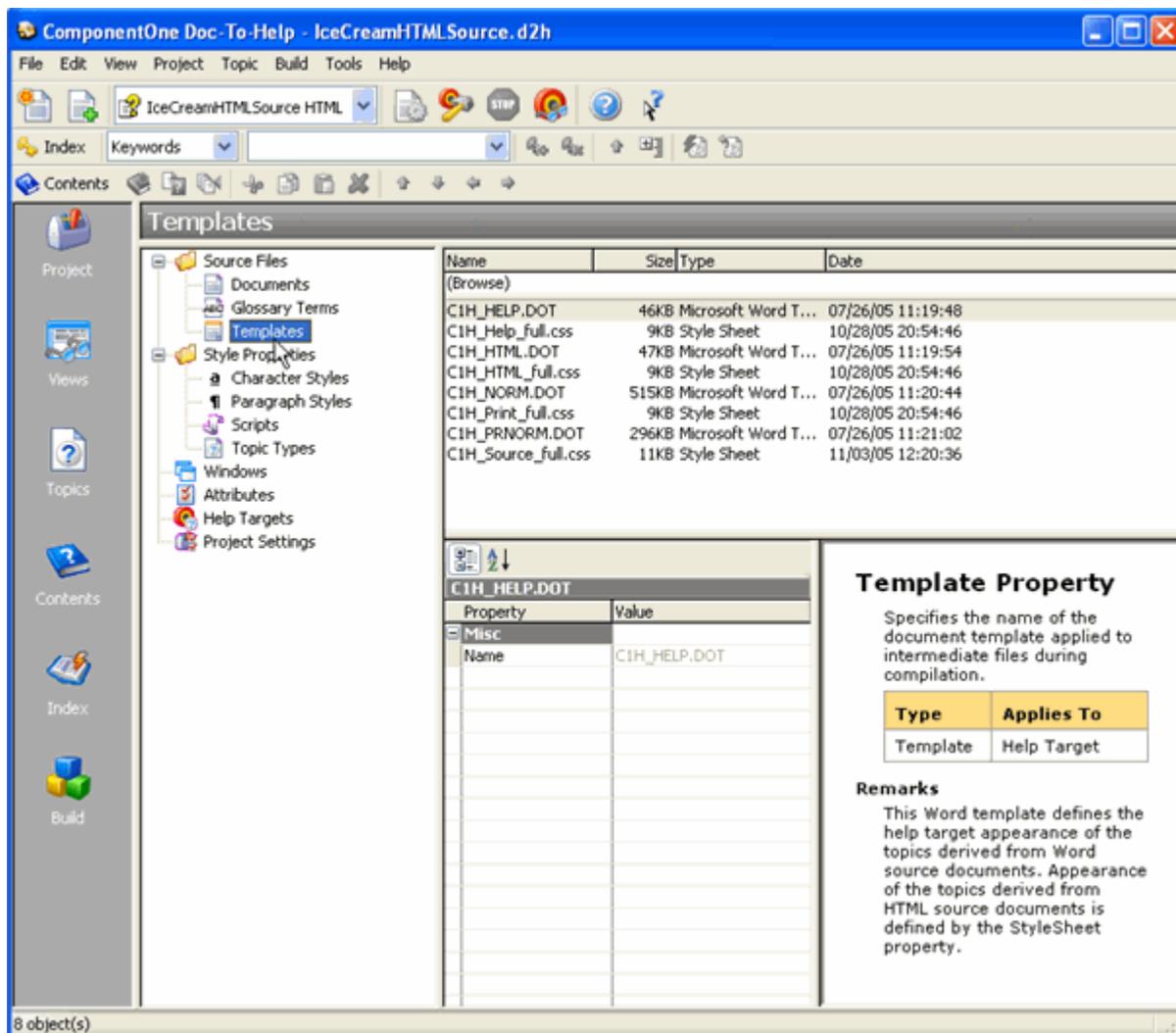
Using Cascading Style Sheets

Doc-To-Help 2006 provides and uses cascading style sheets (CSS) for HTML documents in much the same way it provides and uses Word templates for Word documents. Predefined or customized Word templates are attached to Word source documents and help targets to be used by Doc-To-Help to determine the formatting and styles of your help files. Similarly, for HTML documents, Doc-To-Help uses style sheets attached to HTML source documents and help targets.

The style sheet attached to your HTML source documents is determined by the [HTMLDefaultCSS](#) property under **Project Settings**, which is set by default when you create a project. You may also choose to add and attach your own cascading style sheet. Once a style sheet is attached, you can edit your source documents using the existing styles or, if necessary, create new styles in the style sheet. See [Source Document Style Sheets](#) (page 201) for more information.

Use the [StyleSheet](#) property of each help target to determine the style sheet that is attached to every HTML document in your help file, defining the appearance of the target. See [Help Target Style Sheets](#) (page 205) for more information.

All style sheets and Word templates available within a project are listed in the right-hand pane of the project editor when the **Project** icon is clicked and the **Templates** node is selected. New style sheets can also be easily added later.



Source Document Style Sheets

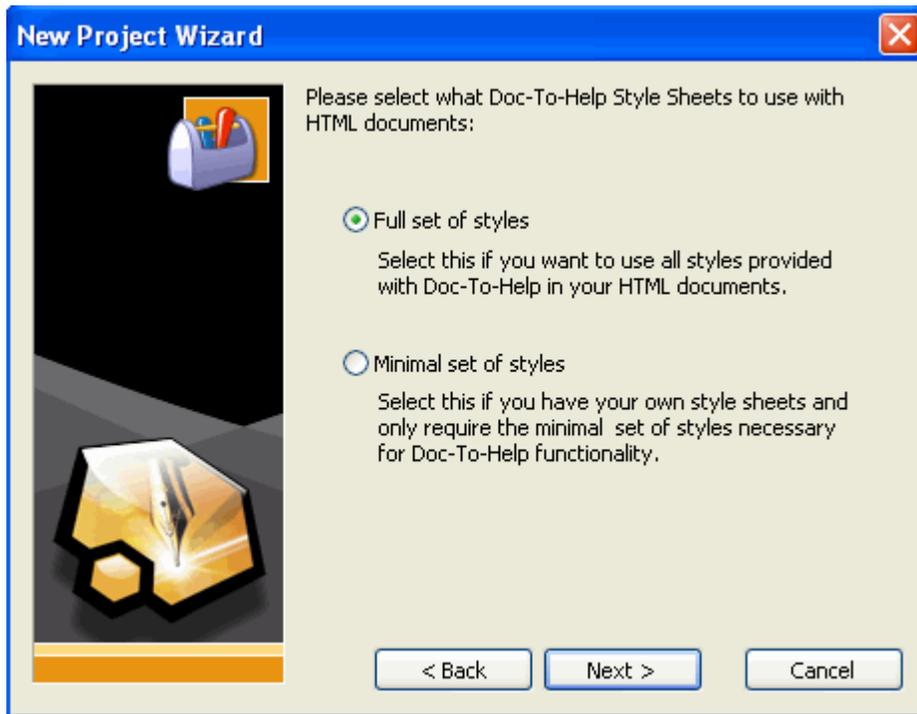
The source style sheet is specified in the Project.[HTMLDefaultCSS](#) property. This default style sheet is attached to every source HTML document when it is added to the project.

Note: There is one exception. If the *Add default style sheet to the document* check box is unchecked in the **Adding HTML Documents** or **Adding New HTML document** dialog box when adding an existing or new document to the project, the default style sheet is not attached to the source document.

Source HTML documents can have other style sheets attached to them as well, in addition to the Doc-To-Help source style sheet, if you want to use other styles besides the ones provided with Doc-To-Help. The main purpose of attaching the Doc-To-Help style sheet is to make [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) styles available in the document.

When building a help target, Doc-To-Help detaches the source document style sheet from the HTML documents and attaches the help target style sheet, just as it detaches the source templates from Word documents and attaches the target template. For more information on Word templates, see [Templates and Cascading Style Sheets](#) (page 199).

When you create a new project using *Mixed* or *HTML source documents*, there are two options for the Doc-To-Help style sheets you can use: *Full set of styles* and *Minimal set of styles*.



The *Full set of styles* option attaches the **C1H_Source_full.css** style sheet to your HTML source documents, which provides a rich set of styles designed for consistency with Word styles defined in Doc-To-Help templates.

The *Minimal set of styles* option attaches the **C1H_Source_short.css** style sheet to your HTML source documents, which provides only the D2HML styles in the source style sheet.

Note: If you are using your own style sheets, select **Minimal set of styles** when creating a new Doc-To-Help project. Then Doc-To-Help will add a minimal style sheet to your document, containing only the necessary D2HML styles for defining help authoring behaviors and nothing else. You can freely use your own style sheets in your source documents to control their appearance, Doc-To-Help will preserve them in the target and will not interfere with them.

If you created an HTML source document outside of the Doc-To-Help project without using the New Project wizard, you can apply the style sheets by including the following HTML code within the <HEAD> tags of your source document:

- `<LINK title=_d2hDefaultCSS href="..\DefaultCSSFiles\C1H_Source_full.css" rel=stylesheet>`

This attaches the **C1H_Source_full.css** style sheet to the document.

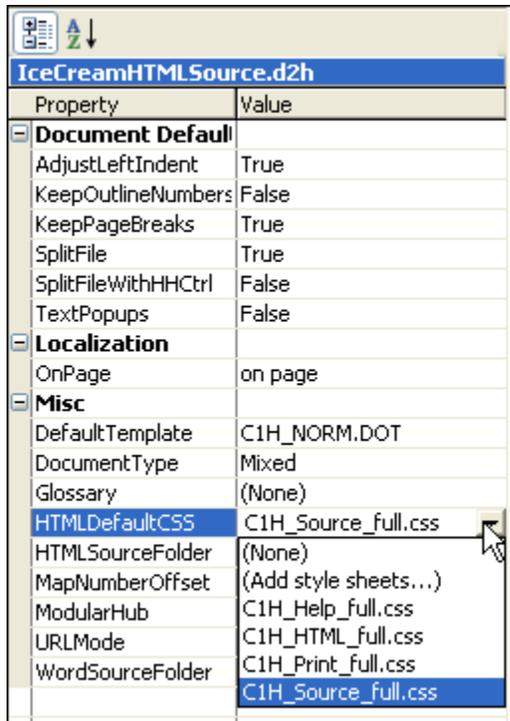
- `<LINK title=_d2hDefaultCSS href="..\DefaultCSSFiles\C1H_Source_short.css" rel=stylesheet>`

This attaches the C1H_Source_short.css style sheet to the document.

Setting the Default CSS

To set the default style sheet to be attached to every source HTML document added to the project:

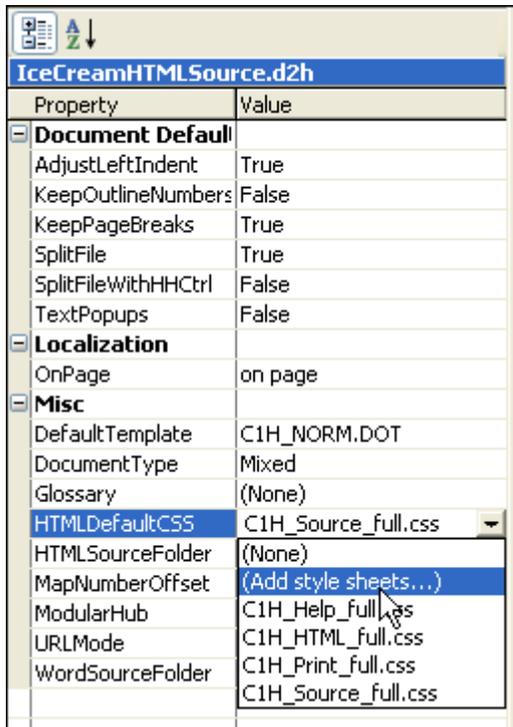
1. Click the **Project** icon.
2. Select **Project Settings** in the left pane.
3. Under **Misc**, locate the [HTMLDefaultCSS](#) property and click the drop-down arrow.



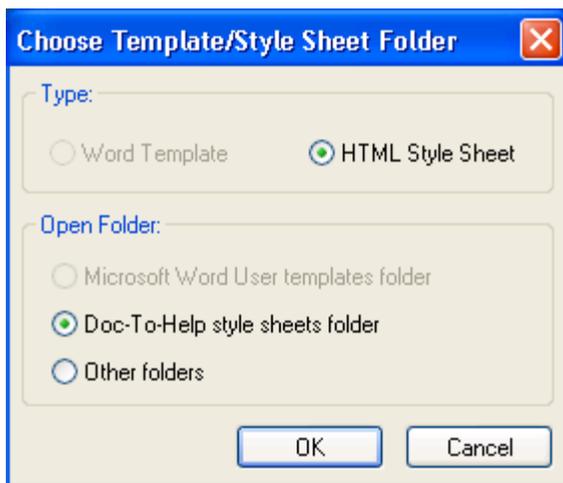
4. Select a style sheet from the list of available style sheets. This style sheet is attached to all documents added to the project.

Adding a New Style Sheet to the Source Document

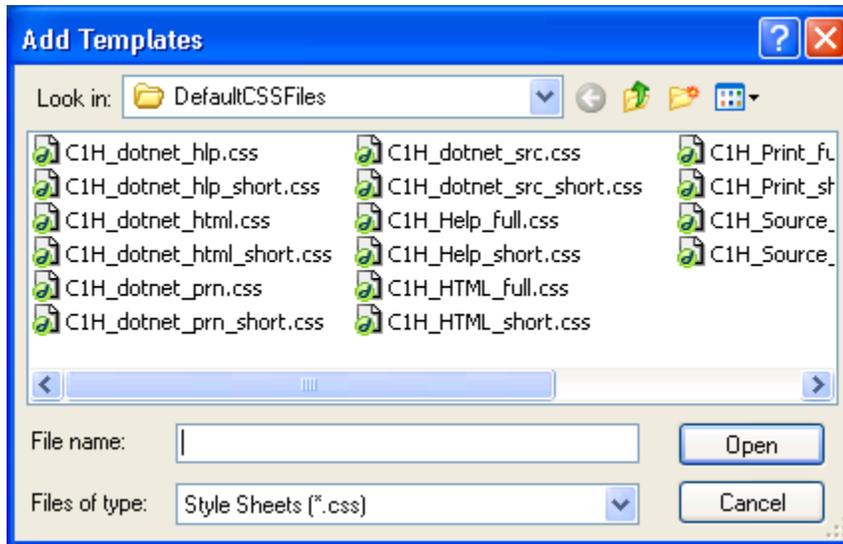
1. Click the **Project** icon.
2. Select **Project Settings** in the left pane.
3. Under **Misc**, locate the [HTMLDefaultCSS](#) property and click the drop-down arrow.



4. Select **Add style sheets**. The **Template/Style Sheet** dialog box opens.



5. If you want to use a predefined Doc-To-Help style sheet, make sure **Doc-To-Help style sheets folder** is selected. Select **Other folders** if you want to use your own style sheet.
6. Click **OK**. The **Add Templates** dialog box opens the DefaultCSSFiles folder containing the Doc-To-Help style sheets.



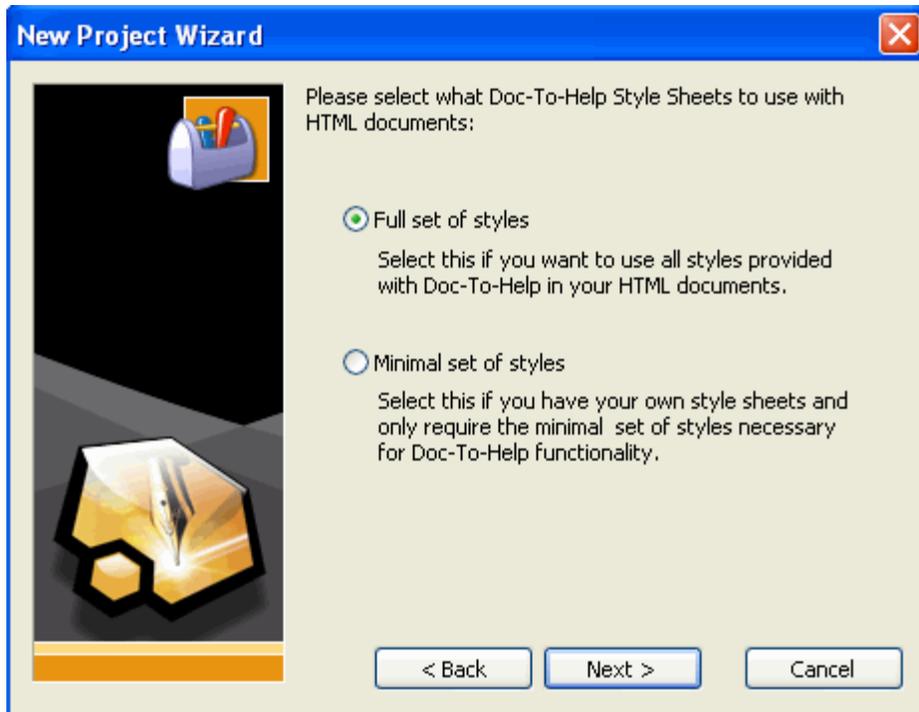
7. Select a style sheet or browse for another style sheet using the **Look in** drop-down arrow, and click **Open**. The style sheet is added to the drop-down list.

Help Target Style Sheets

When using Word source documents, the HelpTarget.[Template](#) property determines the formatting and styles used in the help target. When using HTML source documents, the HelpTarget.[StyleSheet](#) property is used for the same purpose. Similar to the way the HelpTarget.[Template](#) property works, the HelpTarget.[StyleSheet](#) property specifies the style sheet to be attached to every HTML document when Doc-To-Help builds the target. The style sheet provides a consistent appearance in the specified help target for topics originated from different HTML documents.

Note: The HelpTarget.[Template](#) property affects only Word documents in the project, not HTML documents, with the exception of the Manual target where the [Template](#) property is used to attach the template to the resulting Word document, regardless of whether it was produced from Word or HTML source documents.

When you choose one of the style sheet options in the New Project wizard, you determine the style sheet to be attached to the help target.

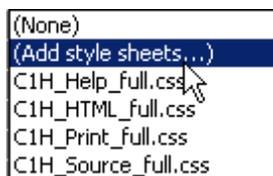


If *Full set of styles* is chosen, the HTML [StyleSheet](#) property is set, by default, to **C1H_HTML_full.css**. This option is intended for users who do not have their own pre-designed style sheets and would rather use the style sheets provided by Doc-To-Help to create a consistent appearance throughout their help files. Full-style Doc-To-Help style sheets are designed to provide the maximum possible consistency in appearance in the help target between HTML generated from Word documents and HTML generated from the HTML source documents.

If *Minimal set of styles* is chosen, the HTML [StyleSheet](#) property is set, by default, to **C1H_HTML_short.css**. This option is intended for users who already have their own style sheets and want to use them to define the appearance in their HTML source documents and help targets.

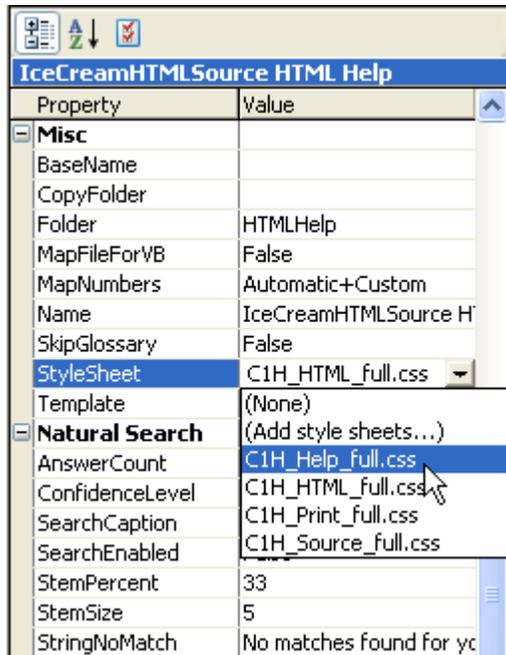
The value of the HelpTarget [StyleSheet](#) property can be changed at any time after the project has been created. If you do not want a style sheet attached to the HTML documents, you can set this property to **None**. This option can be used in more advanced cases where you may want to override the default Doc-To-Help behavior of ensuring the consistent appearance throughout the topics in the target.

Similarly, you can select the [StyleSheet](#) property for any Help target and choose **Add style sheets** from the style sheet selection drop-down. The style sheet is added to the project and attaches as the style sheet for the current Help target.

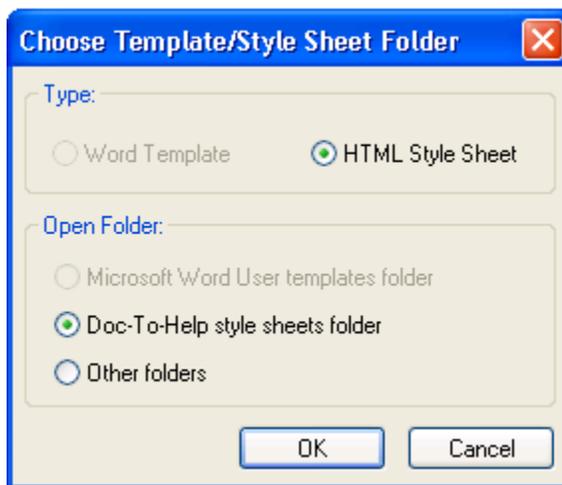


Setting the Help Target Style Sheet

1. Click the **Project** icon.
2. Select **Help Targets** from the left pane.
3. Choose your help target in the right pane.
4. Under **Misc**, locate the **StyleSheet** property click the drop-down arrow and select a style sheet from the list of available style sheets.



5. If the style sheet you want to attach is not available from the list, select **Add Style Sheets**. The **Choose Template/Style Sheet Folder** dialog box opens and **HTML Style Sheet** is selected.



6. In the **Choose Template/Style Sheet Folder** dialog box, you have two options:
- Choose one of the Doc-To-Help style sheets by leaving **Doc-To-Help style sheets folder** selected and clicking **OK**. The **Add Templates** window opens to the DefaultCSSFiles folder. Select one of the style sheets and click **Open** to attach it to the project.

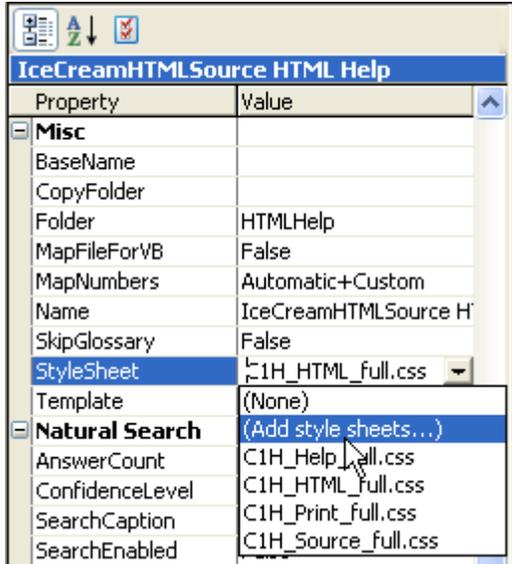
Note: The predefined Doc-To-Help style sheets are located in C:\Program Files\ComponentOne\DocToHelp\DefaultCSSFiles.

- Select **Other folders** and click **OK**. The **Add Templates** window opens, and you can browse to find a style sheet. Click **Open** to attach it to the project.

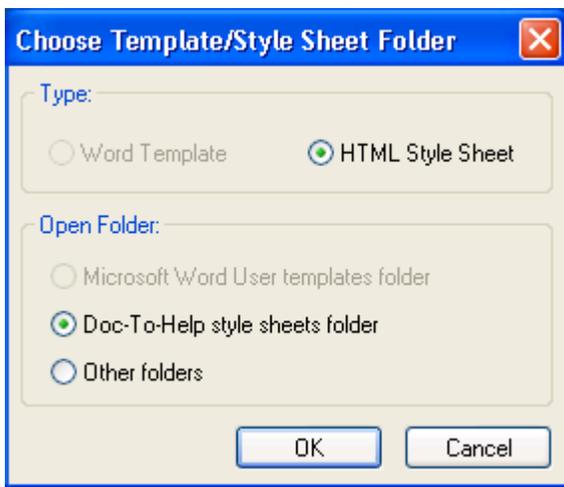
Adding a New Style Sheet to the Help Target

1. Click the **Project** icon.
2. Select **Help Targets** in the left pane.

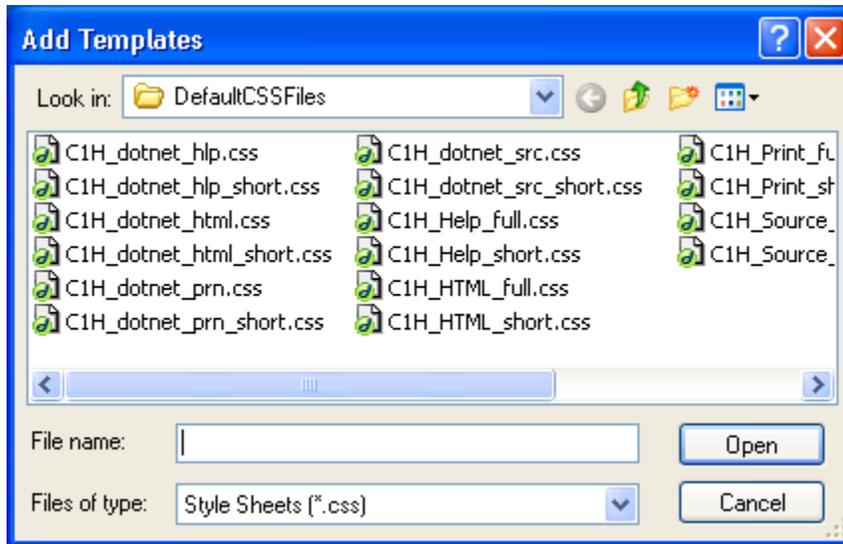
- Under **Misc**, locate the [StyleSheet](#) property and click the drop-down arrow.



- Select **Add style sheets**. The **Template/Style Sheet** dialog box opens.



- If you want to use a predefined Doc-To-Help style sheet, make sure **Doc-To-Help style sheets folder** is selected. Select **Other folders** if you want to use your own style sheet.
- Click **OK**. The **Add Templates** dialog box opens the DefaultCSSFiles folder containing the Doc-To-Help style sheets.



7. Select a style sheet or browse for another style sheet using the **Look in** drop-down arrow, and click **Open**. The style sheet is added to the drop-down list.

Opening a Cascading Style Sheet From a Project

1. On the Icon bar, click **Project**.
2. Select the **Templates** item from the left pane.
3. Right-click the style sheet you wish to open and choose **Open** from the shortcut menu. The sheet you have chosen opens within your [default HTML editor](#) (page 187).

Using Styles in Doc-To-Help

In Microsoft Word and HTML documents, you can use styles to apply paragraph formats (such as indentation and spacing) and character formats (such as font styles and colors) to give your documents a uniform appearance. If you apply styles consistently, any change you make to a style definition will be propagated throughout the document, and the affected regions will be updated accordingly. Styles are also used in Word's outline view to specify the hierarchy of paragraphs within a Word document.

Any paragraph or character style that has special help-authoring behavior should have a matching style in the Doc-To-Help project in order for it to work correctly in your help target. Styles used for appearance only do not need to be in the Doc-To-Help project.

Doc-To-Help provides many predefined styles, which are added to a project by default. If one of the predefined styles cannot be used, you can create your own paragraph and character styles, but you must define a matching style within the Doc-To-Help project. Through those matching Doc-To-Help styles, you define the behavior of the compiled Help system. Please note that this is an advanced feature, and creating your own styles is not usually necessary.

For more information on defining styles in Microsoft Word, see [Working with Styles in Word](#) (page 227) or consult your default HTML editor's documentation for information on defining styles in an HTML document.

Using Paragraph Styles

For each paragraph style that begins a topic or has a special help-authoring behavior, a like-named paragraph style must be defined in the project file so that Doc-To-Help compiles the Help system properly. For example, if you have a paragraph style named SubHeading, you should also have a paragraph style in the project file called SubHeading. In this way, you can simply select which properties you want to apply to the SubHeading style in the property pane of the Doc-To-Help project editor. Defining a style is not usually necessary, because Doc-To-Help provides many predefined styles, but if you need to use your own styles, remember to define a like-named style in your project.

Note that Topic Types and Paragraph Styles have many duplicate properties. When a paragraph style has a [TopicType](#) associated with it, the duplicate Topic Type properties override the Paragraph Style properties. To make it easy to determine which properties are overridden, Doc-To-Help dims the Paragraph Style properties when unavailable.

Paragraph Style Properties

You can edit the Paragraph Style properties by accessing property pane for the specific Paragraph Style. For more information, see [Editing Paragraph Style Properties](#) (page 215).

Display Properties

ContentsOnly	When True, omits topics defined with this paragraph style from the Help target, but uses them as book titles in the Help contents. This property is duplicated in the Topic Types.
Hidden	When True, omits text formatted with this style from the help target.
Nonscrolling	When the NonScrolling property of a paragraph style is set to True, any topic headings formatted with that paragraph style appear in the non-scrolling region of the Help window. This property is duplicated in the Topic Types.
OnlineOnly	When True, omits paragraphs formatted with this style from the printed manual target only.
Popup	When the Popup property of a paragraph style is set to True, any topic formatted with that paragraph style appear in a pop-up window on top of the Help window. This property is duplicated in the Topic Types.

[Replacement](#) Sets or returns a string that overrides the default text generated by Doc-To-Help. For a paragraph style with [AutoNext](#) set to True, the Replacement string is used instead of the topic title next to the generated button. This property is duplicated in the Topic Types.

[SuppressEmptyTopics](#) If set to True, omits topics defined with this paragraph style or topic type that have no text except heading from online help except in TOC.

[Untitled](#) When True, the topic title is not displayed in the Help window. This property is duplicated in the Topic Types.

[Window](#) Shows the name of the window in which topics formatted in this style are displayed. This property is duplicated in the Topic Types.

Indexing Properties

[AutoGlossaryLinks](#) When True, Doc-To-Help examines the project documents for text that matches glossary entries in the glossary document during compilation, converting these matches into hyperlinks. Whenever the user clicks on one of the hyperlinks, a DHTML pop-up opens containing the corresponding glossary entry description.

Indexing Properties

[AutoKeyword](#) When True, this property automatically creates index keywords from topic titles formatted with that paragraph style and associates them with the appropriate topic. This property is duplicated in the Topic Types.

Miscellaneous Properties

[AutoContextID](#) When True, automatically creates Context IDs for all topics using this paragraph style. This property is duplicated in the Topic Types.

[HTMLName](#) Defines the name to be used to identify the style as a cascading style sheet style in HTML source documents.

[Name](#) Returns a string containing the name of a specified object (read-only). This property is duplicated in the Topic Types.

[Script](#) Displays the script to be run during compilation whenever text formatted with this style is encountered. This property is duplicated in the Topic Types.

[TopicType](#) Specifies a named set of display, navigation, and indexing characteristics to be associated with an object (such as what window the Help topic appears in, how the Help topic is accessed, and whether it gets a map number).

[Type](#) Determines the help authoring behavior associated with a style at compile time.

Value	Description
Body Text	Topic body.
Heading Text	Topic with no outline level.
Level 1	Topic at outline level 1.
Level 2	Topic at outline level 2.
Level 3	Topic at outline level 3.
Level 4	Topic at outline level 4.
Level 5	Topic at outline level 5.
Level 6	Topic at outline level 6.
Level 7	Topic at outline level 7.

Value	Description
Level 8	Topic at outline level 8.
Level 9	Topic at outline level 9.

Navigation Properties

AutoButtons	When True, displays subtopic buttons for paragraph styles with numeric outline levels greater than the parent level. This property is duplicated in the Topic Types.
AutoLink	Creates unique link tags for topics formatted with the paragraph style by enabling the auto link tag feature. This property is duplicated in the Topic Types.
AutoNavigate	Determines which paragraph styles are included in the navigation sequence. This property is duplicated in the Topic Types.
AutoNext	Enables the next topic button for any active paragraph style. This property is duplicated in the Topic Types.
ExplicitAccess	When True, defines a specific paragraph style as accessible only through a hyperlink. Paragraph styles defined as explicit are not accessible from the Help contents, the index list, or the text search. This property is duplicated in the Topic Types.
MidTopic	Defines a paragraph style as an “in topic” jump location similar to an HTML anchor. By defining a paragraph style as MidTopic and defining a character style as a jump, end users have jump functionality within a lengthy topic. This property is duplicated in the Topic Types.

Default Paragraph Styles

	Display Properties	Glossary Properties	Indexing Properties	Miscellaneous Properties	Navigation Properties
Heading 1	ContentsOnly=False Hidden=False Nonscrolling=True OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=True Untitled=False Window=main	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=H1 Name=Heading 1 Script=(None) TopicType=Contents Type=Level 1	AutoButtons=False AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False
Heading 2	ContentsOnly=False Hidden=False Nonscrolling=True OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=main	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=H2 Name=Heading 2 Script=(None) TopicType=Conceptual Type=Level 2	AutoButtons=True AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False
Heading 3	ContentsOnly=False Hidden=False Nonscrolling=True OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=main	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=H3 Name=Heading 3 Script=(None) TopicType=Conceptual Type=Level 3	AutoButtons=True AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False

	Display Properties	Glossary Properties	Indexing Properties	Miscellaneous Properties	Navigation Properties
Heading 4	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=proc	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=H4 Name=Heading 4 Script=(None) TopicType=Procedural Type=Level 4	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=False
Heading 5	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=proc	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=H5 Name=Heading 5 Script=(None) TopicType=Procedural Type=Level 5	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=False
MidTopic	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=(Default)	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=.MidTopic Name=MidTopic Script=(None) TopicType=(Inherit from Style) Type=Heading text	AutoButtons=False AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=True
RelatedHead	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=True Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=(Default)	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=.RelatedHead Name=RelatedHead Script=(None) TopicType=(Inherit from Style) Type=Body text	AutoButtons=False AutoLink=False AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=False
WhatsThis	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=True Window=(Default)	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=True HTMLName=.WhatsThis Name=WhatsThis Script=(None) TopicType=(Inherit from Style) Type=Heading text	AutoButtons=False AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False
Glossary Heading	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=False Window=(Default)	AutoGlossaryLinks=True	AutoKeyword=False	AutoContextID=False HTMLName=.GlossaryHeading Name=Glossary Heading Script=D2HGlossaryterm TopicType=Glossary Term Definition Type=Level 5	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False
Glossary Heading (no auto links)	ContentsOnly=False Hidden=False Nonscrolling=False OnlineOnly=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=False Window=(Default)	AutoGlossaryLinks=False	AutoKeyword=False	AutoContextID=False HTMLName=.GlossaryHeading gnoautolinks Name=Glossary Heading (no auto links) Script=D2HGlossaryterm TopicType=Glossary Term Definition Type=Level 5	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False

Creating New Paragraph Styles

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. Right-click anywhere in the paragraph style list pane on the right and select **New** from the shortcut menu.

Note that the cursor has moved to the top of the paragraph style list window.

4. Type the name of the new paragraph style and press **Enter**.

Note that the new paragraph style is added to the list.

Alternatively, you can add a new paragraph style by clicking the **Project** menu, choosing **New**, then clicking **Paragraph Style**.

Editing Paragraph Style Properties

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. In the right pane, select the name of the style to be edited.

Note that the property pane changes to show the properties of the selected style.

4. Change the properties as desired.

Note that your changes affect the selected style instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

Heading 1	
Property	Value
Display	
ContentsOnly	False
Hidden	False
Nonscrolling	True
OnlineOnly	False
Popup	False
Replacement	
SuppressEmptyTopics	True
Untitled	False
Window	main
Indexing	
AutoKeyword	False
Misc	
AutoContextID	False
HTMLName	H1
Name	Heading 1
Script	(None)
TopicType	Contents
Type	Level 1
Navigation	
AutoButtons	False
AutoLink	True
AutoNavigate	True
AutoNext	False
ExplicitAccess	False
MidTopic	False

Note: If your Paragraph style has a “parent” Topic Type associated with it, the duplicate Topic Type properties supersede the Paragraph style properties. See [Topic Type Properties](#) (page 222).

Using Character Styles

Doc-to-Help extends the concept of styles to include Help authoring behavior. For each character style, a like-named character style must be defined in the project file. For example, if you have a character style named *BoldText* in your source document, you should also have a character style in the project file called *BoldText*. In this way, you can simply select which properties you want to apply to the *BoldText* style in the property pane of the Doc-To-Help project editor.

Doc-To-Help comes with a wide variety of predefined character styles which can be used to format text and create hot spots using [Doc-To-Help Markup Language \(D2HML\)](#) (page 233), so it is not usually necessary to create new character styles. Character styles can be created in advanced cases when one of the predefined character styles cannot be used to get the desired result.

Character Style Properties

You can edit the Character Style properties by accessing property pane for the specific Character Style. For more information, see [Editing Character Style Properties](#) (page 219).

Display Properties

AffectsAppearance	Determines whether the style defines the appearance of the text (font, color, etc.) in the target help file, or is only used to create a hotspot or keyword and does not affect target appearance.
Hidden	When True, omits text formatted with this style from the help target.
PageRef	When producing a manual, this property places the page number of the referenced topic next to the text formatted with this style.
Replacement	Sets or returns a string that overrides the default text generated by Doc-To-Help. For a paragraph style with AutoNext set to True, the Replacement string is used instead of the topic title next to the generated button.
Window	Shows the name of the window in which topics formatted in this style are displayed.

Indexing Properties

AutoIndex	When True, this property automatically creates index keywords from text formatted with that character style and associates them with the topic that contains the text.
---------------------------	--

Miscellaneous Properties

Behavior	Determines the help authoring behavior associated with a character style at compile time.
Condition	For a character style, makes this style a conditional text style, places the specified condition on text formatted with this character style.
HTMLName	Defines the name to be used to identify the style as a cascading style sheet style in HTML source documents.
LinkType	Determines whether a link hot spot is a jump or a pop-up.
MultiLink	When False, only the first occurrence of a "jump" character style (in a given topic) generates a link. All other formatted occurrences are skipped.
Name	Returns a string containing the name of a specified object (read-only).
Script	Displays the script to be run during compilation whenever text formatted with this style is encountered.

Default Character Styles

	Display Properties	Indexing Properties	Miscellaneous Properties
C1H Conditional	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Conditional Text Condition= HTMLName=.C1HConditional LinkType=None MultiLink=True Name=C1H Conditional Script=(None)
C1H Contents Title	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Contents Title Condition= HTMLName=.C1HContentsTitle LinkType=None MultiLink=True Name=C1H Contents Title Script=(None)
C1H Context ID	AffectsAppearance=False Hidden=True PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Context ID Condition= HTMLName=.C1HContextID LinkType=None MultiLink=True Name=C1H Context ID Script=(None)
C1H Dropdown Text	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Text Condition= HTMLName=.C1HdropdownText LinkType=None MultiLink=True Name=C1H Dropdown Text Script=(None)
C1H Expand Text	AffectsAppearance=True Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Text Condition= HTMLName=.C1HExpandText LinkType=None MultiLink=True Name=C1H Expand Text Script=(None)
C1H Group	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Group Condition= HTMLName=.C1HGroup LinkType=None MultiLink=True Name=C1H Group Script=(None)
C1H Group Invisible	AffectsAppearance=False Hidden=True PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Group Condition= HTMLName=.C1HGroupInvisible LinkType=None MultiLink=True Name=C1H Group Invisible Script=(None)
C1H Group Link	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Group Link Condition= HTMLName=.C1HGroupLink LinkType=Group MultiLink=True Name=C1H Group Link Script=(None)
C1HIndex	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=True	Behavior=Keyword Condition= HTMLName=.C1HIndex LinkType=None MultiLink=True Name=C1H Index Script=(None)

	Display Properties	Indexing Properties	Miscellaneous Properties
C1H Index Invisible	AffectsAppearance=False Hidden=True PageRef=False Replacement= Window=(Default)	AutoIndex=True	Behavior=Keyword Condition= HTMLName=.C1HIndexInvisible LinkType=None MultiLink=True Name=C1H Index Invisible Script=(None)
C1H Inline Dropdown	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Dropdown Condition= HTMLName=.C1HInlineDropdown LinkType=None MultiLink=True Name=C1H Inline Dropdown Script=(None)
C1H Inline Expand	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Expand Condition= HTMLName=.C1HInlineExpand LinkType=None MultiLink=True Name=C1H Inline Expand Script=(None)
C1H Inline Popup	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Popup Condition= HTMLName=.C1HInlinePopup LinkType=None MultiLink=True Name=C1H Inline Popup Script=(None)
C1HJump	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Topic Link Condition= HTMLName=.C1HJump LinkType=Jump MultiLink=True Name=C1H Jump Script=(None)
C1H Keyword Link	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Keyword Link Condition= HTMLName=.C1HKeywordLink LinkType=Keyword MultiLink=True Name=C1H Keyword Link Script=(None)
C1H Link Tag	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Link Tag Condition= HTMLName=.C1HLinkTag LinkType=None MultiLink=True Name=C1H Link Tag Script=(None)
C1H Link Tag Invisible	AffectsAppearance=False Hidden=True PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Link Tag Condition= HTMLName=.C1HlinkTagInvisible LinkType=None MultiLink=True Name=C1H Link Tag Invisible Script=(None)
C1H Manual	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Conditional Text Condition=Platform=Manual HTMLName=.C1HManual LinkType=None MultiLink=True Name=C1H Manual Script=(None)

	Display Properties	Indexing Properties	Miscellaneous Properties
C1HOnline	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Conditional Text Condition=Platform=Online HTMLName=.C1HOnline LinkType=None MultiLink=True Name=C1H Online Script=(None)
C1HPopup	AffectsAppearance=False Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Topic Link Condition= HTMLName=.C1HPopup LinkType=Popup MultiLink=True Name=C1H Popup Script=(None)
C1H Popup Text	AffectsAppearance=True Hidden=False PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Inline Text Condition= HTMLName=.C1HPopupText LinkType=None MultiLink=True Name=C1H Popup Text Script=(None)
C1H Topic Properties	AffectsAppearance=False Hidden=True PageRef=False Replacement= Window=(Default)	AutoIndex=False	Behavior=Topic Properties Condition= HTMLName=.C1HTopicProperties LinkType=None MultiLink=True Name=C1H Topic Properties Script=(None)

Creating New Character Styles

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane.
3. Right-click anywhere in the character style list in the right pane and select **New** from the shortcut menu.
Note that the cursor has moved to the top of the character style list window.
4. Type the name of the new character style and press **Enter**.
Note that the new character style is added to the list.

Alternatively, you can add a new character style by clicking the **Project** menu, choosing **New**, then clicking **Character Style**.

Editing Character Style Properties

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane.
3. In the right pane, select the name of the style to be edited.
Note that the property pane changes to show the properties of the selected style.
4. Change the properties as desired.

Note that your changes affect the selected style instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

Property	Value
Display	
AffectsAppearance	False
Hidden	False
PageRef	False
Replacement	
Window	(Default)
Indexing	
AutoIndex	False
Misc	
Behavior	Conditional Text
Condition	
HTMLName	.C1HConditional
LinkType	None
MultiLink	True
Name	C1H Conditional
Script	(None)

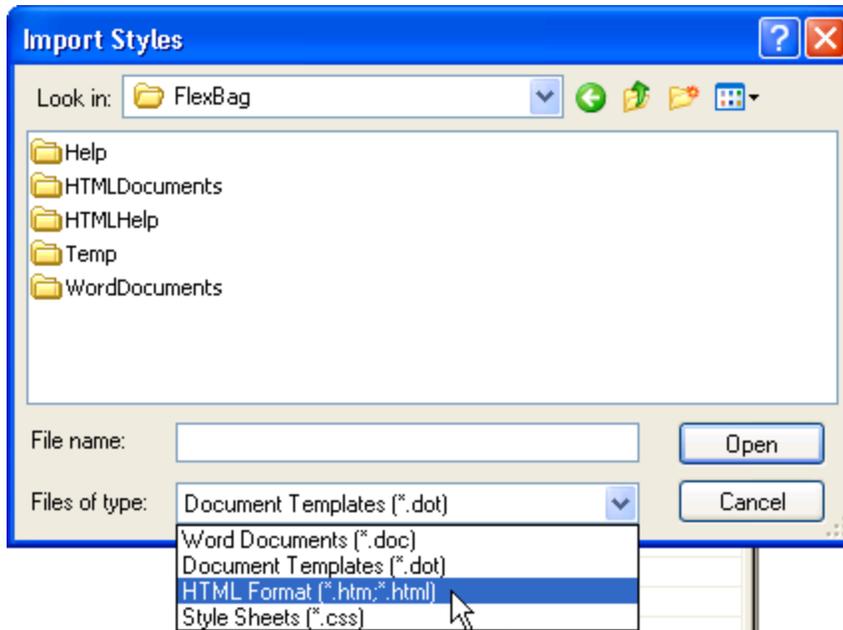
For more information, see [Using Character Styles](#) (page 216).

Importing Styles from a Word or HTML Document, Template, or Cascading Style Sheet

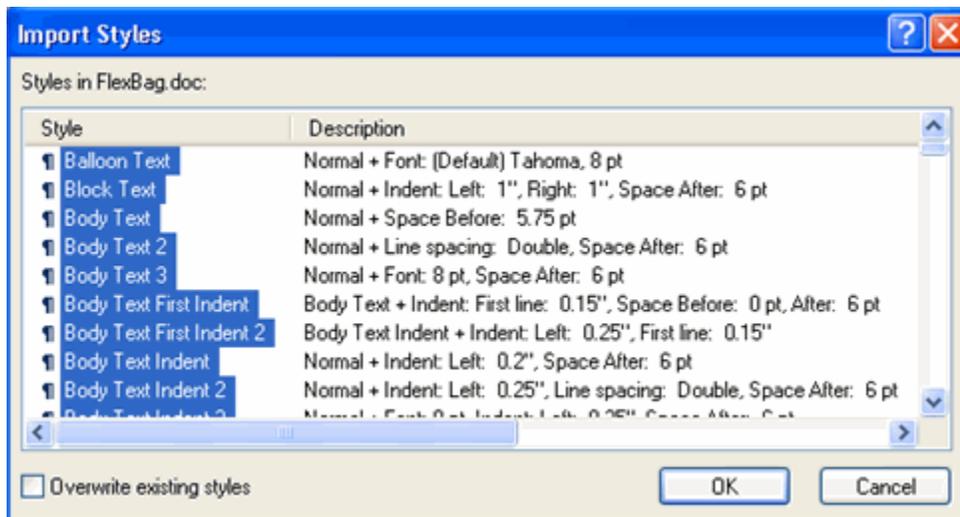
1. On the **Project** menu, click **Import Styles**. A file selection dialog box opens.
2. Specify a folder and file name, then click **Open**.

You can specify any Microsoft Word document or template, an HTML document or a style sheet; you are not restricted to those that you have added to the project. Simply change the **Files of type** drop-down to specify one of the following:

- Word Documents (*.doc)
- Document Templates (*.dot)
- HTML Format (*.htm, *.html)
- Style Sheets (*.css)



Doc-To-Help reads all style definitions from the specified file, and then displays them in a list as shown in the following figure. Initially, all styles are selected.



By default, a selected style is not imported if a like-named counterpart already exists in the active project file. If desired, check the **Overwrite existing styles** box to force Doc-To-Help to re-import existing styles.

3. Click **OK** to import all styles into the active Doc-To-Help project. Alternatively, you can select one or more styles as you would files in Windows Explorer, then click **OK**.

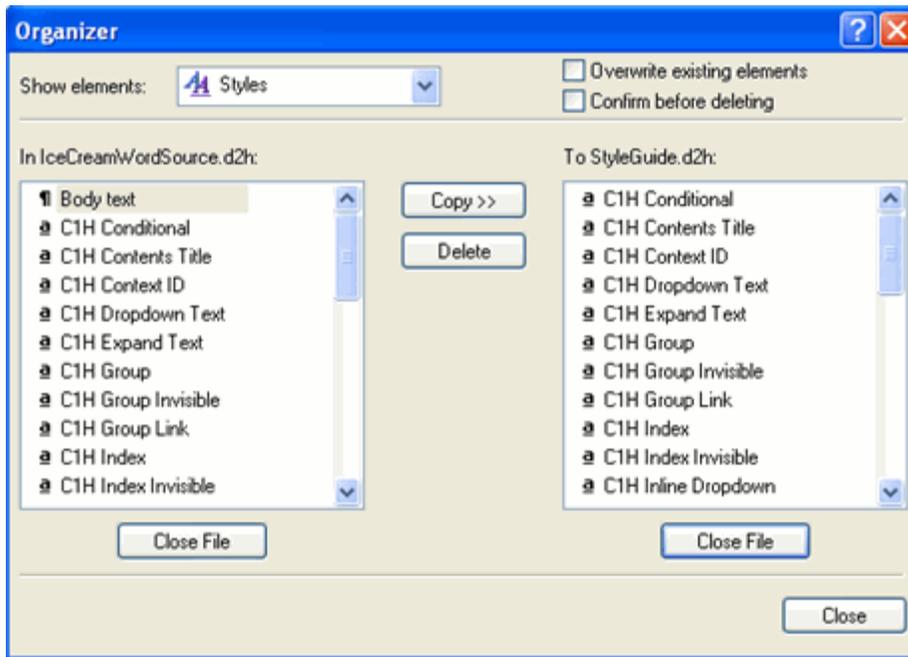
When a paragraph style is imported, it receives the same **Outline level** it had in Microsoft Word or the HTML document. When a character style is imported, its **Behavior** is set to **None**. No other settings are affected.

Importing Styles From Another Project

1. On the **Project** menu, click **Organizer**. The **Organizer** dialog box opens.
2. In the **Show elements** list, select **Styles**.
3. Click **Open File** on the right side of the Organizer to display a file selection dialog box.

- Specify a folder and file name for the existing project, then click **Open**.

The list on the right side of the **Organizer** dialog box now contains all styles in the specified project. Initially, none of the styles are selected, and the **Copy** button is unavailable.



By default, like-named styles are not copied from one project to another. If desired, check the **Overwrite existing elements** box to force the Organizer dialog box to re-import existing styles.

- Select one or more styles in the list on the right, then click **Copy**. When you are finished, click the **Close** button at the lower right corner of the dialog box.

When a style is copied from one Doc-To-Help project to another, it retains all of its attributes, including any script or window associations. If necessary, the **Organizer** dialog box also copies the appropriate script module and/or window definition. The **Overwrite existing elements** setting also applies to scripts and windows associated with a style.

Note that you can also use the **Organizer** dialog box to export styles in the active project to another Doc-To-Help project. When you select one or more styles in the list on the left, the direction of the copy operation is reversed and the list captions change accordingly.

Using Topic Types

A Topic Type is a named collection of topic attributes: what window the Help topic appears in, how it's accessed, whether it gets a map number, where the heading paragraph appears, or if it appears at all. Topics are mapped to Topic Types by the style of their headings. Using Topic Types streamlines the customizing process for you, while Helping maintain consistency. There are two uses for topic types. First, you can use a Topic Type to customize an individual topic, overriding the style properties. Second, you can use a Topic Type to customize the default behavior of a paragraph style.

Note that Topic Types and Paragraph Styles have many duplicate properties. When a paragraph style has a [TopicType](#) associated with it, the duplicate Topic Type properties supercede the Paragraph Style properties.

Topic Type Properties

Edit the Topic Type properties by accessing the property pane for the specific Topic Type. For more information, see [Editing Topic Type Properties](#) (page 225).

Display Properties

ContentsOnly	When True, omits topics defined with this Topic Type from the Help target, but uses them as book titles in the Help contents. This property is duplicated in the Paragraph Styles.
Nonscrolling	When the NonScrolling property of a Topic Type is set to True, any topic headings formatted with that topic type appear in the non-scrolling region of the Help window. This property is duplicated in the Paragraph Styles.
Popup	When the Popup property of a Topic Type is set to True, any topic formatted with that topic type appears in a pop-up window on top of the Help window. This property is duplicated in the Paragraph Styles.
Replacement	Sets or returns a string that overrides the default text generated by Doc-To-Help. For a Topic Type with AutoNext set to True, the Replacement string is used instead of the topic title next to the generated button. This property is duplicated in the Paragraph Styles.
SuppressEmptyTopics	If set to True, omits topics defined with this paragraph style or topic type that have no text except heading from online help except in TOC.
Untitled	When True, the topic title is not displayed in the Help window. This property is duplicated in the Paragraph Styles.
Window	Shows the name of the window in which topics formatted in this style are displayed. This property is duplicated in the Paragraph Styles.

Indexing Properties

AutoKeyword	When True, this property automatically creates index keywords from topic titles formatted with that topic type and associates them with the appropriate topic. This property is duplicated in the Paragraph Styles.
-----------------------------	---

Miscellaneous Properties

AutoContextID	When True, automatically creates Context IDs for all topics using this paragraph style. This property is duplicated in the Paragraph Styles.
Name	Returns a string containing the name of a specified object (read-only). This property is duplicated in the Paragraph Styles.
Script	Displays the script to be run during compilation whenever text formatted with this style is encountered. This property is duplicated in the Paragraph Styles.

Navigation Properties

AutoButtons	When True, displays subtopic buttons for the topic type with numeric outline levels greater than the parent level. This property is duplicated in the Paragraph Styles.
AutoLink	Creates unique link tags for topics formatted with the topic type by enabling the auto link tag feature. This property is duplicated in the Paragraph Styles.
AutoNavigate	Determines which topic types are included in the navigation sequence. This property is duplicated in the Paragraph Styles.
AutoNext	Enables the next topic button for any active topic type.
ExplicitAccess	When True, defines a specific topic type as accessible only through a hyperlink. Topic types defined as explicit are not accessible from the Help contents, the index list, or the text search. This property is duplicated in the Paragraph Styles.
MidTopic	Defines a topic type as an “in topic” jump location similar to an HTML anchor. By defining a topic type as MidTopic and defining a character style as a jump, end users have jump functionality within a lengthy topic. This property is duplicated in the Paragraph Styles.

Default Topic Types

Doc-To-Help comes with eight default topic types. You can modify the properties of these default topic types or create your own as needed. Adding a topic type can be achieved by right-clicking anywhere in the topic type list pane on the right side of the project editor.

	Display Properties	Indexing Properties	Miscellaneous Properties	Navigation Properties
Conceptual	ContentsOnly=False Nonscrolling=True Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=main	AutoKeyword=False	AutoContextID=False Name=Conceptual Script=(None)	AutoButtons=True AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False
Contents	ContentsOnly=False Nonscrolling=True Popup=False Replacement= SuppressEmptyTopics=True Untitled=False Window=main	AutoKeyword=False	AutoContextID=False Name=Contents Script=(None)	AutoButtons=False AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False
Glossary of Terms	ContentsOnly=False Nonscrolling=True Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=main	AutoKeyword=False	AutoContextID=False Name=Glossary of Terms Script=(None)	AutoButtons=False AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=False
Glossary Term Definition	ContentsOnly=False Nonscrolling=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=False Window=(Default)	AutoKeyword=False	AutoContextID=False Name=Glossary Term Definition Script=D2HGlossaryTerm	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False
Margin Note	ContentsOnly=False Nonscrolling=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=True Window=(Default)	AutoKeyword=False	AutoContextID=False Name=Margin Note Script=(None)	AutoButtons=False AutoLink=False AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False
Procedural	ContentsOnly=False Nonscrolling=False Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=proc	AutoKeyword=False	AutoContextID=False Name=Procedural Script=(None)	AutoButtons=True AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=False MidTopic=False
Sub-Contents	ContentsOnly=True Nonscrolling=True Popup=False Replacement= SuppressEmptyTopics=False Untitled=False Window=main	AutoKeyword=False	AutoContextID=False Name=Sub-Contents Script=(None)	AutoButtons=False AutoLink=True AutoNavigate=True AutoNext=False ExplicitAccess=False MidTopic=False
What's This	ContentsOnly=False Nonscrolling=False Popup=True Replacement= SuppressEmptyTopics=False Untitled=True Window=(Default)	AutoKeyword=False	AutoContextID=False Name=What's This Script=(None)	AutoButtons=False AutoLink=True AutoNavigate=False AutoNext=False ExplicitAccess=True MidTopic=False

Adding a New Topic Type

To add a new topic type to the default list:

1. On the Icon bar, click **Project**.
2. Select the **Topic Types** item from the left pane.
3. Right-click anywhere in the Topic Type list in the right pane and select **New** from the shortcut menu. Note that the cursor has moved to the top of the Topic Type list window.
4. Type the name of the new Topic Type and press **Enter**. Note that the new topic type is added to the list.

Alternatively, you can add a new topic type by clicking the **Project** menu, choosing **New**, then clicking **Topic Type**.

Editing Topic Type Properties

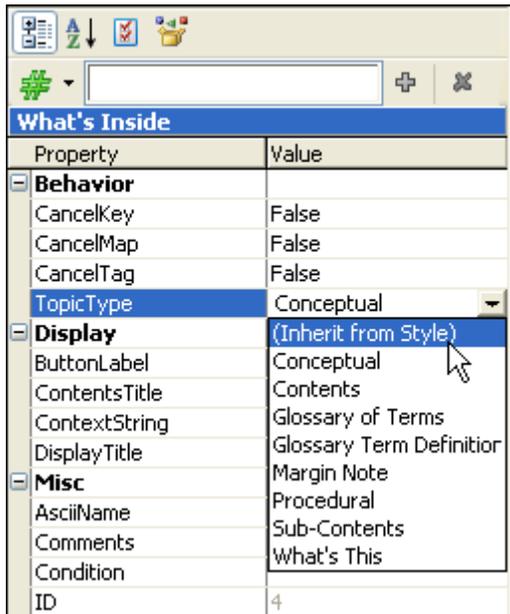
1. On the Icon bar, click **Project**.
2. Select the **Topic Types** item from the left pane.
3. In the right pane, select the name of the Topic Type to be edited. Note that the property pane changes to show the properties of the selected style.
4. Change the properties as desired. Note that your changes affect the selected Topic Type instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

Conceptual	
Property	Value
Display	
ContentsOnly	False
Nonscrolling	True
Popup	False
Replacement	
SuppressEmptyTopics	False
Untitled	False
Window	main
Indexing	
AutoKeyword	False
Misc	
AutoContextID	False
Name	Conceptual
Script	(None)
Navigation	
AutoButtons	True
AutoLink	True
AutoNavigate	True
AutoNext	False
ExplicitAccess	False
MidTopic	False

Applying Topic Types to Topics

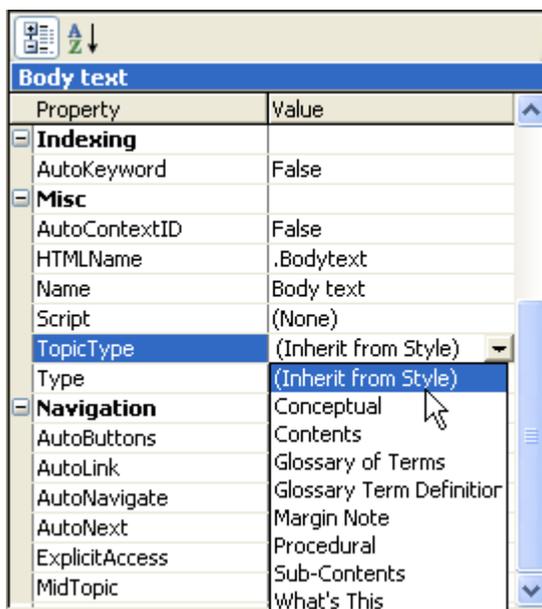
1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane.
3. In the right pane, select the name of the topic to be edited. Note that the property pane changes to show the properties of the selected topic.

- In the property pane, use the [TopicType](#) property drop-down menu to select the Topic Type you wish to associate with the specific topic. Note that your changes affect the selected topic instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.



Applying Topic Types to Paragraph Styles

- On the Icon bar, click **Project**.
- Select the **Paragraph Styles** item from the left pane.
- In the right pane, select the heading to be edited. Note that the property pane changes to show the properties of the selected style.
- Use the [TopicType](#) property drop-down menu to select the Topic Type you wish to associate with the specific Paragraph Style. Note that your changes affect the selected topic type instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.



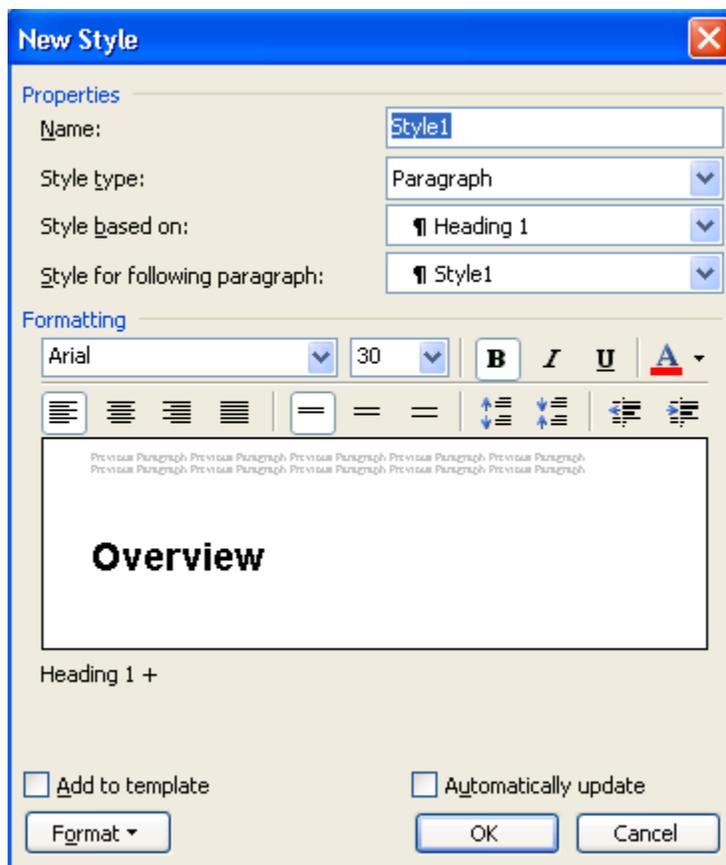
Working With Styles in Word

In Microsoft Word, you can use styles to apply paragraph formats (such as indentation and spacing) and character formats (such as font styles and colors) to give your documents a uniform appearance. If you apply styles consistently, any change you make to a style definition is propagated throughout the document, and the affected regions are updated accordingly. Styles are also used in Word's outline view to specify the hierarchy of paragraphs within a document.

Doc-to-Help extends Word's concept of styles to include Help authoring behavior. For each paragraph style that begins a topic, you define a like-named paragraph style in the project file. Similarly, for each character style that references a topic, you define a like-named character style in the project file. By customizing the properties of Doc-To-Help styles, you define the behavior of the compiled Help system.

Creating Styles in Word

1. From the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** window opens.
2. Click **New Style**. The **New Style** dialog box opens.

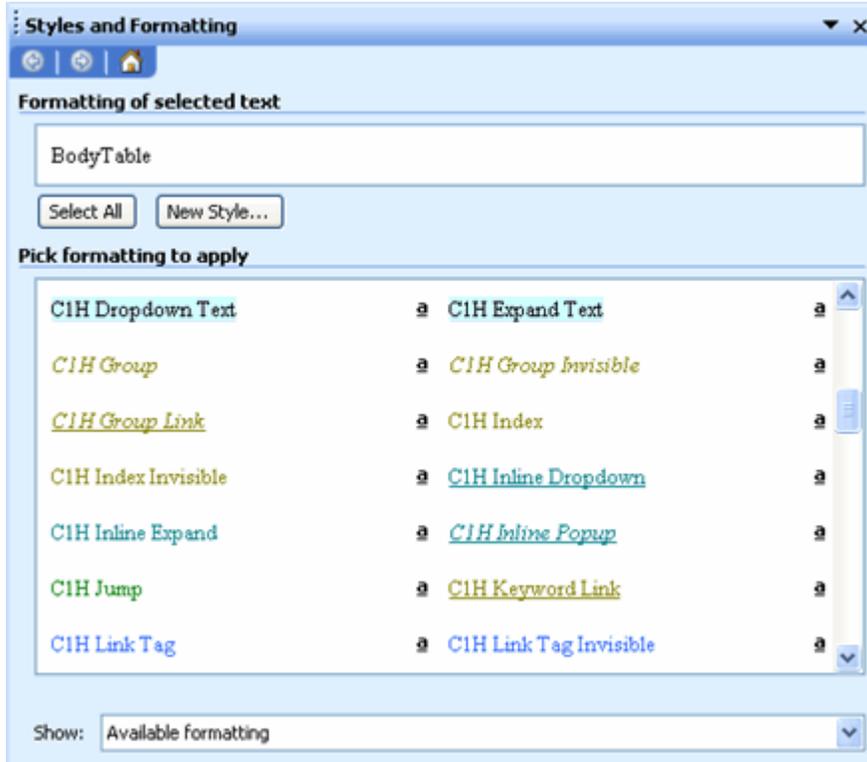


3. In the **Name** box, type the name of the new style.
4. In the **Style type** box, select **Paragraph** or **Character**, depending upon the type of style you want to create.
5. If desired, change the other options on this dialog box, then click **Format** to change the visual attributes of the new style.
6. When you are finished, click **OK**.

Applying Paragraph Styles in Microsoft Word

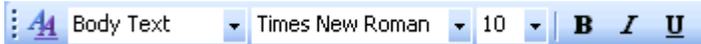
1. Click anywhere within the paragraph to be formatted. You do not need to select the entire paragraph, although doing so will also work.

- From the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** window opens.



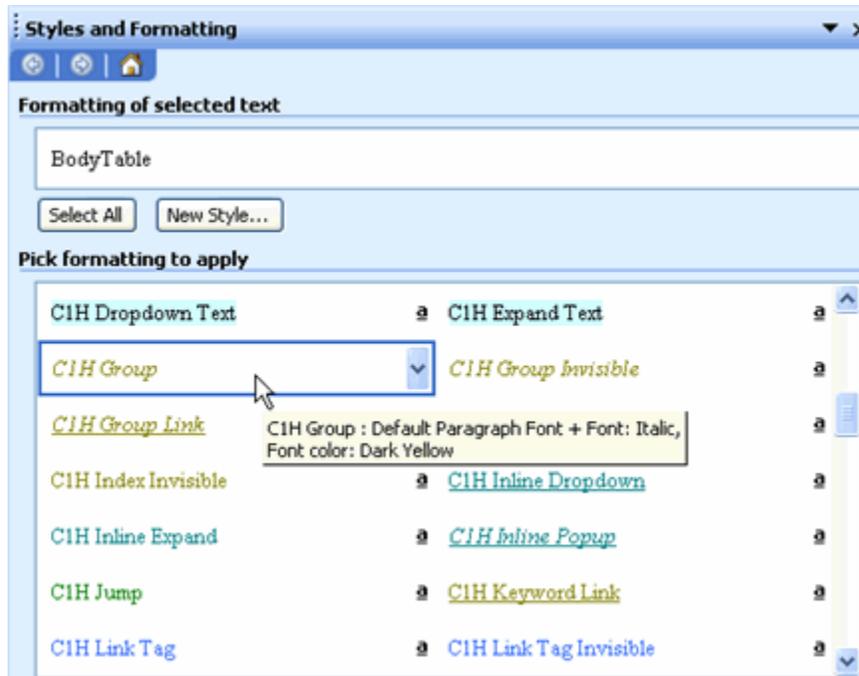
- In the **Styles and Formatting** window, click the name of the paragraph style to be applied. If the desired style is not listed, select another group of styles from the **Show** drop-down list. Paragraph styles are marked with the ¶ symbol.

Alternatively, you can apply the desired style directly by selecting it in the **Style** list of the **Formatting** toolbar.



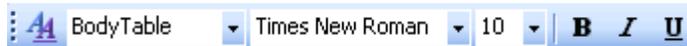
Applying Character Styles in Microsoft Word

- Select the text to be formatted. If only a single word is to be formatted, you can click anywhere within the word.
- From the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** window opens.



3. In the **Styles and Formatting** window, click the name of the character style to be applied. If the desired style is not listed, select another group of styles from the **Show** drop-down list. Character styles are marked with the  symbol.

Alternatively, you can apply the desired style directly by selecting it in the **Style** list of the **Formatting** toolbar.



Using List Styles in Microsoft Word

Doc-To-Help provides two ways to add lists to your Help targets: Microsoft Word lists and the list style buttons.

For those of you familiar with making lists in Microsoft Word, you will be happy to know that Doc-To-Help accepts all numbered and bulleted lists. Because Doc-To-Help will accept any number of levels or styles, using the Word list functionality is the recommended method when working with documents that require multiple bullet or number styles or more than a few levels.

Alternately, when working with simple lists, there are six list styles available on the **Doc-To-Help** toolbar for Microsoft Word. These list style buttons allow you to create numbered and bulleted nested lists quickly and easily.



C1H Number Style Numbers the specified text.



C1H Number 2 Style Numbers and indents the specified text to nest it within a list.



C1H Bullet Style Formats the specified text with a bullet.



C1H Bullet 2 Style Bullets and indents the specified text to nest it within a list.



C1H Bullet 2A Style Bullets and indents the specified text to nest it within a list.



Continue List Formats the specified text as an unnumbered list at the same level as the previous numbered list.

Creating Lists with Doc-To-Help List Styles

1. On the Icon bar, click **Project**.
2. In the left pane, select **Documents**.
3. Select and right-click a document from the right pane, and select **Open** from the shortcut menu.
4. Highlight the text in your document to which you want to apply a list style.
5. On the **Doc-To-Help** toolbar, click one of the style buttons.

Alternatively, you can apply the desired style directly by selecting it in the **Styles** and **Formatting** list of the **Formatting** toolbar or from the **Format | ComponentOne Doc-To-Help Styles** menu command.

For more information, see [Applying Lists in Doc-To-Help](#) (page 90) and [Working With Styles in Word](#) (page 227).

Nesting Lists with Doc-To-Help List Styles

1. On the Icon bar, click **Project**.
2. In the left pane, select **Documents**.
3. Select and right-click a document from the right pane, and select **Open** from the shortcut menu.
4. Highlight the text in your document to which you want to apply the bulleted list style.
5. Click the **C1H Bullet 2 Style** button .

Doc-To-Help applies a nested bulleted list.

Alternatively, you can apply the desired style directly by selecting it in the **Style** list of the **Formatting** toolbar or from the **Format | ComponentOne Doc-To-Help Styles** menu command.

For more information, see [Applying Lists in Doc-To-Help](#) (page 90) and [Working With Styles in Word](#) (page 227).

Using Lists in the Printed Manual Target

Microsoft Word has a number of known issues concerning the preservation of numbered list formatting when copy/paste and other text manipulations are performed. These problems are generally restricted to Word 2000 and are largely solved in Word 2002 (XP). They originate in the sophisticated manner in which Word tries to make numbered list behavior as smart and intuitive as possible. As a result, numbered lists are sometimes re-numbered and re-formatted incorrectly when you perform copy/paste and other modifications.

For this same reason, the numbering and formatting of some lists can be broken when compiling the master document (print target) with the [Concatenate](#) property is set to **True**. This problem is impossible to predict and may be difficult to find, so when the [FixLists](#) property is set to default (Word 2000 Only), Doc-To-Help automatically fixes, re-numbers and re-formats lists where it detects this problem in the resulting document.

However, on very rare occasions, when Doc-To-Help attempts to fix this problem, it can cause Word to crash, freeze or corrupt the resulting document. In these rare cases, if you encounter such problems while the Doc-To-Help progress indicator reads "Restoring lists formatting...", you may want to make Doc-To-Help skip this action.

Set the **FixLists** property to **Never** if you want to skip the fix for Word 2000.

Set the **FixLists** property to **Always** if you want to force the fix for all versions of Word.

Using HTML Styles in a Doc-To-Help Project

Using styles for help markup has always been the central feature of Doc-To-Help. As Word styles are used in Word documents, Cascading Style Sheet (CSS) styles, or *classes* in HTML, are used in HTML documents. CSS styles can be used to format text for appearance only in a help file, or they can have special meaning in Doc-To-Help when it is building help targets. CSS styles used for appearance only can have any names defined in the style sheet, but the

special-meaning CSS styles must be identified with corresponding Word styles having the same meaning. Since style name conventions in CSS and Word differ, for example, CSS does not allow spaces in style names, and since there are two kinds of style *selectors* in CSS, class and tag, a special mechanism is needed to provide one-to-one correspondence between Word styles and CSS styles. This mechanism is the Doc-To-Help property `Style.HTMLHelp`.

Every special-meaning style specified in a Doc-To-Help project has two names: (1) the `Style.Name` property that identifies it as a Word style or a Style object in Doc-To-Help if the project does not use Word, and (2) the `Style.HTMLName` property that identifies it as a CSS style, or selector.

Note: When you use a style in an HTML Document and that style is defined in your Doc-To-Help project, make sure you use the HTML name of the style specified in the `Style.HTMLName` property, not the `Style.Name` property.

According to the standard CSS rules, `Style.HTMLName` can have one of three forms:

- (1) `.<stylename>` - The style can be used with any HTML tag.
- (2) `<tag>` - The specified tag, for example, H1 for style Heading 1 is considered by Doc-To-Help as having this style, even if the user did not format this tag with any particular CSS style.
- (3) `<tag>.<stylename>` - The style can be used only with the specified tag; it is ignored if used in other tags.

The following are some examples of the Word style names and their corresponding CSS style names:

Style.Name Property	Style.HTMLName Property
Body text	.Bodytext
Heading 1	H1

Styles included by default in a Doc-To-Help project already have the appropriate `HTMLName`. If you add new styles to a project, the default `HTMLName` assigned to it is “`.stylename`” where `Style.Name = stylename`. Note that spaces and non-ASCII characters are removed. It means that this style can be applied to any tag in HTML. If you want to restrict the use of this style to certain tags, or if you want to associate this style with a tag in the same way that styles Heading 1,2,... are associated with tags H1, H2,..., you can change the value of the `HTMLName` property.

Working with Cascading Style Sheets and Word Documents

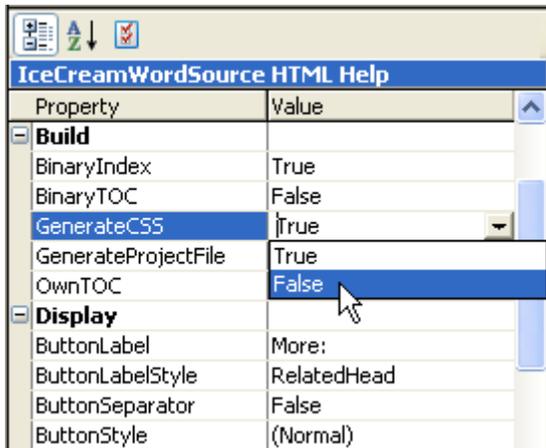
When compiling any HTML based Help target and using Word source documents, Doc-To-Help produces a Cascading Style Sheet file (.css) based on the template that is attached to your Help target. This file governs the appearance of your HTML Help targets in much the same way as a Microsoft Word template governs the appearance of a Word document. In most cases, there is no need to do anything to the CSS file, Doc-To-Help modifies it as needed during each compilation of the Help target. This file is produced only if Word documents and templates are used in your project.

Modifying the Cascading Style Sheet

In some cases the Help author may wish to modify the CSS file or even replace it with a customized CSS file of their own. To make sure that Doc-To-Help does not overwrite the modified CSS file, simply set the `GenerateCSS` property to **False**.

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. In the right pane, select the Help target to be edited.

4. In the Property Pane, change the [GenerateCSS](#) property **False**.



Note: Cascading Style Sheets are only used when creating HTML output.

Doc-To-Help Markup Language (D2HML)

D2HML is a collection of styles and rules for formatting text that allows you to author your documentation solely in Word or the HTML editor of your choice without even having to open Doc-To-Help, although having Doc-To-Help open makes formatting with D2HML styles and dialog boxes even more convenient. With D2HML, you simply format your text with a D2HML style, and, optionally, attach additional information to it as you type. It's that easy! Your source files contain nothing more than pure text formatted with styles. No D2H-proprietary information is stored in these documents.

This Doc-To-Help innovative technology is intended to automate help creation. While some help-authoring tools help you produce help files out of your documentation, Doc-To-Help frees you from the time-consuming process of both writing the documentation and preparing it for use with a specific help-authoring tool at the same time. With D2HML, you do not need to master a help-authoring tool or tie your documents to one. You simply write your source documents in Word, HTML or both, and follow a few simple rules for formatting them with D2HML styles.

D2HML Styles

Doc-To-Help 2006 comes with a set of pre-defined D2HML styles that can be used to format your source documents. These styles allow you to create jumps, keywords, groups, conditional text and much more. D2HML even allows you to work without opening Doc-To-Help, if you so choose. While this is the main advantage of using D2HML, it is highly recommended that Doc-To-Help remains open when using Word, FrontPage and Dreamweaver to edit your source documents. This allows you to take advantage of the **D2HML Styles** toolbar and dialog boxes provided with Doc-To-Help, making formatting as simple as clicking a button. The functionality of the dialog boxes is limited if the Doc-To-Help project is not open.

The D2HML styles are defined in Doc-To-Help's accompanying templates and cascading styles sheets, which can be attached to your Word and HTML source documents, respectively.

When you apply a D2HML style to text, a *hot spot* is created. There are different types of hot spots, each with different Behavior values that tell the hot spot what to do. Each hot spot type also corresponds with a Word style and an HTML class.

For additional information on the hot spot types and pre-defined D2HML styles, please see the [Hot Spot Types](#) (page 252) section.

Exploring the D2HML Styles Toolbar

The **D2HML Styles** toolbar allows you to easily apply a D2HML style to text. Select the desired text and click a D2HML toolbar button to format it with a D2HML style. A dialog box appears when you click the toolbar buttons, which allows you to define the hot spot even further, if necessary, with only a few clicks.

The **D2HML Styles** toolbars in Word, FrontPage and Dreamweaver are slightly different from each other.

For more information on the link styles and dialog boxes, see [Hot Spot Types](#) (page 252).



D2HML Styles Toolbar in Microsoft FrontPage



D2HML Styles Toolbar in Macromedia Dreamweaver



D2HML Styles Toolbar in Microsoft Word

	Jump Topic Link	Format text with C1H Jump style.
	Popup Topic Link	Format text with C1H Popup style.
	Keyword Link	Format text with C1H Keyword Link style.
	Group Link	Format text with C1H Group Link style.
	Glossary Link	Format text with C1H Popup style.
	Inline Expand	Format text with C1H Inline Expand style.
	Inline DropDown	Format text with C1H Inline Dropdown style.
	Inline Popup	Format text with C1H Inline Popup style.
	Keyword	Format text with C1H Index style.
	Invisible Keyword	Format text with C1H Index Invisible style.
	Group	Format text with C1H Group style.
	Invisible Group	Format text with C1H Group Invisible style.
	Link Tag	Format text with C1H Link Tag style.
	Invisible Link Tag	Format text with C1H Link Tag Invisible style.
	Context ID	Format text with C1H Context ID style.
	Contents Title	Format text with C1H Contents Title style.
	Topic Properties	Format text with C1H Topic Properties style.
	Online Text	Format text with C1H Online style.
	Manual Text	Format text with C1H Manual style.
	Conditional Text	Format text with C1H Conditional style.
	Clear Formatting	Removes any D2HML formatting, returning the text to the style it was before it was formatted in D2HML.
	Clear Condition	Clears conditional text styles while leaving other formatting intact. Located on the FrontPage and Dreamweaver D2HML Styles toolbar only.
	Apply Style	Formats any selection with any of the D2HML styles.

**Show Hidden Hot Spots**

Shows any invisible hot spots. Located on the Word and Dreamweaver **D2HML Styles** toolbar only.

**D2HML Options**

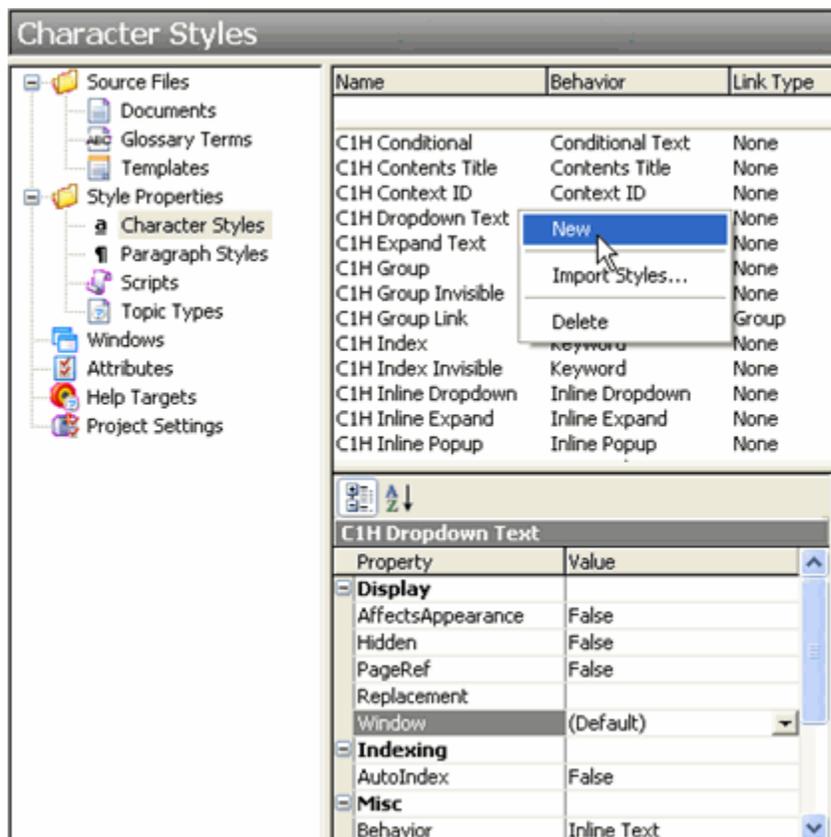
Determines whether conditional text is formatted with a comment or a style. Located on the Word toolbar only.

Doc-to-Help comes with a set of predefined character styles, but you can also add your own. Simply add a character style to the Doc-To-Help project and specify its Behavior property. For more information on adding D2HML character styles, see [Creating a D2HML Style](#) (page 235).

Note: A character style included in a Doc-To-Help project is used in building a help target if its CharacterStyle.[Behavior](#) property is set to a value other than **None**.

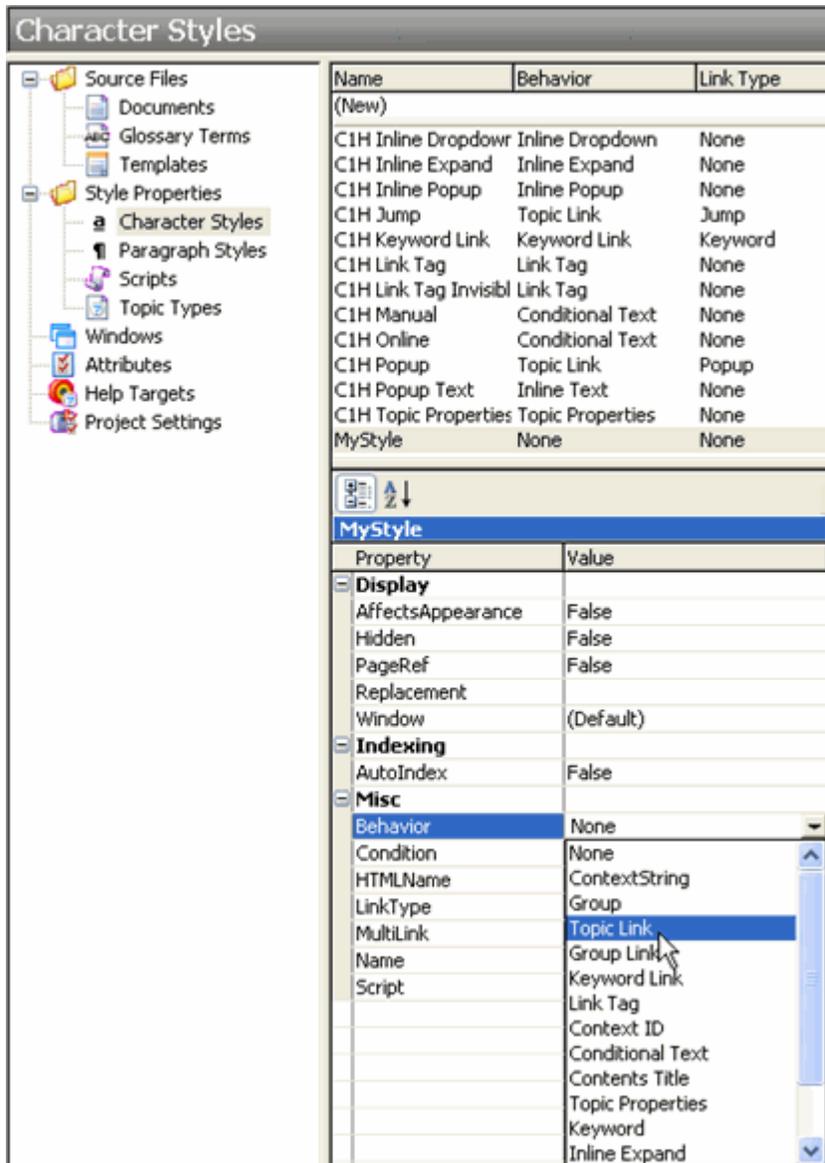
Creating a D2HML Style

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane.
3. Right-click anywhere in the character style list in the right pane and select **New** from the shortcut menu.



Note that the cursor moves to the top of the character style list window.

4. Type the name of the new character style and press **Enter**. The new character style is added to the list.
5. Under **Misc**, click the drop-down arrow next to the **Behavior** property and select a behavior from the list.

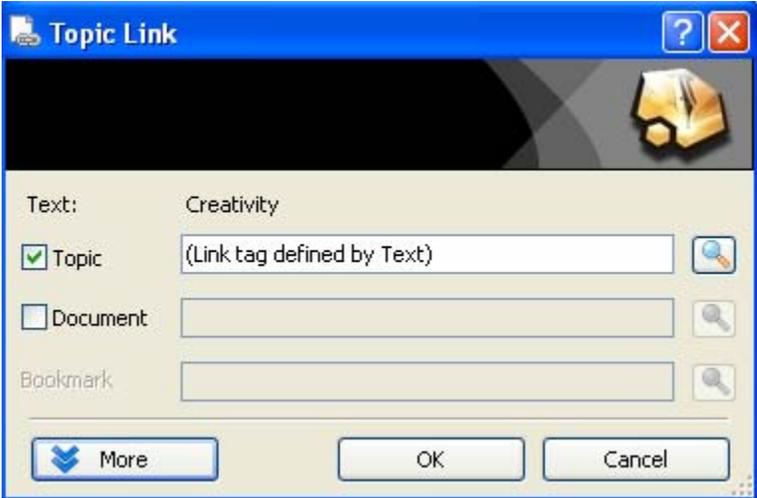


Using D2HML Dialog Boxes

Doc-To-Help simplifies the formatting process by providing the D2HML Styles dialog boxes when using the **D2HML Styles** toolbar in Microsoft Word or FrontPage and Macromedia Dreamweaver. Each button on the toolbar corresponds with a dialog box that appears when the button is clicked, allowing you to set the properties of the hot spot.

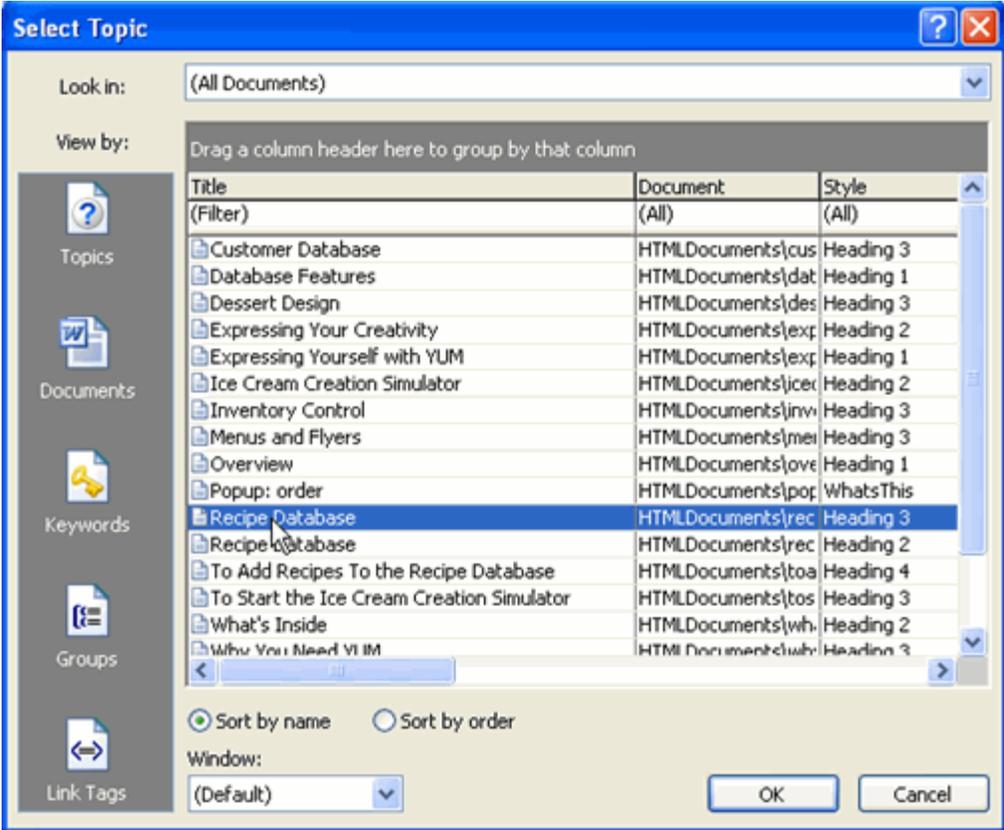
The following example shows you how to use the **Topic Link** dialog box:

1. Select some link text in your source document and click the **Jump Topic Link** button. The **Topic Link** dialog box appears:



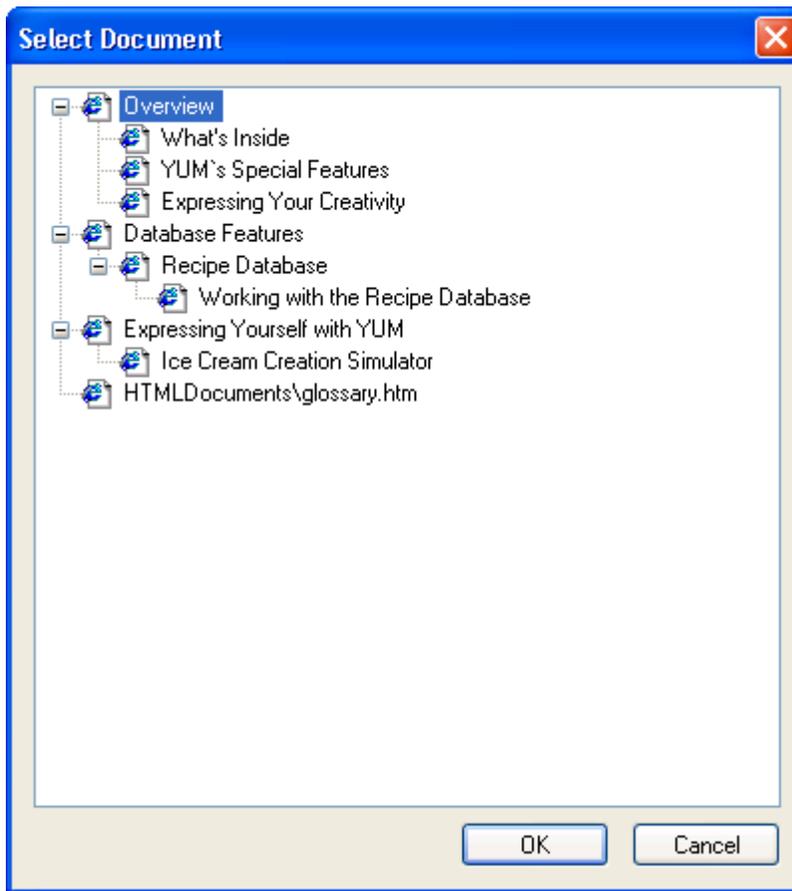
The Link dialog boxes, including **Topic Link**, **Popup Link**, **Keyword Link** and **Group Link**, allow you to browse for an existing topic, keyword or group to which to link your hot spot.

- 2. Click the **Browse** button . Note that the Doc-To-Help project containing this document must be opened in order to use this button. The **Select Topic** dialog box opens. The selection dialog box that appears may be different from the one below, depending on which Link dialog box you are using.

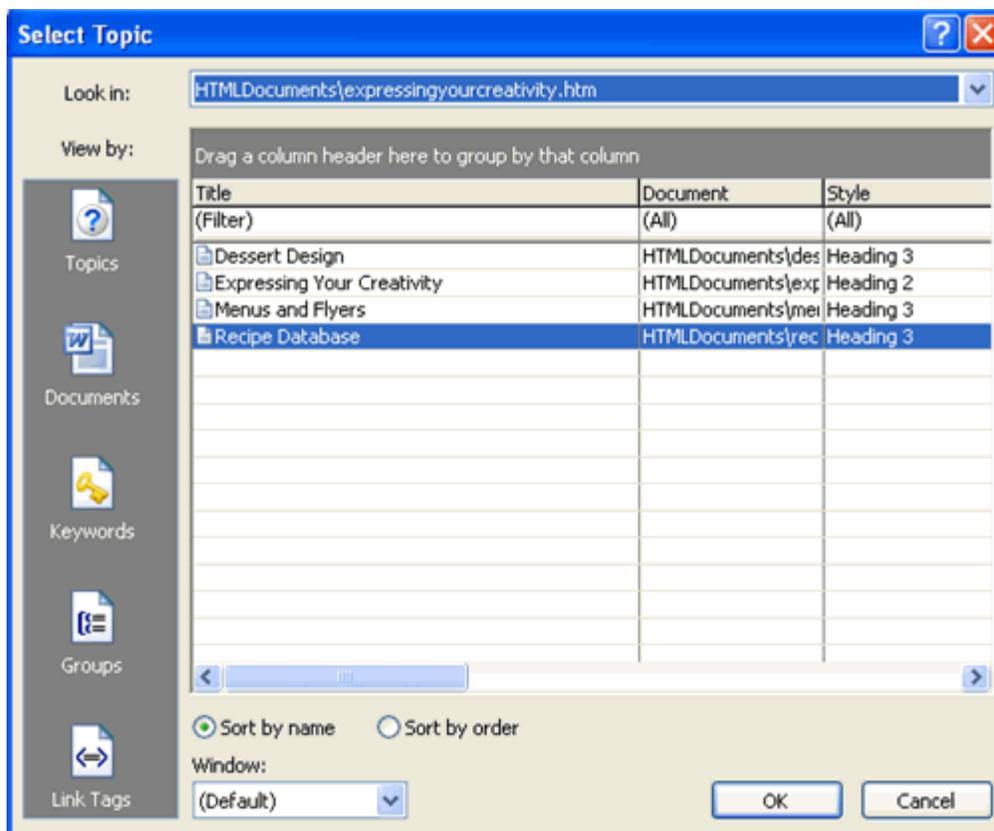


Note: If the text you have selected for the hot spot is an exact textual match of the topic, keyword or group that you want to link to, you do not have to browse to find the topic. You can just click **OK**, and Doc-To-Help creates the link. For example, suppose you want to link to the topic *Topic1* in your documentation from the hot spot text *topic1*. Since the hot spot text matches the topic text, you can simply click **OK** in the **Topic Link** dialog box without having to browse for the *Topic1* topic.

3. Select a topic from the list, or use the **Look in** drop-down box to browse the source documents and filter the available topics by document. For this example, click the **Look in** drop-down box and select **(Choose Documents)**. The **Select Document** window opens.



4. Select **Expressing Your Creativity**. Now, only the topics within the **expressingyourcreativity.htm** document appear in the **Select Topic** window. Note that if you want to switch back to view all topics in all the documents, select **(All Documents)** from the **Look in** drop-down box.



5. Select the **Menus and Flyers** topic and click **OK**. The topic appears in the **Link To** text box in the **Topic Link** dialog box.
6. Most dialog boxes have a **More** button  that expands, allowing you to set additional options and properties. Click **More**, and select *proc* from the **Window** drop-down box in the **Topic Link** dialog box. For additional information on the **More** options and properties, see [Hot Spot Types](#) (page 252).
7. Click **OK**.

When the help file is built and the link text is clicked, the Menus and Flyers topic appears in the procedural window.

It is not required that you use the **Browse** button and selection dialog box to define your hot spot. These options are available for your convenience and are not included in all dialog boxes. You can manually enter the topic to link to in the **Link To** area of the dialog box.

See [Hot Spot Types](#) (page 252) for additional information on each hot spot's properties and dialog box. Please note that some dialog boxes look slightly different from each other in Word, FrontPage and Dreamweaver.

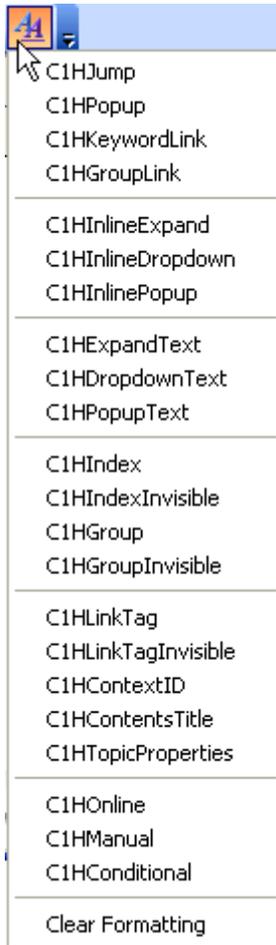
Formatting D2HML without Dialog Boxes

You are not required to use the dialog boxes when using the **D2HML Styles** toolbar in Word, FrontPage and Dreamweaver. They are there for your convenience. If you want to apply a style to selected text without having to open the D2HML dialog box:

- Hold down the **Control** key while clicking a button from the **D2HML Styles** toolbar.

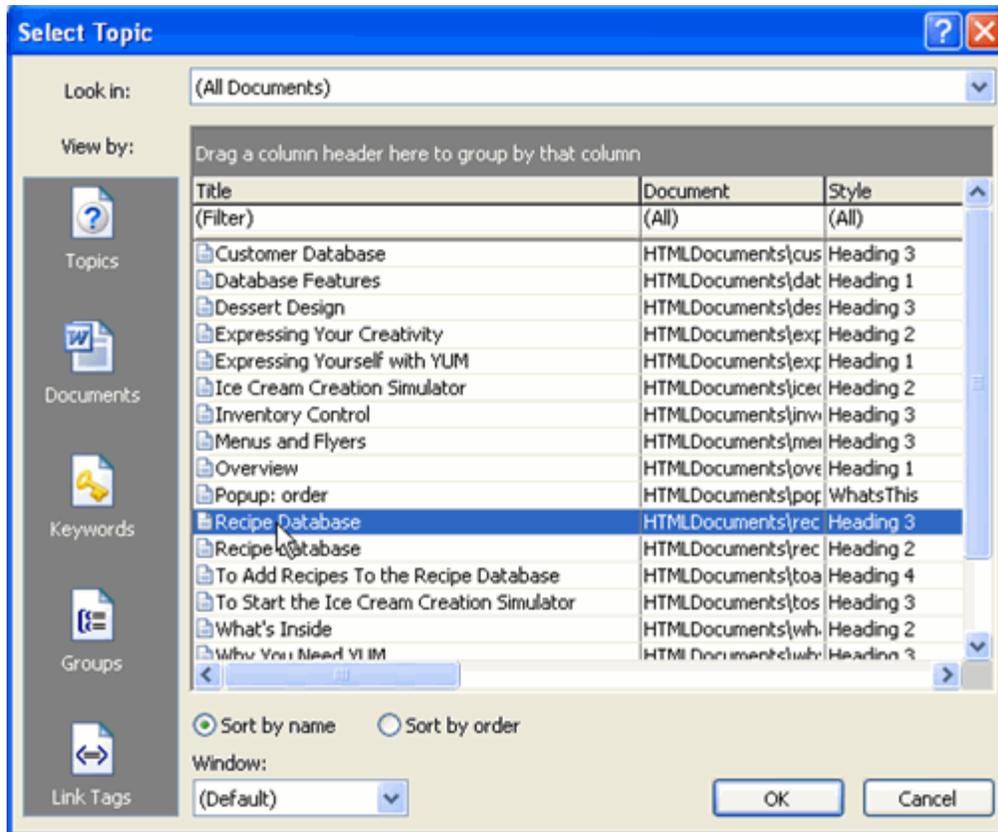
OR

- Click the **Apply Style** button  on the **D2HML Styles** toolbar and select a style from the menu.



You also have the option of opening a selection dialog box, such as **Select Topic**, without having to go through the **D2HML** dialog box. Hold down the **Shift** key and press one of the D2HML style buttons from the toolbar. For example, to open the **Select Topic** button while applying the C1HJump style:

- Select some link text in your source document.
- Hold down the **Shift** key and click the **Jump Topic Link** button. The **Select Topic** dialog box appears:



You do not need to go through the **Topic Link** dialog box to select the topic to which you want to link.

Note: This option is available when using the following **D2HML Style** toolbar buttons: **Topic Link**, **Popup Link**, **Keyword Link**, **Group Link** and **Conditional Text**.

D2HML Formatting Rules

Hot spot text consists of the **primary** part (left) and an optional **secondary** part (right); each part is separated by a separator bar (|).

- **Primary part:** This is the text that will be shown in the target. This part is the main, default source of information specifying what the hot spot should do. It contains plain text and no properties.

Note: When the primary part is formatted with a style with the **Hidden** property set to **True** in the Doc-To-Help project, it will not be visible in the target.

- **Secondary part:** This text will not be visible in the target. This part specifies the property values of the hot spot. The secondary part is optional and used only in advanced cases. It consists of one or more property settings of the form “<property name>=<value>”. The properties are separated by a semicolon (;).

Here are some examples of both primary and secondary parts:

This is a link to [My Topic](#).

My Topic is the primary part of the hot spot. There is no secondary part. Doc-To-Help creates a link to the topic **My Topic**.

This is a link to [My Topic | window=proc](#).

My Topic is the primary part of the hot spot. *Window=proc* is the secondary part, which contains one property and one property value. The property is *window*, and the property value is *proc*. Doc-To-Help creates a link to the topic **My Topic**, which appears in the procedural window.

This is a link to [My Topic|tag=Topic;window=proc](#).

My Topic is the primary part of the hot spot. *Tag=Topic;window=proc* is the secondary part, which contains two properties and two property values. The first property is *Tag*, which is a primary property, and its value is *Topic*. The second property is *window*, and its value is *proc*. Doc-To-Help creates a link to a different topic titled **Topic**, which appears in the procedural window.

Both the primary and secondary parts of the hot spot must be formatted with a D2HML style in order for the hot spot to work correctly.

Reserved Characters:

The following characters are reserved for use with D2HML:

- | Separates the primary part from the secondary part.
- ; Separates properties in the secondary part.
- = Separates the property name and the property value.
- + Separates values in the list of a collection property.
- , Separates parent and child keywords. Used only in the **Keyword** property value.
- : Separates the attribute name and the attribute value. Used only in the **Attribute** property value.

Escaping Rules for Reserved Characters:

If you want to use the reserved character ‘|’ in your primary part text and suppress its reserved meaning of ending the primary part, escape it with a backslash: ‘\|’. The escaping backslash will be removed and the vertical line character preserved in the target as normal text. For example, the following hot spot formatted with the C1H Jump style

[File\|Open|tag=File_Open](#)

will create a link in the target with text File|Open.

All remaining reserved characters except the vertical line (semicolon, equality, plus, comma and colon) have special meaning in the secondary part. If you want to use them in the secondary part of a hot spot as simple text without special meaning, escape them with backslash. For example, the following hot spot formatted with the C1H Index style

[Text|keyword=low+high](#)

creates two keywords ‘low’ and ‘high’, but

[Text|keyword=low\+high](#)

creates one keyword ‘low+high’.

Note that since there is only one reserved character in the primary part, the vertical line, escaping of other characters is not supported in the primary part, so the following hot spot formatted with the C1H Index style

[low\+high](#)

creates a single keyword ‘low\+high’.

Hot Spot Properties

Each hot spot type supports its own set of properties. These properties are defined by specific characteristics that determine how each property can be used. For example, the **ContextID** property of the Context ID hot spot type is a collection property. It accepts multiple values, allowing you to specify more than one context ID for a topic. Other properties, such as the **Tag** property of the Topic Link hot spot type, are simple properties, accepting only one value.

Below is a description of the types of properties and their characteristics.

Primary and Secondary Properties

We distinguish between two kinds of properties: **primary** and **secondary**. Primary properties are used to override the hot spot meaning of the primary part. Secondary properties simply provide additional information for the hot spot; they do not affect the meaning of the primary part.

For example, the Topic Link hot spot type has a primary property **Tag**, which is used to specify the target of the link. The following Topic Link hot spot does not use the primary property **Tag**; therefore, the **primary** part defines what the hot spot does:

This is a link to [My Topic Link](#).

This hot spot jumps to the topic *My Topic Link*.

But suppose you would like the link to jump to a topic other than *My Topic*. You can use the primary property **Tag** to specify a different target:

This is a link to [My Topic Link|tag=My Topic](#).

This hot spot sets the primary property **Tag** to *My Topic*; therefore, the **primary** part of the hot spot, *My Topic Link*, is ignored. Once a primary property is introduced, the **primary** part of the hot spot only defines the text that is shown in the target. The hot spot action is then determined by the value of the primary property, or **Tag**, in this case. This link jumps to the topic *My Topic* and not to *My Topic Link*.

If you were to add the Topic Link hot spot type's secondary property **window** to the hot spot, this property does not affect the meaning of the primary part, it only tells the link to open in a specific window:

This is a link to [My Topic Link|window=proc](#).

This hot spot jumps to the topic *My Topic Link*, and it appears in the window named *proc*.

Note: Both primary and secondary properties appear in the secondary part of the hot spot. Each hot spot type has at least one primary property, but some may have more than one.

Some hot spot types have more than one primary property. Multiple primary properties are prioritized, one primary property takes precedence over another. If multiple primary properties are defined in the same hot spot, only the highest prioritized primary property takes effect; the others are ignored.

The order of property/value pairs in the secondary part of the hot spot is unimportant.

Default Properties

Almost every hot spot type has a default property. The name of a default property can be omitted from the secondary part of a hot spot. The corresponding '=', however, must always be present, even for a default property. For example, here is a topic link with the *Tag* property set to a specific topic:

Here is a [Topic Link|tag=My Topic](#).

Since *Tag* is the default property of this hot spot type, the topic link can be also be defined as:

Here is a [Topic Link|=My Topic](#).

Either way is correct.

Simple and Collection Properties

A property can be a simple property, where each value is a single value, or a collection property, which can have a single value or a list of values.

The primary property **Tag** used in the examples above is an example of a simple property. You may specify only one topic as a target of a link. **Tag** does not accept multiple values.

A collection property value is a plus-separated list of values: “value1+value2+value3”. The **ContextID** property of the Context ID hot spot type is an example of a collection property, because it allows you to set multiple context IDs for one topic. For example:

My Topic|contextid=3+6

My Topic is assigned two context IDs, 3 and 6.

If a hot spot type supports a primary collection property, then its primary part is processed, only if this property is not set in the secondary part, according to the same rules as the primary collection property. In other words, it is also treated as a semicolon-separated list. The previous example can be rewritten as:

3+6

The primary part is **3+6**, and the primary property, **contextid**, is a collection property; therefore, **3+6** is processed as a list. The topic containing this hot spot is assigned two context IDs, 3 and 6.

Case-sensitive and Case-insensitive Properties

Property names are case-insensitive. D2HML recognizes property names written in either upper or lower case. Property values may be case-sensitive or case-insensitive, depending on the property. See each property under the [Hot Spot Types](#) (page 252) to determine its case-sensitivity.

For additional information on the different hot spot types and their properties, see [Hot Spot Types](#) (page 252).

Using the D2HML Attribute in HTML Source Documents

The *d2hml* attribute is a special attribute that can be used in your HTML source documents to keep the secondary part of a hot spot, which contains all of the D2HML properties for the hot spot, from appearing as text in the design view of your HTML editor. It can also be used to specify a hot spot on HTML elements that are not allowed to have text, such as the <AREA> tag.

For example, here is how a topic link with the D2HML property *Tag* set to a specific topic appears in the design view of your HTML editor:

Here is a **Topic Link|tag=My Topic**.

In HTML code, this sentence looks like this:

```
Here is a <span class=C1HJump>Topic Link<span style="display:none">|tag=My Topic</span>
```

You can keep the secondary part from appearing in the design view by using the *d2hml* attribute. In HTML code, the same sentence looks like this:

```
Here is a <span class="C1HJump" d2hml="tag=My Topic">Topic Link.</span>
```

In the design view of your HTML editor, it appears without the secondary part:

Here is a **Topic Link**.

The *d2hml* attribute can also be used to specify the D2HML hotspot style without changing the *class* attribute of an HTML element. The value of the *d2hml* attribute must start with the style name followed by the secondary part, separated by a separator bar (|), as in the following example:

```
Here is a <span class="myclass" d2hml="C1HJump|tag=My Topic">Topic Link</span>.
```

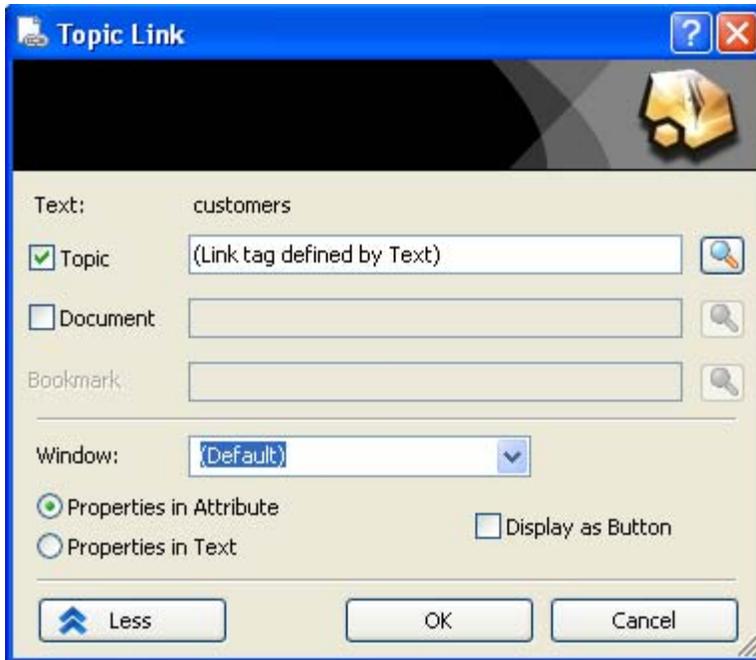
This allows you to keep the class, *myclass*, which may be used for appearance purposes, while converting the HTML element into a D2HML hot spot. The secondary part does not appear in the design view of your HTML editor.

When using the *d2hml* attribute, the style name, **C1HJump** in the previous examples, must exactly match the value of the [HTMLName](#) property of a character style in the Doc-To-Help project. In Doc-To-Help, the value of the [HTMLName](#) property for **C1H Jump** is **.C1HJump**. Note that you do not need to include the leading period in the *d2hml* attribute. If the value of the [HTMLName](#) property includes a tag name, such as **SPAN.MyStyle**, the tag name must be used in the *d2hml* attribute as in the following example:

```
Here is a <span class="myclass" d2hml="SPAN.MyStyle|tag=My Topic">Topic Link.</span>
```

Hot spot properties are stored in the *d2hml* attribute by default, unless you are creating one of the invisible hot spot types. In this case, properties are stored as text in your HTML document.

Most D2HML dialog boxes contain a **More** button that, when clicked, displays additional properties for the hot spot. **Properties in Attribute** is checked by default. If you do not want the hot spot properties to be stored in the *d2hml* attribute, select **Properties in Text**.



Note: If using Dreamweaver, you can choose to show or hide the hot spot properties in your document. To hide them, select **Hide Properties in Document**. Hot spot properties are always hidden in FrontPage, so this checkbox is not available in the D2HML dialog boxes for FrontPage.

Using D2HML Comments in Word Source Documents

A D2HML comment is simply a Word comment containing D2HML. You may want to use a comment rather than a D2HML style if you have a large range of text containing style formatting, because Word does not support nested styles. Therefore, if you apply a D2HML style to formatted text, some style formatting will not appear correctly in your Help target.

Conditional, expanding and drop-down text is applied as a D2HML comment in your Word documents by default; this default setting can be changed using the [D2HML Options](#) (page 413) dialog box. All other types of hot spots can be created manually using the following steps.

To create a D2HML comment:

1. In your Word source document, select the text/image range to which you want to have a comment applied. Select the range as if you were going to copy it to the Clipboard.
2. Select **Insert | Comment**.
3. Add the following text, which will become your comment text:

```
D2HML | <style name> | <secondary part>
```

- The **<style name>** is the name of a D2HML style such as C1H Conditional, C1H Jump or C1H Dropdown Text.
- The **<secondary part>** is a [D2HML hot spot](#) (page 233) secondary part.

For example, if you wanted to create a link to a specific topic, select your text/image range and insert the following comment text:

D2HML|C1H Jump|tag=My Topic

details that you
advertising and
with you, the Fr
easy for you to t

Revision 1, 2/21/2006 2:05:00 PM
commented:
D2HML|C1H Jump|tag=My Topic

This is a link to [My Topic](#) [Rev1]].

To remove a D2HML comment, right-click the commented text and select **Delete Comment**.

Note: For more information on using D2HML comments for conditional, drop-down and expanding text, see [Expanding, Dropdown and Popup Text](#) (page 313) and [Working with Conditional Text in Word Documents](#) (page 403).

D2HML Quick Reference

The following table is a quick reference guide to be used when formatting your HTML and Word source documents using the **D2HML Styles** toolbar. The table provides a description and the corresponding HTML code and Word style for each button, along with the properties it supports.

Icon	Definition	Properties	HTML Code	Word Style	Word Shortcut Keys	Syntax Notes
	Topic Link: Creates links to a specified topic.	<i>bookmark</i> <i>document</i> <i>tag</i> <i>windowDisplayAsButton</i>	<code> </code>	C1H Jump	Alt+Ctrl+J	N/A
	Popup Link: Opens a topic in a pop-up window.	<i>bookmark</i> <i>document</i> <i>tag</i> <i>window</i> <i>DisplayAsButton</i>	<code> ></code>	C1H Popup	Alt+Ctrl+P	N/A
	Keyword Link: Links hot spot to a topic with the specified keyword.	<i>keyword</i> <i>window</i> <i>DisplayAsButton</i>	<code></code>	C1H Keyword Link	Alt+Ctrl+K	Can link to multiple keywords: Hotspot keyword=Keyword 1 + Keyword 2
	Group Link: Links hot spot to a group of topics.	<i>group</i> <i>window</i> <i>DisplayAsButton</i>	<code></code>	C1H Group Link	Alt+Ctrl+G	Can link to multiple groups: Hotspot group=Group 1 + Group 2
	Glossary Link: Links hot spot to a glossary term.	<i>tag</i> <i>DisplayAsButton</i>	<code> ></code>	C1H Popup	Alt+Ctrl+A	N/A
	Inline Expand: When clicked, the expanding text appears.	N/A	<code> </code>	C1H Inline Expand	Alt+Ctrl+E	N/A
	Inline DropDown: When clicked, the drop-down text appears.	N/A	<code></code>	C1H Inline Dropdown	Alt+Ctrl+D	N/A

Icon	Definition	Properties	HTML Code	Word Style	Word Shortcut Keys	Syntax Notes
	Inline Popup: When clicked, the pop-up text appears.	N/A	<code></code>	C1H Inline Popup	Alt+Ctrl+U	N/A
	Keyword: Designates hot spot as an index keyword.	<i>keyword</i>	<code></code>	C1H Index	Alt+Ctrl+W	Can create multi-level keywords: Hotspot keyword=Parent Keyword, Child Keyword.
	Invisible Keyword: Designates hot spot as an index keyword only the hot spot does not appear in the output.	<i>keyword</i>	<code></code>	C1H Index Invisible	Alt+Ctrl+Shift+W	Can create multiple keywords: Hotspot keyword=Keyword 1 + Keyword 2
	Group: Designates hot spot as a member of a group.	<i>group</i>	<code></code>	C1H Group	Alt+Ctrl+R	Can create multiple groups: Hotspot group=Group 1 + Group 2
	Invisible Group: Similar to Group except the text does not appear in the output.	<i>group</i>	<code></code>	C1H Group Invisible	Alt+Ctrl+Shift+R	N/A
	Link Tag: Designates hot spot as a link target.	<i>linktag</i>	<code></code>	C1H Link Tag	Alt+Ctrl+L	Can create multiple Link Tags: Hotspot linktag=Link Tag 1 + Link Tag 2
	Invisible Link Tag: Similar to Link Tag except the text does not appear in the output.	<i>linktag</i>	<code></code>	C1H Link Tag Invisible	Alt+Ctrl+Shift+L	N/A
	Context ID: Assigns a context ID to the topic.	<i>contextid</i>	<code></code>	C1H Context ID	Alt+Ctrl+I	Can create multiple Context IDs: Hotspot contextid= 1 + 2
	Contents Title: Assigns a title to the topic for the TOC. Text is hidden in the output.	<i>contentstitle</i>	<code></code>	C1H Contents Title	Alt+Ctrl+N	N/A

Icon	Definition	Properties	HTML Code	Word Style	Word Shortcut Keys	Syntax Notes
	Topic Properties: Sets various topic properties. Text is hidden in the output.	<i>asciiname</i> <i>buttonlabel</i> <i>comments</i> <i>contentstitle</i> <i>displaytitle</i> <i>searchenabled</i> <i>url</i> <i>default</i> <i>attribute</i> <i>platform</i> <i>target</i>	<code></code>	C1H Topic Properties	Alt+Ctrl+T	Platform can be: Manual Online HTMLAny NetHelp HTMLHelp MSHelp JavaHelp WinHelp
	Conditional Text – Online: Text only appears in the Online Help output.	N/A	<code></code>	C1H Online	Alt+Ctrl+O	N/A
	Conditional Text – Manual: Text only appears in a printed manual.	N/A	<code></code>	C1H Manual	Alt+Ctrl+M	N/A
	Conditional Text: Allows you to specify one or more platform, target, attribute, or any combination of these in which hot spot text appears.	<i>attribute</i> <i>platform</i> <i>target</i> <i>passthrough</i>	<code></code>	C1H Conditional	Alt+Ctrl+C	Platform or target can be: Manual Online HTMLAny NetHelp HTMLHelp MSHelp JavaHelp WinHelp Passthrough applies only to Word source documents.
	Clear Condition: Clears conditional text styles while leaving other formatting intact. Located on the FrontPage and Dreamweaver D2HML Styles toolbar only.	N/A	N/A	N/A	N/A	N/A
	Clear Formatting: Erases any formatting you apply to text.	N/A	N/A	N/A	Alt+Ctrl+F	N/A
	Show Hidden Hot Spots: Shows any invisible hot spots. Located on the Word and Dreamweaver D2HML Styles toolbar only.	N/A	N/A	N/A	Alt+Ctrl+H	N/A
	Apply Style: Formats any selection with any of the D2HML styles.	N/A	N/A	N/A	Alt+Ctrl+S	N/A
	D2HML Options: Sets the default type of conditional text hot spots in Word, either As Comments or As Styles .	N/A	N/A	N/A	N/A	N/A

Creating a Hot Spot

There are several ways to create a hot spot, including using the **D2HML Styles** toolbar in Microsoft Word, FrontPage and Dreamweaver, using the style drop-down box or menu provided with most WYSIWYG HTML editors and Word, or using plain HTML code in an HTML editor. There are no major differences between the methods and how they work with Doc-To-Help; each applies the D2HML style. There are some differences, though, when it comes to convenience.

It is recommended that you use the **D2HML Styles** toolbar, because it offers the D2HML dialog boxes and creates the D2HML for you. What could be easier!

In Word and most WYSIWYG HTML editors, you can also apply a D2HML style by selecting it from the list of available styles in the style drop-down box or menu. This is not as convenient as using the **D2HML Styles** toolbar and dialog boxes, but it is an option.

Formatting with plain HTML code is usually done by advanced HTML users, when not using FrontPage, Dreamweaver or another WYSIWYG HTML editor.

Using the D2HML Styles Toolbar in Word:

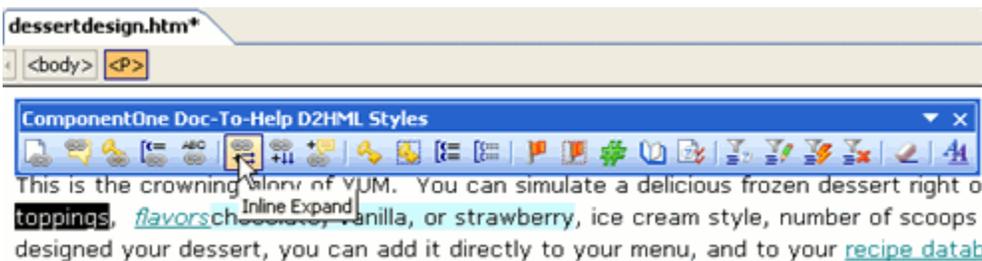
1. Select the topic or text you want to be part of the hot spot.
2. Click a button from the **D2HML Styles** toolbar to apply the desired D2HML style.



3. Edit the information in the dialog box, if necessary. In this example, **Key Property** is assigned to the **Tag** property of the hot spot. The hot spot text is **additional properties | tag=Key Property**; it has been formatted with the C1H Jump style. When the hot spot is clicked, it jumps to the **Key Property** topic. The secondary part, **|tag=Key Property**, does not appear in the help target when it is built.

Using the D2HML Styles Toolbar in FrontPage and Dreamweaver:

1. Select the topic or text you want to be part of the hot spot.
2. Click a button from the **D2HML Styles** toolbar to apply the desired D2HML style.



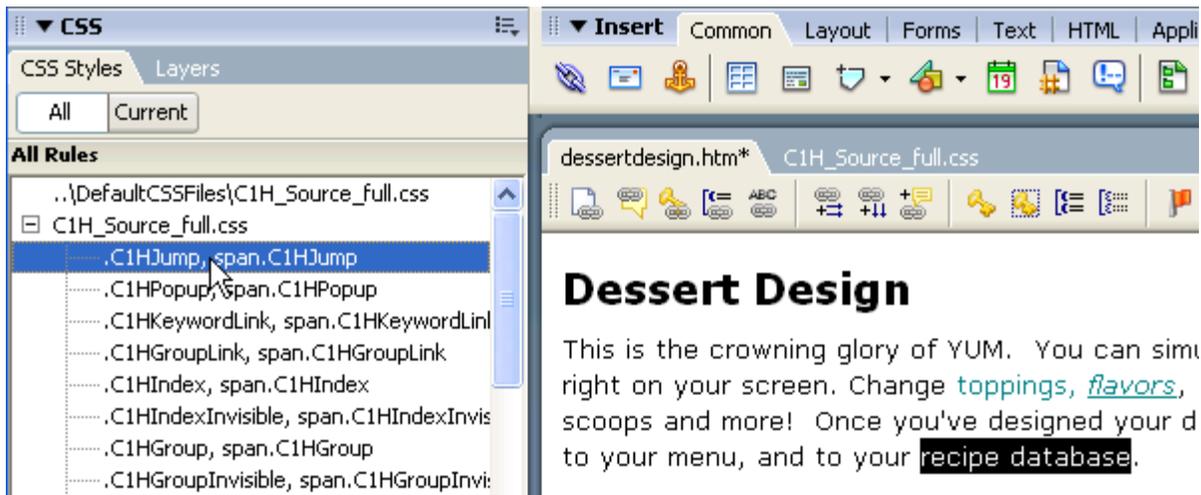
3. Edit the information in the dialog box, if necessary.

In this example, **Working with the Recipe Database** is assigned to the **Tag** property. The hot spot text is **Recipe Database | Tag=Working with the Recipe Database**; it has been formatted with the C1H Jump style. When the hot spot is clicked, it jumps to the **Working with the Recipe Database** topic. The secondary part, **|Tag=Working with the Recipe Database**, does not appear in the help target when it is built.

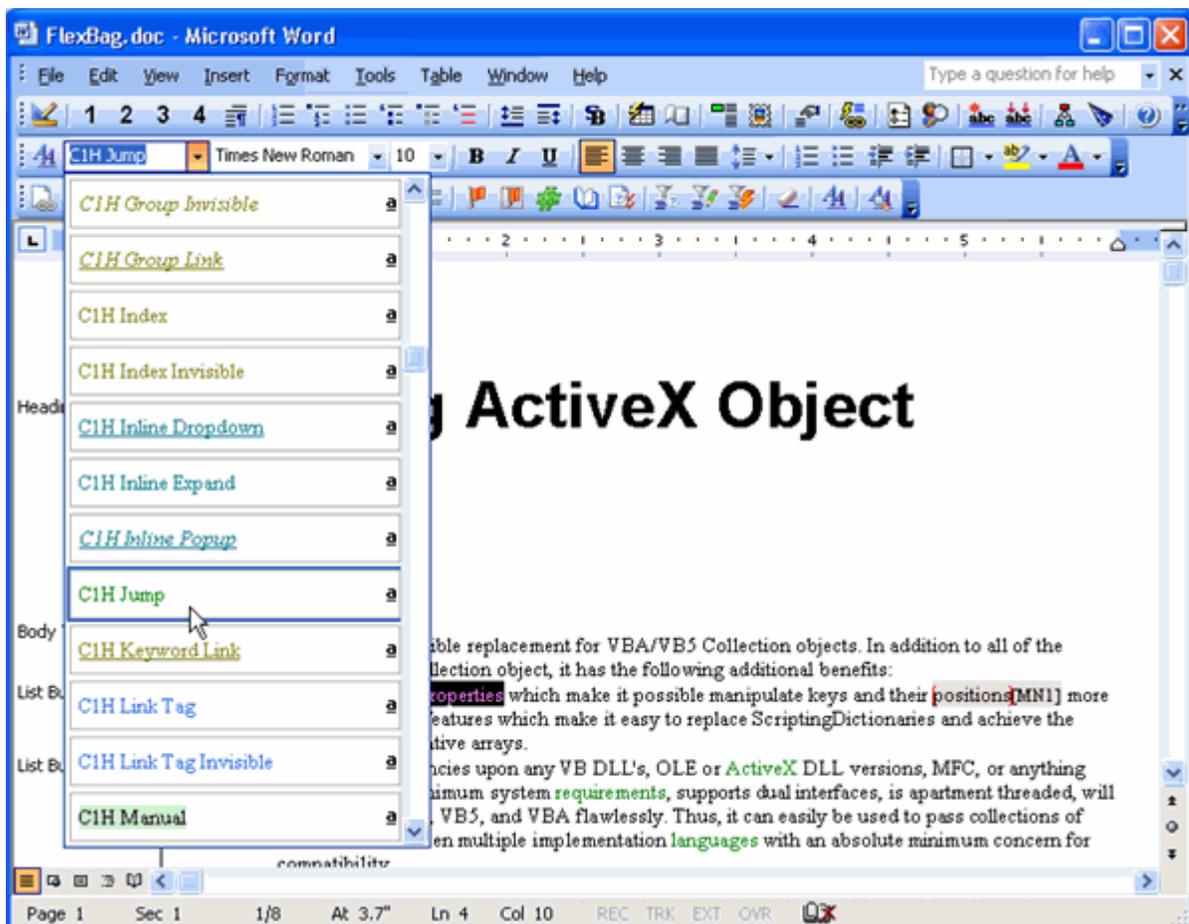
Using the Style Drop-down Box in Word and WYSIWYG HTML Editors

1. Set the properties for your hot spot type, if necessary.

2. Select the topic or text you want to be part of the hot spot, including the secondary part if it exists.
3. Select a D2HML style from the styles drop-down box or relevant menu.



D2HML Styles in Dreamweaver, a WYSIWYG HTML Editor



D2HML Styles in Word

In this example, **Working with the Recipe Database** is assigned to the **Tag** property. The hot spot text is **Recipe Database | Tag=Working with the Recipe Database**; it has been formatted with the C1H Jump style.

When the hot spot is clicked, it jumps to the **Working with the Recipe Database** topic. The secondary part, |Tag=Working with the Recipe Database, does not appear in the help target when it is built.

Using HTML Code in Any HTML Editor:

1. Determine the topic or text that you want to be part of the hot spot, and, if necessary, add a secondary part.
2. Add an opening HTML `` tag with the appropriate class in front of the hot spot text.

```

8 <BODY>
9 <H4>To Start the Ice Cream Creation Simulator</H4>
10 <OL class=List>
11 <LI>From the <B>File</B> Menu, choose <B>New</B>. The File New dialog box will
12 appear with several choices.
13 <LI>Choose <B>Ice Cream Creation</B>. The Ice Cream Creation Simulator dialog
14 box will appear.
15 <LI>From this dialog box you can choose ice cream style, flavor, toppings and
16 much more.
17 <LI>Click the different choices in the dialog .box until you are satisfied
18 with your selection.
19 <LI>Click <B>OK</B>. A dialog box will appear, prompting you to give your
20 creation a name.
21 <LI dir=ltr>
22 <P dir=ltr>Enter the name of your creation. By checking the corresponding
23 boxes, you can add the recipe to the <span class="CIHJump">Recipe
24 Database|tag=Working with the Recipe Database</span> and to your
25 Menu. </P></LI></OL>
26 <P>
27 <P align=center>
28 <HR>
29
30 <P></P></BODY></HTML>

```

3. Add a closing `` tag after the hot spot text. Make sure to include the secondary part, if it exists, in the tag.

```

8 <BODY>
9 <H4>To Start the Ice Cream Creation Simulator</H4>
10 <OL class=List>
11 <LI>From the <B>File</B> Menu, choose <B>New</B>. The File New dialog box will
12 appear with several choices.
13 <LI>Choose <B>Ice Cream Creation</B>. The Ice Cream Creation Simulator dialog
14 box will appear.
15 <LI>From this dialog box you can choose ice cream style, flavor, toppings and
16 much more.
17 <LI>Click the different choices in the dialog .box until you are satisfied
18 with your selection.
19 <LI>Click <B>OK</B>. A dialog box will appear, prompting you to give your
20 creation a name.
21 <LI dir=ltr>
22 <P dir=ltr>Enter the name of your creation. By checking the corresponding
23 boxes, you can add the recipe to the <span class="CIHJump">Recipe
24 Database|tag=Working with the Recipe Database</span> and to your
25 Menu. </P></LI></OL>
26 <P>
27 <P align=center>
28 <HR>
29
30 <P></P></BODY></HTML>

```

In this example, the hot spot text is **Recipe Database | Tag=Working with the Recipe Database**. It has been formatted with the CIH Jump style. When the hot spot is clicked, it jumps to the **Working with the Recipe Database** topic. The secondary part, |Tag=Working with the Recipe Database, does not appear in the help target when it is built.

Hot Spot Types

In order for a hot spot to work correctly, it must have a corresponding character style in the Doc-to-Help project. The Behavior property of the style is used to determine what the hot spot does. For example, C1H Jump is a predefined character style in Doc-to-Help. The Behavior property of C1H Jump is **Topic Link**, so Doc-To-Help knows to treat the hot spot formatted with the C1H Jump style as a link to the topic.

In some cases, more information is necessary to get a desired behavior. In these instances, the properties of the hot spot type can be used in the secondary part of the hot spot. For example, suppose you want to create a link from the text **MyTopic** to the topic **My First Topic**. Because the link text is not an exact textual match of the topic, you can use the *Tag* property of the Topic Link hot spot type to specify the topic to which you want to link. The link text would look like this:

[MyTopic|tag=My First Topic](#)

You can also add your own hot spot types simply by adding a character style to the Doc-To-Help project and specifying its Behavior property. For more information on adding character styles, see [Creating a D2HML Style](#) (page 235).

Note that a character style included in a Doc-To-Help project is used in building a help target if its CharacterStyle.Behavior property is set to a value other than **None**.

The following topics provide more information on each hot spot type, including the predefined D2HML styles, behaviors, properties and corresponding HTML tags and Word styles. The dialog boxes shown are from Microsoft Word. They may look slightly different in FrontPage and Dreamweaver.

Note: It is recommended that you apply D2HML styles to text using the **D2HML Styles** toolbars in Word, FrontPage and Dreamweaver. Formatting an HTML source document with plain HTML code is usually done by advanced HTML users, when not using FrontPage, Dreamweaver or another WYSIWYG HTML editor.

Topic Link Hot Spot

The Topic Link hot spot creates a jump or pop-up link in the topic containing this hot spot.

The Behavior property of this hot spot type is set to **Topic Link**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Jump (Hidden=False)		C1H Jump
C1H Popup (Hidden=False)		C1H Popup

The following properties are supported by this hot spot type:

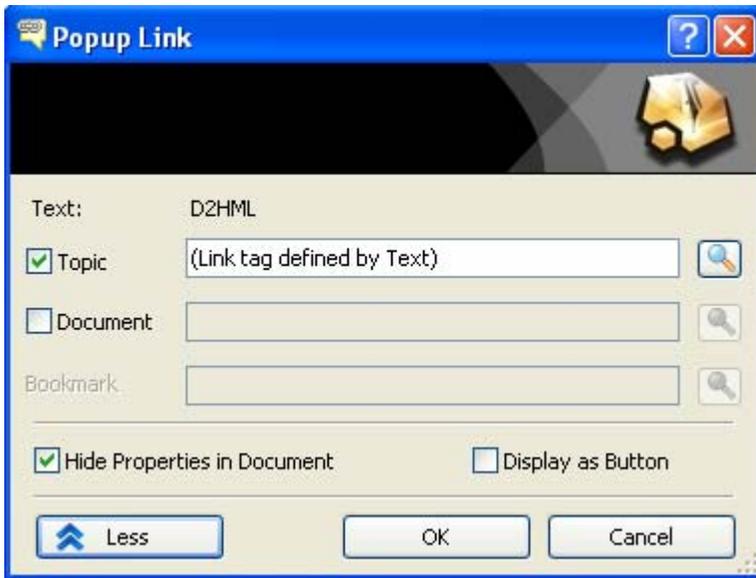
Supported Properties:	Description
Tag	<p>This is the link tag of a topic, or the target of the link. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Default • Simple • Case-insensitive
Document	<p>This is the document where the target of the link is located. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Secondary • Case-insensitive <p>There is a special value reserved for the this property: Document=d2hmlDefault. This value refers to the current document, or the document where the hot spot is located.</p>

Supported Properties:	Description
Bookmark	<p>This is the bookmark where the target of the link is located within the document specified by the Document property. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Secondary • Case-insensitive <p>In Word, this is a bookmark name. In HTML source documents, this is the name of an <A> anchor tag.</p>
Window	<p>This is the name of the window where the target topic appears. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Secondary • Case-insensitive <p>Note: This property does not apply to the C1H Popup style.</p>
DisplayAsButton	<p>This property specifies whether the hot spot is represented by a button in help targets. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Secondary • Boolean value True or False • Case-insensitive • Default value: False

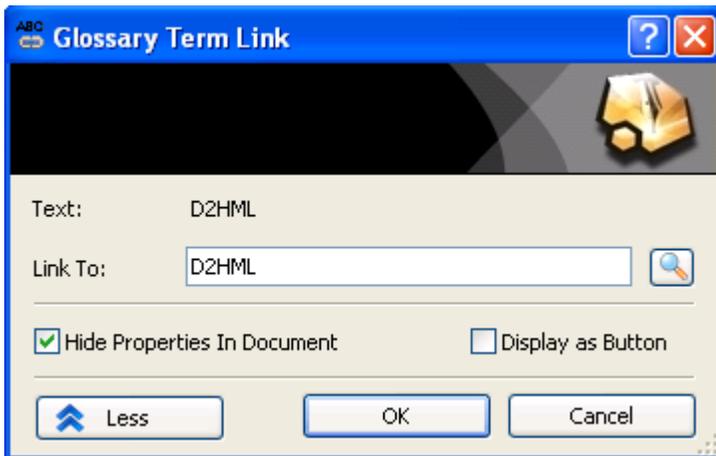
To create a Topic Link hot spot, select the link text and press **Jump Topic Link** , **Popup Topic Link**  or **Glossary Link**  on the **D2HML Styles** toolbar. Use the **Topic Link**, **Popup Link** and **Glossary Term Link** dialog boxes to specify the properties of the link.



Topic Link Dialog Box



Popup Link Dialog Box



Glossary Term Link Dialog Box

There are several options for creating topic and pop-up links. You can create a link to a topic using the topic's link tag or topic title and the **Tag** property. You can also create a link to a bookmark inside a topic using the **Document** and **Bookmark** properties. If you have multiple, identical topics, you can specify in which document the topic you want to link to is located using the **Tag** and **Document** properties.

To link to a topic using the topic's link tag and the **Tag** property:

- If the topic you are linking to is identical to the link text, you can leave the **Topic** field set to the default (*Link tag defined by text*). The link tag is specified in the **Tag** property of the hot spot:

Topic Link

- If the topic you are linking to does not match the link text, enter a link tag in the **Topic** field. If the link tag is unknown, click the **Browse** button next to the **Topic** field, select **Link Tags** in the **Select Topic** dialog box, and choose one of the link tags for the topic. The link tag is specified in the **Tag** property of the hot spot:

Topic Link|tag=My_Topic

To link to a topic using the topic's title and the **Tag** property:

Enter the topic's name, if known, in the **Topic** field or click the **Browse** button next to the **Topic** field and select the topic.

The topic title is specified in the **Tag** property of the hot spot:

`Topic Link|tag=My Topic`

To link to a bookmark inside a document using the **Document** and **Bookmark** properties:

Note: You may specify a topic in the **Topic** field to ensure the bookmark belongs to the correct topic, but this is not necessary once a document and bookmark is selected. The **Topic** checkbox should be unchecked if you are specifying a document and bookmark but not a topic.

1. Check the **Document** checkbox and select the document where the bookmark is located. If the bookmark is in the current document, leave the (*Current document*) setting, which sets the **Document** property to **d2htmlDefault**.
2. Click the **Browse** button next to the **Bookmark** field and select a bookmark from the specified document. The document is specified in the **Bookmark** property of the hot spot:

`Topic Link|document=d2htmlDefault;bookmark=Bookmark1`

If your project contains multiple, identical topics, it is recommended that you create a unique link tag for each topic and then link to the topic using the topic's link tag. For more information on link tags, see [Link Tag and Invisible Link Tag Hot Spots](#) (page 266) and [Adding a Link Tag to a Topic](#) (page 289). Alternatively, if you want to link to a topic using the topic title, you can use the **Document** property to specify which topic you want to link to.

To link to a topic using the **Tag** and **Document** properties:

1. Specify a topic. If you click **Browse** and use the **Select Topic** dialog box, you can look for topics in different documents by clicking the **Look in** drop-down arrow and selecting a document.
2. Check the **Document** checkbox and select the document where the topic you want to link to is located. If it is located in the current document, leave the default (*Current document*) setting, which sets the **Document** property to **d2htmlDefault**.

`Topic Link|tag=My Topic; document=WordDocuments\Doc1.doc`

Note: If the default option for the **Topic** field (*Link tag defined by Text*) is used when the **Document** checkbox is checked, the **Tag** property is automatically set to **d2htmlDefault**. In this case, the link tag used to create the link matches the hot spot text. If the default option for the **Document** field (*Current document*) is used, the document property is automatically set to **d2htmlDefault**. Doc-To-Help then knows to link to the current document.

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML `` tag to format it with the D2HML style.

The following table gives you some examples of using the Topic Link hot spot. Each of the following hot spots appears as Topic 12 in the help target. The secondary part is not visible.

Hot Spot in Word and HTML Editor	HTML Code	Description
Topic 12	<code>Topic 12</code>	A pop-up link to a topic having Topic_12 as one of its link tags is created.
Topic 12 window=proc	<code>Topic 12 window=proc</code>	A jump to a topic having Topic_12 as one of its link tags is created. That topic will be shown in the 'proc' window.
Topic 12 tag=Topic 12;window=proc	<code>Topic 12 tag=Topic 12;window=proc</code>	A jump to a topic having Topic_12 as one of its link tags is created. That topic will be shown in the window with the name proc .
Topic 12 =op	<code>Topic 12 =op</code>	A pop-up link to a topic having op as one of its link tags is created.

Hot Spot in Word and HTML Editor	HTML Code	Description
Topic 12 tag=Topic 12;bookmark=Bookmark1;document=WordDocuments\Doc1.doc	 d2hml="tag=Topic 12;bookmark=Bookmark1;document=HTMLDocuments\Doc1.htm	A jump to Bookmark1 in a topic having Topic_12 as one of its link tags in the Doc1.doc is created.
Glossary Term	Glossary Term	A pop-up link to a glossary heading having gloss_Glossary Term as one of its link tags is created.

Hot Spot in Word	Description
Topic 12 tag=Topic 12;bookmark=Bookmark1;document=WordDocuments\Doc1.doc	A jump to Bookmark1 in a topic having Topic_12 as one of its link tags in Doc1.doc is created.
Topic 12 bookmark=Bookmark1;document=WordDocuments\Doc1.doc	A jump to Bookmark1 in Doc1.doc is created. It is not necessary for the tag property to be specified since bookmark names are unique.

Hot Spot in HTML Editor	HTML Code	Description
Topic 12 tag=Topic 12;bookmark=Bookmark1;document=HTMLDocuments\Doc1.htm	 Topic12 tag=Topic 12;bookmark=Bookmark1;document=HTMLDocuments\Doc1.htm	A jump to Bookmark1 in a topic having Topic_12 as one of its link tags in Doc1..htm is created.
Topic 12 bookmark=Bookmark1;document=HTMLDocuments\Doc1.htm	 Topic12 bookmark=Bookmark1;document=HTMLDocuments\Doc1.htm	A jump to Bookmark1 in Doc1..htm is created. It is not necessary for the tag property to be specified since bookmark names are unique.

Keyword Link Hot Spot

The Keyword Link hot spot creates a link to the topics associated with the keyword contained in the hot spot or the specified keyword. A topic must be associated with the keyword before this hot spot can link to it. For example, you can create a link to the topics associated with the keyword *MyKeyword*. To associate a topic with a keyword, see [Keyword and Invisible Keyword Hot Spots](#) (page 263).

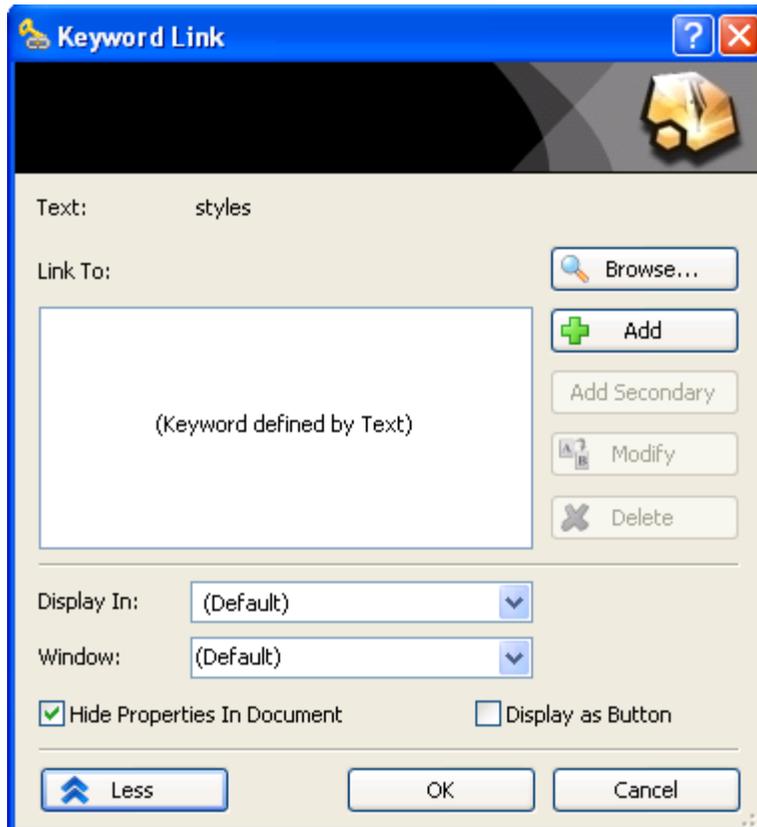
The Behavior property of this hot spot type is set to **Keyword Link**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Keyword Link (Hidden=False)		C1H Keyword Link

The following properties are supported by this hot spot type:

Supported Properties:	Description
Keyword	This is one or more keywords defining the list of topics shown to the user as link targets. The characteristics of this property are: <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive
Window	This is the name of the window displaying the topic after the user selected the target from the list . The characteristics of this property are: <ul style="list-style-type: none"> • Secondary • Case-insensitive
DisplayAsButton	This property specifies whether the hot spot is represented by a button in help targets. The characteristics of this property are: <ul style="list-style-type: none"> • Secondary • Boolean value: True or False • Case-insensitive • Default value: False

To create a Keyword Link hot spot, select the link text and press **Keyword Link**  on the **D2HML Styles** toolbar. Use the **Keyword Link** dialog box to specify the properties of the link.



Keyword Link Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

Note: When the **More** button is clicked while using this dialog box in FrontPage and Dreamweaver, there is a **Display as a button** checkbox, not a **Hide Properties in Document** checkbox. When this checkbox is selected and the help target is built, the link appears as a button rather than a link.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML `` tag to format it with the D2HML style.

The following table gives you some examples of using the Keyword Link hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
MyTopic	<code>MyTopic</code>	A link to the topics associated with the keyword MyTopic is created.
MyTopic window=proc	<code>MyTopic window=proc</code>	A link to the topics associated with the keyword MyTopic is created. The topic selected by the user is shown in the window with the name proc .
Text keyword=Keyword;window=proc	<code>Text keyword=Keyword;window=proc</code>	A link to the topics associated with the keyword Keyword is created. The topic selected by the user is shown in the window with the name proc .

Hot Spot in Word and HTML Editor	HTML Code	Description
Text keyword=Keyword1+Keyword2	Text keyword=Keyword1+Keyword2	A link to the topics associated with keyword Keyword1 or Keyword2 is created.
Text =Keyword1+Keyword2	Text =Keyword1+Keyword2	A link to the topics associated with keyword Keyword1 or Keyword2 is created.
Text keyword=Keyword1+Keyword2,Secondary keyword	Text keyword=Keyword1+Keyword2,Secondary keyword	A link to the topics associated with keyword Keyword1 or with the second-level keyword Secondary keyword nested in the keyword Keyword1 is created.

Group Link Hot Spot

The Group Link hot spot creates a link to the topics belonging to the group contained in the hot spot or the specified group. A topic must be a group member before this hot spot can link to it. For example, you can create a link to the topics belonging to the group *MyGroup* if the topics are members of *MyGroup*. To assign a topic to a group, see [Group and Invisible Group Hot Spots](#) (page 265).

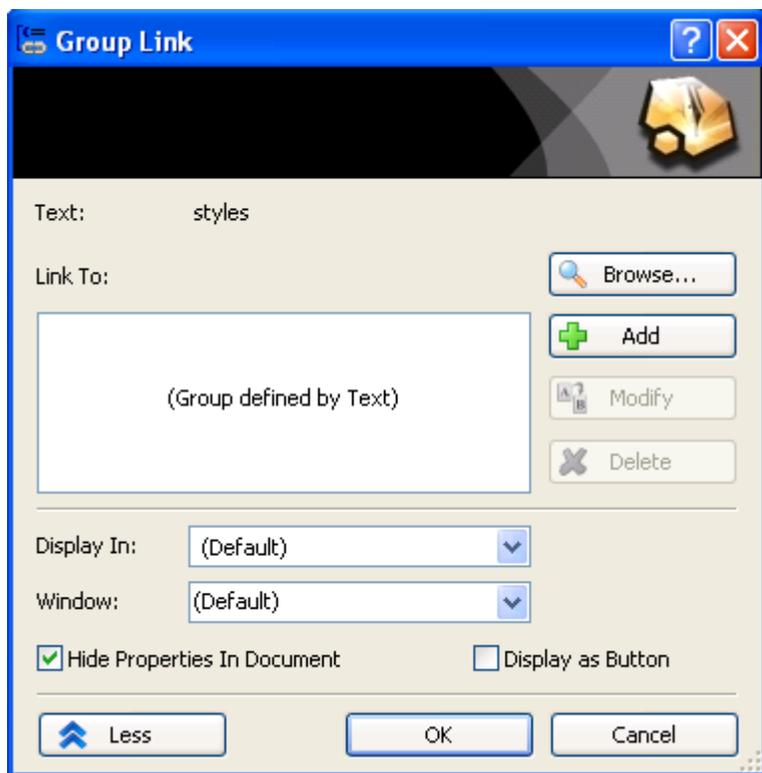
The Behavior property of this hot spot type is set to **Group Link**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Group Link (Hidden=False)		C1H Group Link

The following properties are supported by this hot spot type:

Supported Properties:	Description
Group	This is one or more groups (usually one) defining the list of topics shown to the user as link targets. The characteristics of this property are: <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive
Window	This is the name of the window displaying the topic after the user selects the target from the list. The characteristics of this property are: <ul style="list-style-type: none"> • Secondary • Case-insensitive
DisplayAsButton	This property specifies whether the hot spot is represented by a button in help targets. The characteristics of this property are: <ul style="list-style-type: none"> • Secondary • Boolean value True or False • Case-insensitive • Default value: False

To create a Group Link hot spot, select the link text and press **Group Link**  on the **D2HML Styles** toolbar. Use the **Group Link** dialog box to specify the properties of the link.



Group Link Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the View tab, check the **Hidden text** checkbox.

Note: When the **More** button is clicked while using this dialog box in FrontPage and Dreamweaver, there is a **Display as a button** checkbox, not a **Hide Properties in Document** checkbox. When this checkbox is selected and the help target is built, the link appears as a button rather than a link.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following table gives you some examples of using the Group Link hot spot. Each of the following hot spots appears as MyTopic in the help target. The secondary part is not visible.

Hot Spot in Word and HTML Editor	HTML Code	Description
<i>MyTopic</i>	<code>MyTopic</code>	A link to the topics belonging to the group MyTopic is created.
<i>MyTopic window=proc</i>	<code>MyTopic window=proc</code>	A link to the topics belonging to the group MyTopic is created. The topic selected by the user is shown in the window with the name proc .
<i>MyTopic group=Group;window=proc</i>	<code>MyTopic group=Group;window=proc</code>	A link to the topics belonging to the group Group is created. The topic selected by the user is shown in the window with the name proc .
<i>MyTopic group=Group1+Group2</i>	<code>MyTopic group=Group1+Group2</code>	A link to the topic list containing topics in groups Group1 and Group2 is created.
<i>MyTopic =Group1+Group2</i>	<code>MyTopic =Group1+Group2</code>	A link to the topic list containing topics in groups Group1 and Group2 is created.

Inline Expand and Expand Text Hot Spots

The Inline Expand hot spot appears as a link in the help target, and, when clicked, shows the expanding text specified in the Expand Text hot spot. These hot spots work together and cannot be used separately from one another.

Note: Inline Expand and Expand Text hot spot types are not supported in Manual, WinHelp and JavaHelp help targets.

The Behavior property of the Inline Expand hot spot type is set to **Inline Expand**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

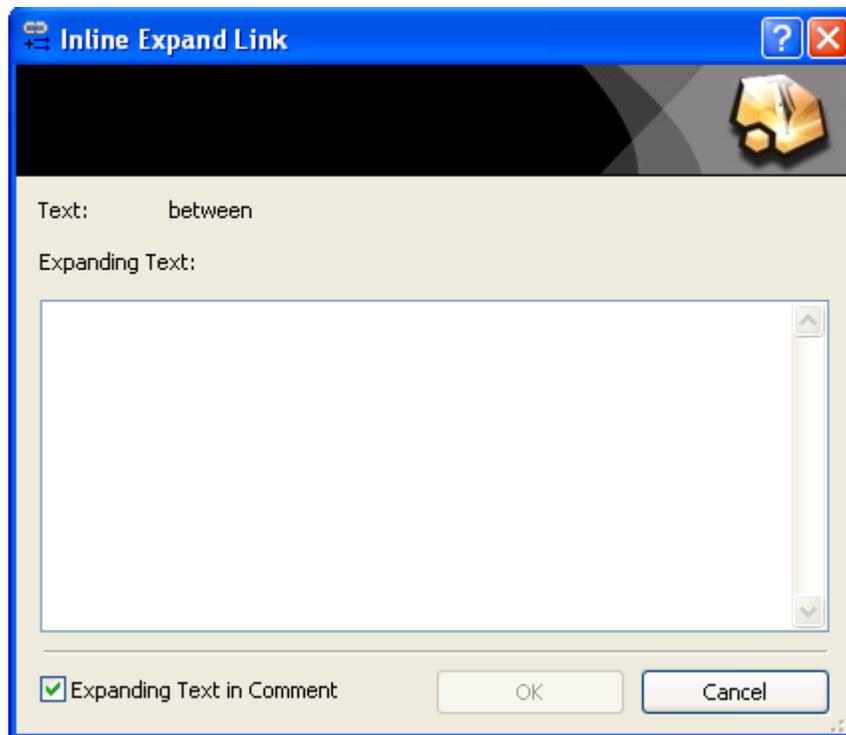
D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Inline Expand (Hidden=False)	 	C1H Inline Expand

The Behavior property of the Expand Text hot spot type is set to **Inline Text**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Expand Text (Hidden=False)	 	C1H Expand Text

No properties are necessary for these hot spot types.

To create an Inline Expand hot spot, select the link text and press **Inline Expand**  on the **D2HML Styles** toolbar. Enter the expanding text within the **Inline Expand Link** dialog box.



If you are not using the toolbar, place the link text within the corresponding HTML tag, and place the expanding text within the corresponding HTML tag to format them with these D2HML styles.

The following table gives you an example of using the Inline Expand and Expand Text hot spots:

Hot Spot in Word and HTML Editor	HTML Code	Description
My Link Text My Expanding Text	<code>My TextExpanding Text</code>	When My Link Text is clicked, My Expanding Text appears next to it.

Inline DropDown and Dropdown Text Hot Spots

The Inline DropDown hot spot appears as a link in the help target, and, when clicked, shows the drop-down text specified in the Dropdown Text hot spot below the Inline DropDown hot spot. These hot spots work together and cannot be used separately from one another.

Note: Inline DropDown and Dropdown Text hot spot types are not supported in Manual, WinHelp and JavaHelp help targets.

The Behavior property of the Inline DropDown hot spot type is set to **Inline DropDown**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

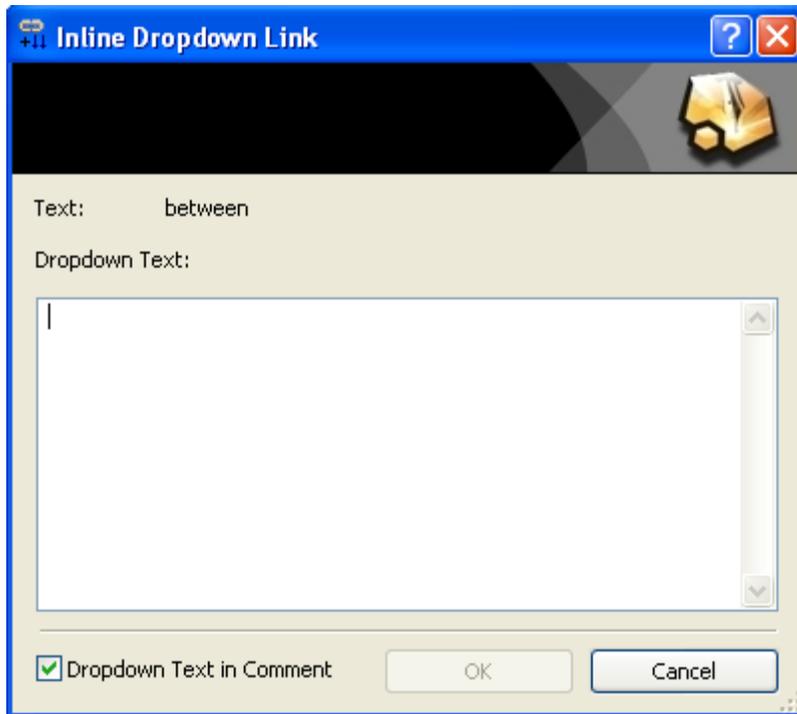
D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Inline Dropdown (Hidden=False)	<code> </code>	C1H Inline Dropdown

The Behavior property of the Dropdown Text hot spot type is set to **Inline Text**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Dropdown Text (Hidden=False)	<code><div class="C1HDropdownText"> </div></code>	C1H Dropdown Text

No properties are necessary for these hot spot types.

To create an Inline DropDown hot spot, select the link text and press **Inline DropDown**  on the **D2HML Styles** toolbar. Enter the drop-down text within the **Inline Dropdown Link** dialog box.



Drop-down text may consist of more than one paragraph with style formatting. Microsoft Word does not allow you to apply style formatting within drop-down text because it is already formatted with the **C1H Dropdown Text** style. By default, Doc-To-Help applies drop-down text as a comment rather than a D2HML style in your Word documents so that style formatting may be used within the comment. For more information, see [Creating Dropdown Text](#) (page 315).

If you are not using the toolbar, place the link text within the corresponding HTML `` tag, and place the drop-down text within the corresponding HTML `<div class="C1HDropdownText">` tag to format them with these D2HML styles.

The following table gives you some examples of using the Inline Text hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
My Link Text My Dropdown Text	<code><p>...My Text...</p><div class="C1HDropdownText">Dropdown Text</div></code>	When My Link Text is clicked, My Dropdown Text appears below it.

Inline Popup and Popup Text Hot Spots

The Inline Popup hot spot appears as a link in the help target, and, when clicked, opens a pop-up box containing the pop-up text specified in the Popup Text hot spot. These hot spots work together and cannot be used separately from one another.

Note: Inline Popup and Popup Text hot spot types are not supported in Manual, WinHelp and JavaHelp help targets.

The Behavior property of the Inline Popup hot spot type is set to **Inline Popup**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

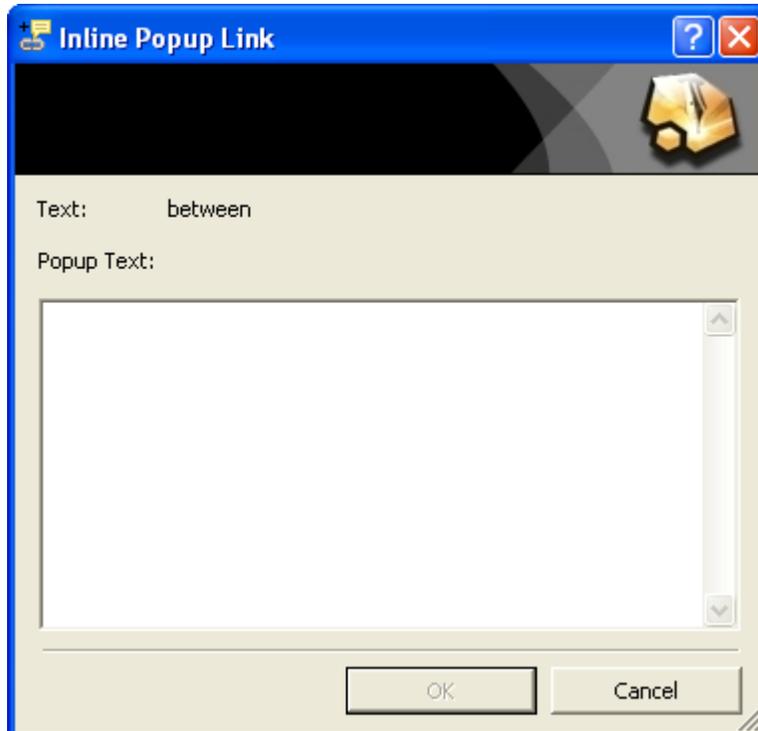
D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Inline Popup (Hidden=False)	<code> </code>	C1H Inline Popup

The Behavior property of the Popup Text hot spot type is set to **Inline Text**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Popup Text (Hidden=False)	 	C1H Popup Text

No properties are necessary for these hot spot types.

To create an Inline Popup hot spot, select the link text and press **Inline Popup**  on the **D2HML Styles** toolbar. Enter the pop-up text within the **Inline Popup Link** dialog box.



If you are not using the toolbar, place the link text within the corresponding HTML `` tag, and place the pop-up text within the corresponding HTML `` tag to format them with these D2HML style.

The following table gives you some examples of using the Inline Text hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
<i>My Link Text</i> <i>My PopupText</i>	<code>My Link TextMy Popup Text</code>	When My Link Text is clicked, My Popup Text appears in a pop-up box below My Text.

Keyword and Invisible Keyword Hot Spots

The Keyword and Invisible Keyword hot spots assign a keyword or list of keywords to the topic containing the hot spot. The keyword appears in the index, and when it is clicked, it will jump to the topic containing this hot spot. The difference between the two hot spot types is that the text of the Invisible Keyword hot spot does not appear in the help target.

The Behavior property of these hot spot types is set to **Keyword**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

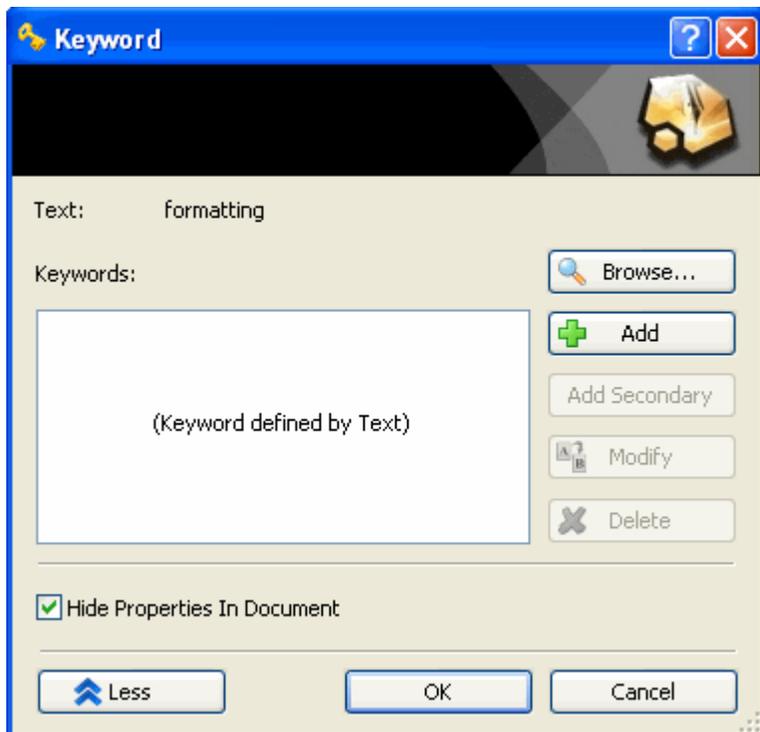
D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Index (Hidden=False)		C1H Index
C1H Index Invisible (Hidden=True)		C1H Index Invisible

Note: The **AutoIndex** property of these character styles is set to **True**.

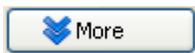
The following properties are supported by this hot spot type:

Supported Properties:	Description
Keyword	<p>This is the keyword or list of keywords to assign to the topic containing this hot spot. You can create keywords, if necessary. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive

To create a Keyword hot spot, select the link text and secondary part, if it exists, and press **Keyword**  or **Invisible Keyword**  on the **D2HML Styles** toolbar. Use the **Keyword** or **Invisible Keyword** dialog boxes to specify the properties of the link.



Keyword Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following table gives you some examples of using the Keyword hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
My Topic	My Topic	The topic containing this hot spot is associated with the keyword My Topic ; the keyword is created if it does not already exist.
My Topic =Keyword	My Topic =Keyword	The topic containing this hot spot is associated with the keyword Keyword ; the keyword is created if it does not already exist.
My Topic keyword=Keyword	My Topic keyword=Keyword	The topic containing this hot spot is associated with the keyword Keyword ; the keyword is created if it does not already exist.
My Topic keyword =Keyword1+Keyword2	My Topic keyword =Keyword1+Keyword2	The topic containing this hot spot is associated with the keywords Keyword1 and Keyword2 ; the keywords are created if they do not already exist.
My Topic Keyword1+Keyword2, Secondary keyword	My Topic Keyword1+Keyword2, Secondary keyword	The keywords Keyword1 , Keyword2 and Secondary keyword are associated with the topic containing this hot spot. Secondary keyword is a child (second-level) keyword with parent Keyword2 . The keywords are created if they do not already exist.
My Topic keyword=Keyword1	My Topic keyword=Keyword1	The topic containing this hot spot is associated with the keyword Keyword1 ; the keyword is created if it does not already exist. My Topic keyword=Keyword1 is not visible in the help target.

Remarks:

In a second-level keyword (one containing a comma), only the child keyword is associated with the topic. If you need to associate the parent keyword with the topic, include it in the list as a separate first-level keyword.

Group and Invisible Group Hot Spots

The Group and Invisible Group hot spots add the topic containing the hot spot to a group or list of groups. This allows the topic to become a destination for a group link. The difference between the two hot spot types is that the text of the Invisible Group hot spot does not appear in the help target.

The Behavior property of these hot spot types is set to **Group**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

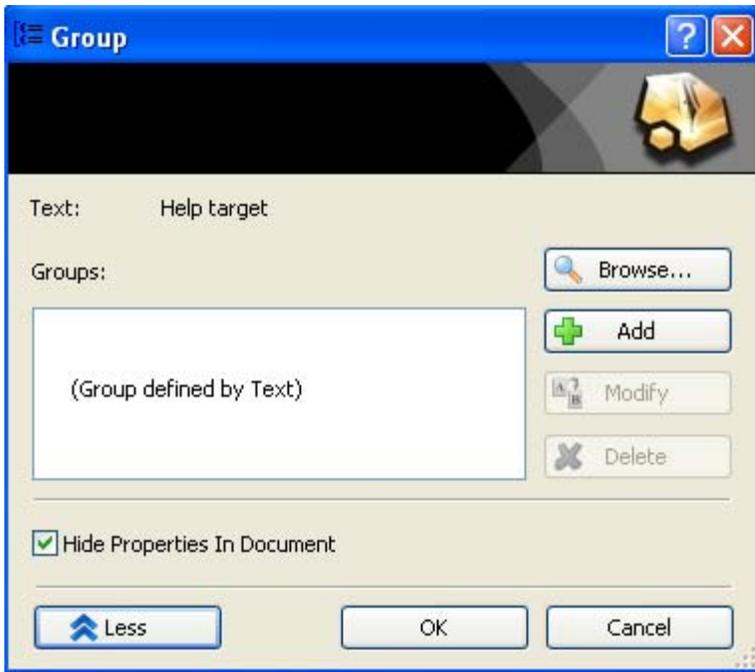
D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Group (Hidden=False)		C1H Group
C1H Group Invisible (Hidden=True)		C1H Group Invisible

The following properties are supported by this hot spot type:

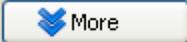
Supported Properties:	Description
Group	<p>This is the list of groups in which to include the topic containing this hot spot. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive

To create a Group hot spot, select the link text and secondary part, if it exists, and press **Group**  or **Invisible**

Group  on the **D2HML Styles** toolbar. Use the **Group** or **Invisible Group** dialog box to specify the properties of the link.



Group Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML `` tag to format it with the D2HML style.

The following table gives you some examples of using the Group hot spot. Each of the following hot spots appears as My Topic in the help target. The secondary part is not visible.

Hot Spot in Word and HTML Editor	HTML Code	Description
<i>My Topic</i>	<code>My Topic</code>	The topic containing this hot spot is included with the group My Topic ; the group is created if it does not already exist.
<i>My Topic Group</i>	<code>My Topic Group</code>	The topic containing this hot spot is included with the group Group ; the group is created if it does not already exist.
<i>My Topic group=Group</i>	<code>My Topic group=Group</code>	The topic containing this hot spot is included with the group Group ; the group is created if it does not already exist.
<i>My Topic group=Group1+Group2</i>	<code>My Topic group=Group1+Group2</code>	The topic containing this hot spot is included with the groups Group1 and Group2 ; the groups are created if they do not already exist.
<i>My Topic group=Group1</i>	<code>My Topic group=Group1</code>	The topic containing this hot spot is included with the group Group ; the group is created if it does not already exist. My Topic group=Group1 is not visible in the help target.

Link Tag and Invisible Link Tag Hot Spots

The Link Tag and Invisible Link Tag hot spots assign a link tag to the topic containing the hot spot. Link tags are automatically created for topic headings formatted with a paragraph style that has its AutoLink property set to True in the Doc-To-Help project. If you want to create additional link tags for the topic; the Link Tag and Invisible Link Tag hot spots allow you to do so. The difference between the two hot spot types is that the text of the Invisible Link Tag hot spot does not appear in the help target.

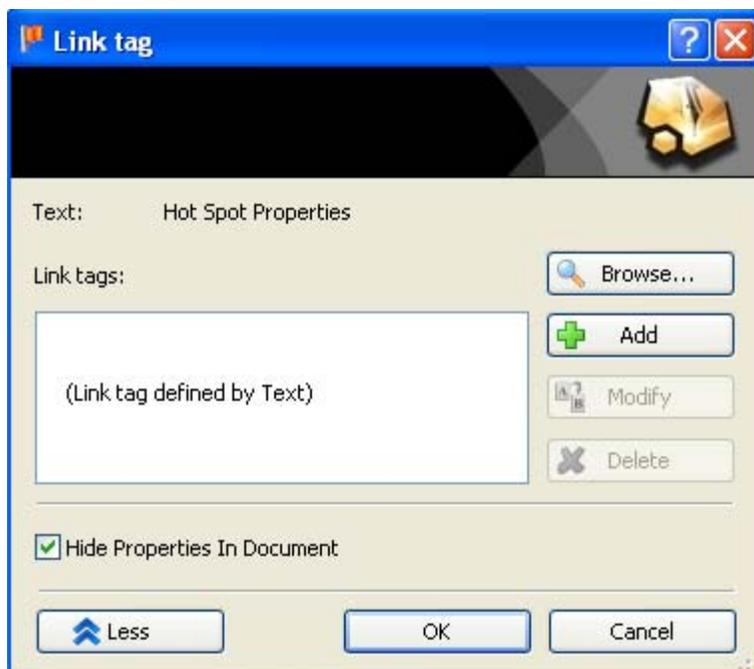
The Behavior property of these hot spot types is set to **Link Tag**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Link Tag (Hidden=False)		C1H Link Tag
C1H Link Tag Invisible (Hidden=True)		C1H Link Tag Invisible

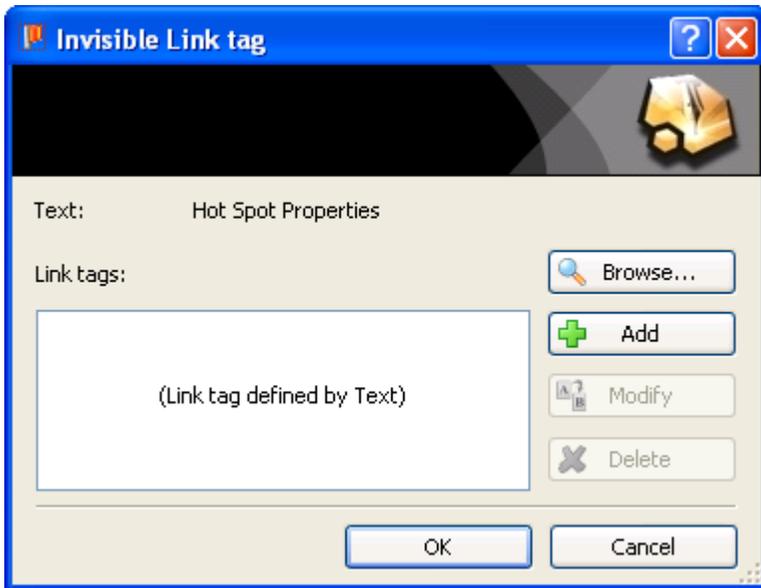
The following properties are supported by this hot spot type:

Supported Properties:	Description
linktag	<p>This is the link tag assigned to the topic containing this hot spot. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive

To create a Link Tag hot spot, select the link text and secondary part, if it exists, and press **Link Tag**  or **Invisible Link Tag**  on the **D2HML Styles** toolbar. Use the **Link Tag** or **Invisible Link Tag** dialog box to specify the properties of the link tag.



Link Tag Dialog Box



Invisible Link Tag Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML `` tag to format it with the D2HML style.

The following table gives you some examples of using the Link Tag hot spot. Each of the following hot spots appears as My Topic in the help target. The secondary part is not visible.

Hot Spot in Word and HTML Editor	HTML Code	Description
My Topic	<code>My Topic</code>	The topic containing this hot spot is assigned the link tag My_Topic .
My Topic	My <code>Topic</code>	If the Link Tag style is applied only to the letters Top in MyTopic, Top becomes a hot spot, and the topic containing this hot spot is assigned the link tag Top .
My Topic linktag=Tag	<code>My Topic linktag=Tag</code>	The topic containing this hot spot is assigned the link tag Tag .
My Topic linktag=Tag1+Tag2	<code>My Topic linktag=Tag1+Tag2</code>	The topic containing this hot spot is assigned link tags Tag1 and Tag2 .
My Topic =Tag	<code>My Topic =Tag</code>	The topic containing this hot spot is assigned the link tag Tag . Because linktag is the default property for the LinkTag hot spot, it does not need to be included in the secondary part.

Context ID Hot Spot

The Context ID hot spot assigns a context ID to the topic containing this hot spot. This allows the topic to be used in context-sensitive help. For additional information on context ID numbers and context-sensitive help, please see [Context Sensitive Help](#) (page 297).

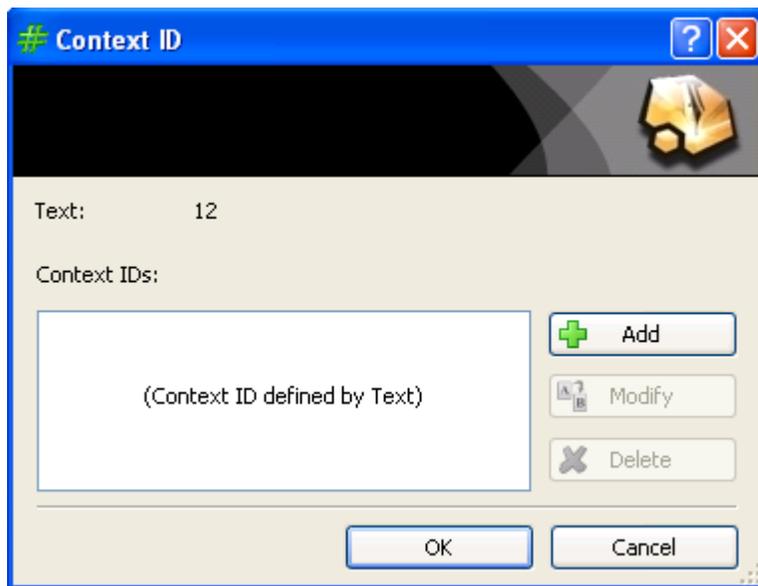
The Behavior property of this hot spot type is set to **Context ID**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Context ID (Hidden=False)	<code></code>	C1H Context ID

The following properties are supported by this hot spot type:

Supported Properties:	Description
ContextID	ContextID value assigned to the topic containing this hot spot. The characteristics of this property are: <ul style="list-style-type: none"> • Primary • Default • Collection • Numeric

To create a Context ID hot spot, select the link text and secondary part, if it exists, and press **Context ID**  on the **D2HML Styles** toolbar. Use the **ContextID** dialog box to specify the properties of the link.



ContextID Dialog Box

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following table gives you some examples of using the Context ID hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
12	12	The ContextID of the topic containing this hot spot is set to 12 .
=11+14+16	11+14+16	The ContextIDs of the topic containing this hot spot are set to 11 , 14 and 16 .
Text contextid=3+6 contextid=3+6	Text contextid=3+6	The ContextIDs of the topic containing this hot spot are set to 3 and 6 .
Text =3+6	Text =3+6	The ContextIDs of the topic containing this hot spot are set to 3 and 6 .

Contents Title Hot Spot

The Contents Title hot spot assigns a title you want to appear in the table of contents to the topic containing this hot spot. For example, if you have a topic My New Topic, and you would like the table of contents (TOC) entry for this topic to read New Topic, you can use the contents title hot spot to set the TOC title. You can also achieve this by using the [ContentsTitle](#) property in the Doc-To-Help Project Editor.

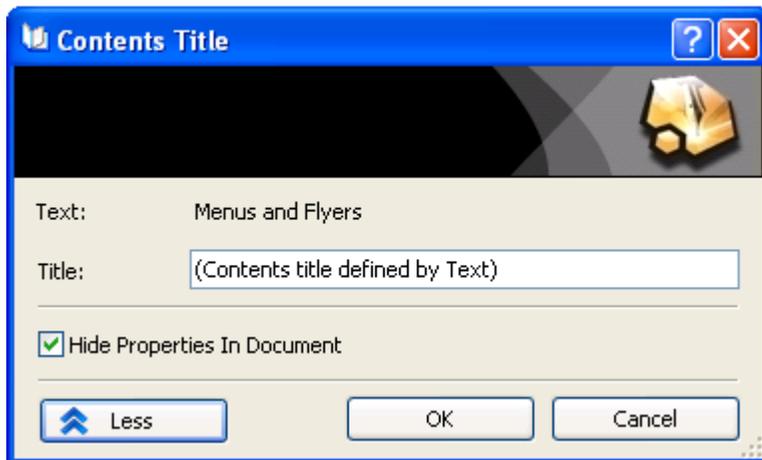
The Behavior property of this hot spot type is set to **Contents Title**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Contents Title (Hidden=False)	 	C1H Contents Title

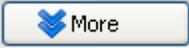
The following properties are supported by this hot spot type:

Supported Properties:	Description
ContentsTitle	The title of the topic used in TOC. The characteristics of this property are: <ul style="list-style-type: none"> • Primary • Default

To create a Contents Title hot spot, select the link text and secondary part, if it exists, and press **Contents Title**  on the **D2HML Styles** toolbar. Use the **Contents Title** dialog box to specify the properties of the link.



Contents Title Dialog Box

Click the **More** button  to access additional properties. When the **More** button is clicked while using this dialog box in Word, you may select the **Hide Properties in Document** checkbox to hide the properties in your source document. To show the properties, uncheck the **Hide Properties in Document** checkbox, or select **Tools | Options** in Word and on the **View** tab, check the **Hidden text** checkbox.

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following table gives you some examples of using the Contents Title hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
New Title	New Title	The ContentsTitle of the topic containing this hot spot is set to New Title .
New Title =A Title	New Title =A Title	The ContentsTitle of the topic containing this hot spot is set to A Title .
New Title contentstitle=A Title	New Title contentstitle=A Title	The ContentsTitle of the topic containing this hot spot is set to A Title .

Topic Properties Hot Spot

The Topic Properties hot spot allows you to set a variety of properties for your topics. Most of these properties can also be found in the Doc-To-Help Project Editor by clicking the Topics icon and selecting a topic. By using the Topic Properties hot spot, you do not need to set the property in the Project Editor.

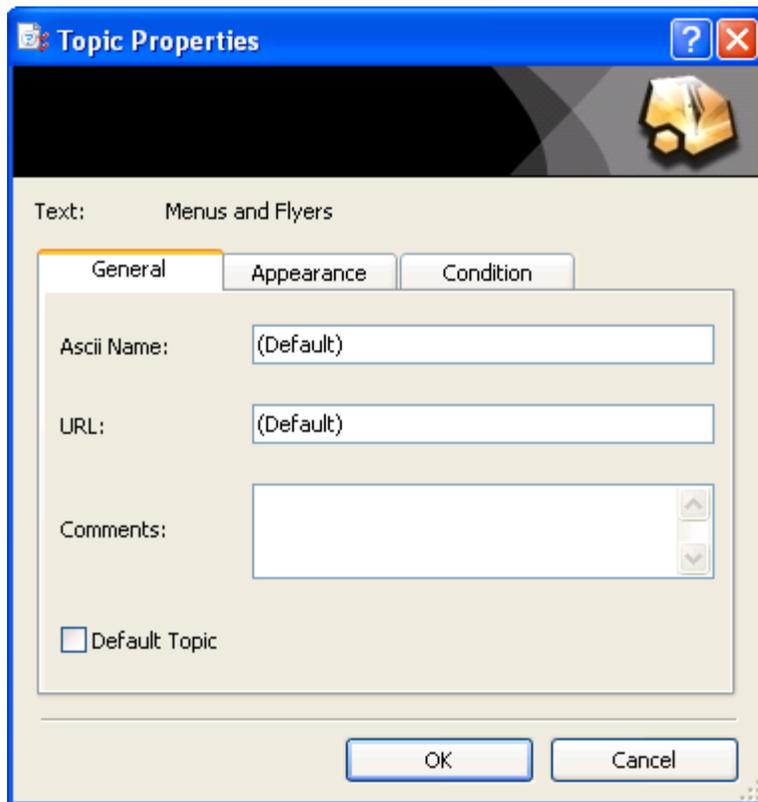
The [Behavior](#) property of this hot spot type is set to **Topic Properties**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Topic Properties (Hidden=True)		C1H Topic Properties

To create a Topic Properties hot spot, select the link text and secondary part, if it exists, and press **Topic Properties**



on the **D2HML Styles** toolbar. Use the **Topic Properties** dialog box to specify the properties of the topic.



Topic Properties Dialog Box

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following properties are supported by the **Topic Properties** hot spot, and they can be found in the D2H Project Editor as well as in the **Topic Properties** dialog box tab specified below:

Property	Description	Tab
AsciiName	This is the value of the Topic.AsciiName property. It specifies the ASCII-only string used to identify the topic in situations where non-ASCII characters are not allowed. The characteristics of this property are: <ul style="list-style-type: none"> • String 	General
ButtonLabel	This is the value of the Topic.ButtonLabel property. It overrides the Help target ButtonLabel text on a topic by topic basis. The characteristics of this property are: <ul style="list-style-type: none"> • Case-sensitive • String 	Appearance

Property	Description	Tab
Comments	<p>This is the value of the Topic.Comments property. When activated, this property provides an editable textbox to be used for comments by the help author. These comments are not accessible by the end user. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Case-sensitive • String 	General
ContentsTitle	<p>This is the value of the Topic.ContentsTitle property. This property allows for modification of a topic title in the help window contents area without changing the actual topic title in the source document. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Case-sensitive • String 	Appearance
DisplayTitle	<p>The value of the Topic.DisplayTitle property. This property allows for modification of a topic title with respect to help file searches without changing the actual topic title in the source document. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Case-sensitive • String 	Appearance
SearchEnabled	<p>This is the value of the Topic.SearchEnabled property. It controls whether ComponentOne Natural Search is enabled for a Help target, document, or topic. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Boolean value • True or False 	Appearance
URL	<p>This is the value of the Topic.URL property. It specifies the name of the generated .htm file for a topic. The characteristics of this property are:</p> <ul style="list-style-type: none"> • String 	General
Default	<p>If this value is set to True, the topic becomes the default topic in all Help targets. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Boolean value • True or False 	General
Attribute	<p>A list of attributes in <i>Name:Value</i> format. The topic is included in the target if these attribute values are included in that target's Attributes collection. The characteristics of this property are:</p> <ul style="list-style-type: none"> • String • Collection 	Condition - use the Add button to open the Condition window and select one of the attributes.
Platform	<p>This is a list of platforms in which the topic is included. The characteristics of this property are:</p> <p>Supported values:</p> <p><i>Manual</i></p> <p><i>Online</i></p> <p><i>HTMLAny</i></p> <p><i>NetHelp</i></p> <p><i>HTMLHelp</i></p> <p><i>MShelp</i></p> <p><i>JavaHelp</i></p> <p><i>WinHelp</i></p>	Condition – use the Add button to open the Condition window and select one of the platform options.
Target	<p>This is a list of targets in which the topic is included. The characteristics of this property are:</p> <ul style="list-style-type: none"> • String • Collection 	Condition – use the Add button to open the Condition window and select one of the help target options.

The following table gives you some examples of using the Topic Properties hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
Platform=Manual	 Platform=Manual	The topic containing this hot spot appears in the manual help target only.
default=true	 default=true	The topic containing this hot spot becomes the default topic in all help targets.
buttonlabel=Next	 buttonlabel=Next	The word Next precedes the subtopic buttons under the topic containing this hot spot.

Here is an example of the Topic Properties hot spot:

|Platform=HTMLHelp;asciiName=mytopic;buttonlabel=SeeAlso;Comments=none;contentTitle=My Topic;default=true;displayTitle=Found my topic;searchEnabled=false;url=myurl.htm;

A topic with this hot spot has the following properties:

- This topic is included in **HTML Help** targets only.
- Topic.AsciiName = “**mytopic**”
- Topic.ButtonLabel = “**See Also**”
- Topic.Comment = “**none**”
- Topic.ContentTitle = “**My Topic**”
- This topic becomes the default topic for all targets.
- Topic.DisplayTitle = “**Found My Topic**”
- This topic is excluded from Natural Search.
- Topic.URL = “myurl.htm”

Conditional Text Hot Spot

The Conditional Text hot spot specifies the hot spot text to be used in various platforms and help targets. For example, you can use a conditional text hot spot to make your text appear in online help only. Or you may want certain text to only appear in a help target you have named *MyHelpTarget*.

The Behavior property of this hot spot type is set to **Conditional Text**. The following predefined styles have this Behavior, and the corresponding HTML tag or Word Style can be used when formatting without the **D2HML Styles** toolbar:

D2HML Style	Corresponding HTML Tag	Corresponding Word Style
C1H Online (Hidden=False)		C1H Online
C1H Manual (Hidden=False)		C1H Manual
C1H Passthrough HTML (Hidden=False)	N/A	C1H Passthrough HTML
C1H Conditional (Hidden=False)		C1H Conditional

Note: All styles that have a non-empty Style.[Condition](#) property have the Conditional Text behavior.

The following properties are supported by this hot spot type:

Supported Properties:	
Attribute	<p>This is the list of attribute pairs in <i>Name:Value</i> format that must be defined in the help target to include this text in the target. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Collection • Case-insensitive

Supported Properties:	
Platform	<p>This is the list of help platforms in which the conditional text is included. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Default • Collection • Case-insensitive <p>The possible values of this property are:</p> <ul style="list-style-type: none"> • Manual • Online • HTMLAny • HTML • HTMLHelp • MSHelp • JavaHelp • WinHelp
Target	<p>This is the list of names of the help targets in which the conditional text is included in the target. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Primary • Collection • Case-insensitive
Passthrough	<p>This property is supported in Word source documents only. If set to True, this property defines the text as passthrough HTML for HTML-based targets. The characteristics of this property are:</p> <ul style="list-style-type: none"> • Secondary • Case-insensitive

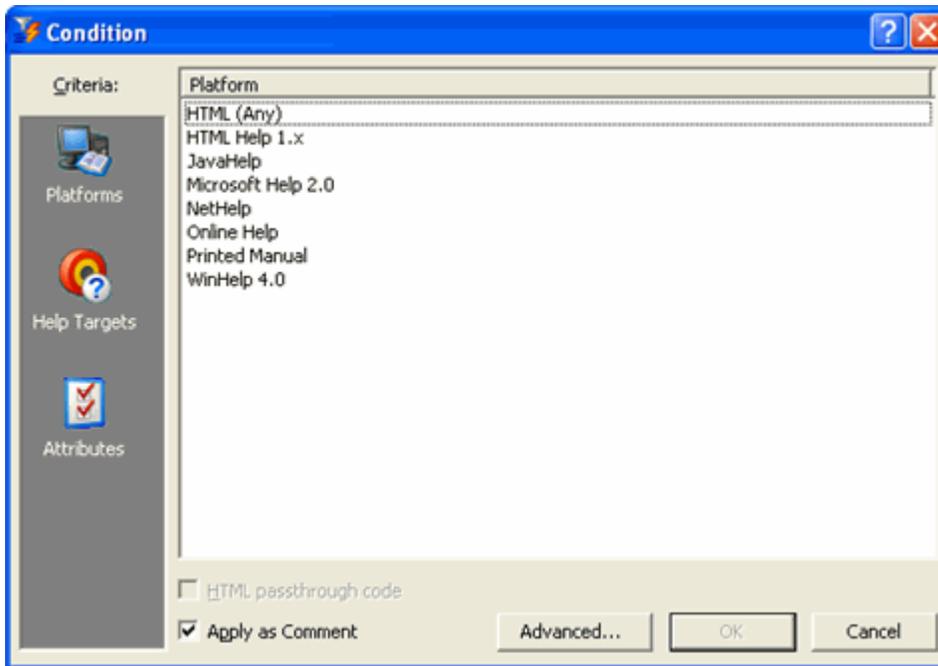
All styles above, except **C1H Conditional**, have a non-empty Style.Condition property value. This means that the user can apply these styles to the text without entering any conditions. The **C1H Conditional** style is used in cases where the user needs to define a condition explicitly in each hot spot. A condition must be specified when the C1H Conditional style is used in order for the hot spot to work correctly.

Conditional text is not removed from a target if the value of one of the primary properties defines a condition satisfied for the current help target. This condition can be specified locally in each conditional text hot spot using one of the primary properties, but that is only needed in advanced cases. Usually, the condition is associated with the style itself, so the user does not need to specify the condition in each hot spot.

To associate the condition with a style, use the Style.**Condition** property. This string property has the same value format as the secondary part of a conditional text hot spot. If the style has a non-empty **Condition** property and the hot spot has secondary part, the primary properties in the hot spot's secondary part have higher priority. For example, if the hot spot sets **Platform=WinHelp**, and Style.**Condition = Platform=Online**, then the effective platform will be **WinHelp**.

All primary properties are collections, so the user can specify more than one value. The condition is satisfied (so the conditional text is not removed) if it is satisfied for at least one value. For example, with condition **Platform=HTMLHelp+WinHelp** the conditional text will appear in **WinHelp** and **HTMLHelp** target.

To create a Conditional Text hot spot, select the link text and secondary part, if it exists, and press either **Online Text** , **Manual Text** , or **Conditional Text**  on the **D2HML Styles** toolbar. Dialog boxes are not necessary when using the **Online Text** and **Manual Text** buttons. Clicking the **Conditional Text** button opens the **Condition** window, where you can specify your conditions.



Condition Window

Conditional text may consist of more than one paragraph with style formatting. Microsoft Word does not allow you to apply style formatting within conditional text because it is already formatted with the **C1H Conditional** style. By default, Doc-To-Help applies conditional text as a comment rather than a D2HML style in your Word documents so that style formatting may be used within the comment. For more information, see [Specifying Conditional Text](#) (page 401).

If you are not using the toolbar, place the link text and secondary part in the corresponding HTML tag to format it with the D2HML style.

The following table gives you some examples of using the Conditional Text hot spot:

Hot Spot in Word and HTML Editor	HTML Code	Description
Online text	<code>Online text</code>	The text formatted with this style is removed from Manual targets only.
Manual text	<code>Manual text</code>	The text formatted with this style is displayed in Manual targets only.
Conditional text platform=winhelp	<code>Conditional text platform=winhelp</code>	The text Conditional text is displayed in WinHelp targets only.
Conditional text =winhelp	<code>Conditional text =winhelp</code>	The text Conditional text is displayed only in WinHelp targets only.
Conditional text attribute=build:release	<code>Conditional text attribute=build:release</code>	The text Conditional text is displayed only in targets that have the attribute value Build set to Release. Please note, the separator bar () and everything to the right of it is always removed from the target.
Conditional text target=my html help;passthrough=true	<code>Conditional text target=my html help;passthrough=true</code>	The text Conditional text is displayed in the target with the name "My HTML Help", and it is passthrough HTML.

Showing Hidden Hot Spots

Invisible hot spot types, such as Invisible Keyword and Topic Properties, are not visible in help targets and Word source documents. The **Show Hidden Hot Spots** button  available on the **D2HML Styles** toolbar in Word allows you to show any invisible hot spots in your source document, making it easier to see all formatted text.

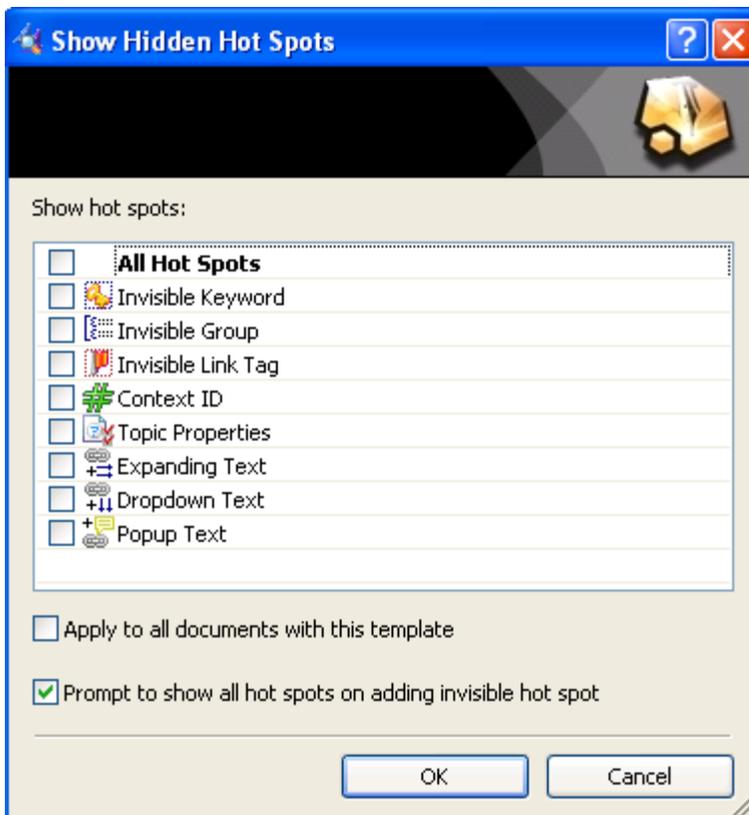
Note: The **Show Hidden Hot Spots** button is not available in the FrontPage **D2HML Styles** toolbar, because all styles are visible in HTML source documents in **Design** view. Click the **Preview** tab to see how the styles will look in the help target.

The following hot spot types are invisible by default in Word:



To make your hidden hot spots visible:

- Click the **Show Hidden Hot Spots** button  on the **D2HML Styles** toolbar. The **Show Hidden Hot Spots** dialog box appears.



- Select **All Hot Spots**, or you can choose the individual hot spot types you would like to make visible.
- Select **Apply to all documents with this template** if you would like your selections in this dialog box to apply to any other Word documents that have the same template attached.

- You also have the option of being prompted to show all hidden hot spots when you format text as one of the invisible hot spot types. The **Prompt to show all hot spots on adding invisible hot spot** is checked by default. In Word, when you format text as one of the invisible hot spot types, the following dialog box opens:

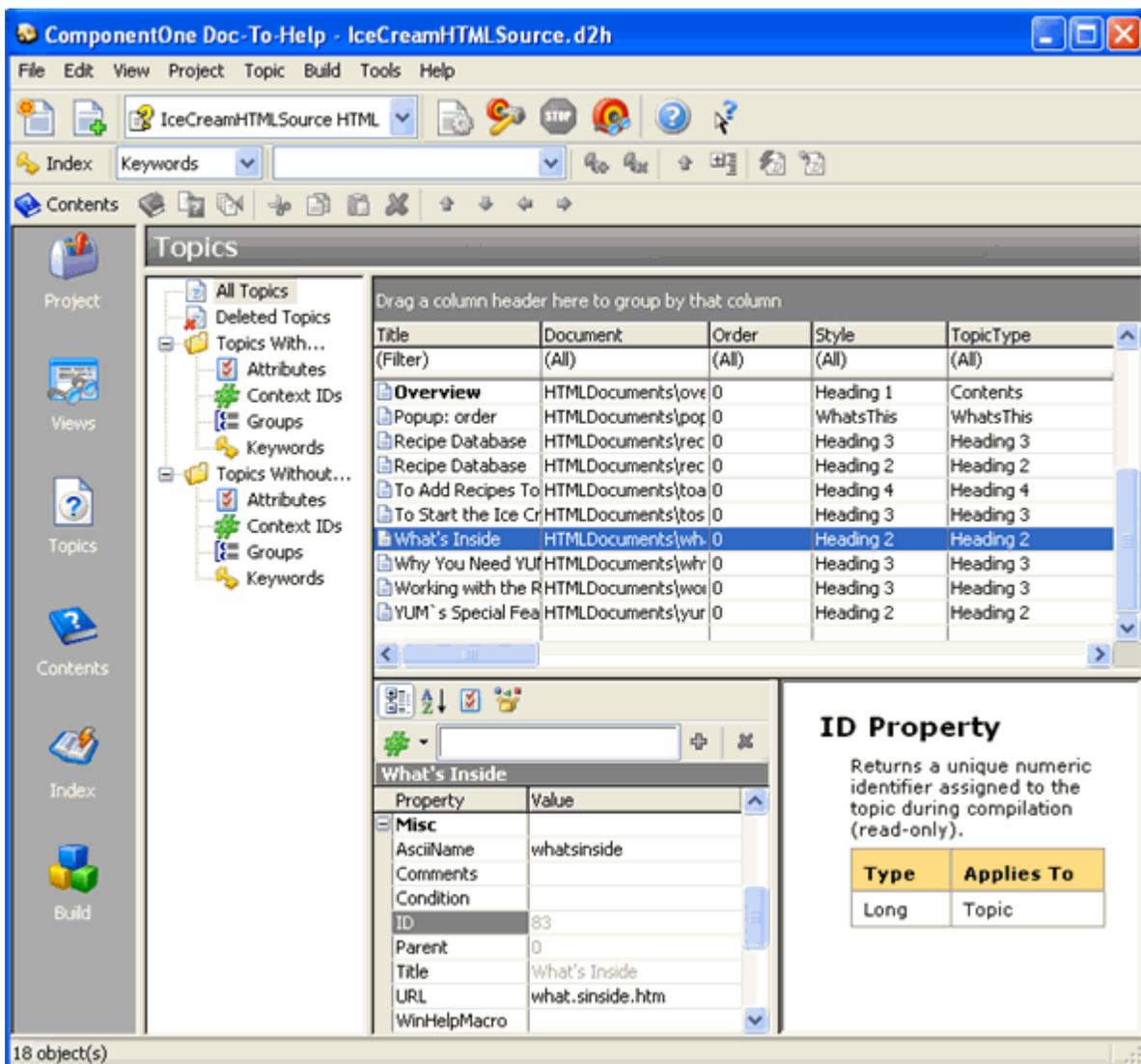


- Click **Yes** to show hidden hot spots. If you choose **No**, you can always make them visible later by using the **Show Hidden Hot Spots** button and dialog box. If you do not want this dialog box to appear, and you do not want to make all hidden hot spots visible, uncheck the **Prompt to show all hot spots on adding invisible hot spot** checkbox in the **Show Hidden Hot Spots** dialog box.

Defining and Organizing Topics

The **Topics** Icon displays a grid view of the Help topics derived from the source documents during compilation. It also provides an interface for setting properties of individual topics, such as index keywords and context-sensitive Help IDs. The left hand pane of the project editor contains a tree list allowing the author to view specific types of topics. You can view all topics, deleted topics or topics with and without Attributes, Context IDs, Groups and Keywords. Additionally, you can sort the topics in the right project editor pane by using the convenient outlook style grouping and filtering features provided with ComponentOne Doc-To-Help 2006.

Doc-To-Help derives topics from your source documents. In multiple topic documents, there is not an explicit way for you to create a new topic from scratch. Instead, you create topics indirectly by applying a paragraph style to one or more source document paragraphs in Microsoft Word or your HTML editor, defining a like-named active paragraph style in your Doc-To-Help project, if one does not already exist, then compiling the affected documents either individually or by building the Help target. After compilation, you can view the newly created topics and edit their properties in the **Topics** window. With single topic documents, a new topic is created in your Doc-To-Help project when you create a new document. You can view the new topic and its properties immediately once it is created; you do not need to recompile the document.



The term *active paragraph style* denotes a style that Doc-To-Help uses to define topics. Any style with a numeric outline level is active. For example, the predefined Doc-To-Help styles *Heading 1*, *Heading 2*, and *Heading 3* are active, while,

for example, the built-in Word styles *Normal*, *Body Text*, and *List Bullet* are not. If you have your own document templates or cascading style sheets that use other styles to format topic titles, you can import them into a Doc-To-Help project so that they are recognized during compilation.

A topic from single topic document is automatically available when it is added to the project. Topics from a multiple topic document do not formally exist in a Doc-To-Help project file until you build a Help system. During a build, each Word source document is compiled as follows:

1. Doc-To-Help saves a copy of the document, which is placed in a folder created by Doc-To-Help. Your original document is never modified. Once created, the copy is not modified, although Doc-To-Help may delete it during subsequent builds. This applies to all source documents, single and multiple topic.
2. For each paragraph in the document, Doc-To-Help compares the name of the associated style against the names of the active paragraph styles defined in the Doc-To-Help project. This applies to multiple topic Word or HTML source documents.
3. For each paragraph that matches an active style, Doc-To-Help adds a topic record to the project file using the text of the entire paragraph as the topic title. (If the topic was already encountered during a prior build, Doc-To-Help does not add a new record.) This applies to multiple topic Word or HTML source documents.
4. Doc-To-Help generates one or more additional intermediate files in a format that is appropriate for the current Help target. These files include suitable mark-up symbols derived from the topic records and other elements of the project file. This applies to all source documents, single and multiple topic.

Once all source documents have been compiled, Doc-To-Help combines all marked-up intermediate files into a single entity. For WinHelp and HTML Help, this step involves running a Microsoft compiler. For other Help targets, Doc-To-Help generates ancillary files (such as HTML framesets) as needed.

Notes: Only the topic title should be formatted with the active paragraph style. In the topic you are now reading, the paragraph above containing the words *Defining and Organizing Topics* is formatted with the *Heading 1* style, but the remaining paragraphs are formatted with the *Normal* style.

Topic titles must be no more than 255 characters long, or else a compile-time error occurs.

Topic titles defined by a paragraph style with a *Level 1* type must be unique within a document. That is, a single document can contain multiple topics named *Example* if each is formatted with a different level 1 paragraph style, but it cannot contain multiple topics named *Example* that are formatted with the same style. The only way to have multiple like-named topics that are formatted with the same style is to place each in a separate source document. These restrictions are enforced at compile time.

Topic Properties

By selecting any of the Topics in the right pane, the property pane displays the property settings for that topic.

Behavior Properties

CancelKey	When True, this property disables the AutoKeyword property on a topic by topic basis.
CancelMap	When True, this property disables the AutoContextID property on a topic by topic basis.
CancelTag	When True, this property disables the AutoLink property on a topic by topic basis.
TopicType	Defines a pre-determined collection of attributes to a topic (i.e. what window the Help topic appears in, how the Help topic is accessed, whether it gets a map number).

Display Properties

ButtonLabel	Specifies the text that precedes the subtopic buttons on a topic by topic basis. Set this property to an empty string to suppress the default More: label.
ContentsTitle	Allows for modification of the topic titles in the Help window contents area without changing the actual topic title in the source document.
ContextString	When using context-sensitive and dynamic help with Microsoft Help 2.0, specifies a context string for the topic.
DisplayTitle	Allows for modification of the topic title in respect to Help file searches without changing the actual topic title in the source document.

Miscellaneous Properties

AsciiName	Specifies the ASCII-only string used to identify the topic in situations where non-ASCII characters are not allowed.
Comments	When activated, this property provides an editable textbox to be used for comments by the Help author. These comments are not accessible by the end user.
Condition	For a topic, defines a condition for including this topic in a help target.
ID	Returns a unique numeric identifier assigned to the topic during compilation (read-only).
Parent	Shows the ID number of the related parent topic.
Title	Returns the topic title (read-only).
URL	Specifies the name of the generated .htm file for a topic.
WinHelpMacro	Specifies a macro to run when a Help target, a topic, or secondary window is opened.

Modular Properties

MergeContents	For a placeholder topic in a modular hub project, this property specifies the platform-dependent contents file to be loaded dynamically.
MergeFile	For a placeholder topic in a modular hub project, this property specifies the platform-dependent Help file to be loaded dynamically.
MergeTitle	For a placeholder topic in a modular hub project, this property specifies the text used to clarify like-named topics in keyword search lists.

Natural Search

SearchEnabled	Controls whether ComponentOne Natural Search is enabled for a Help target, document, or topic.
-------------------------------	--

Navigation

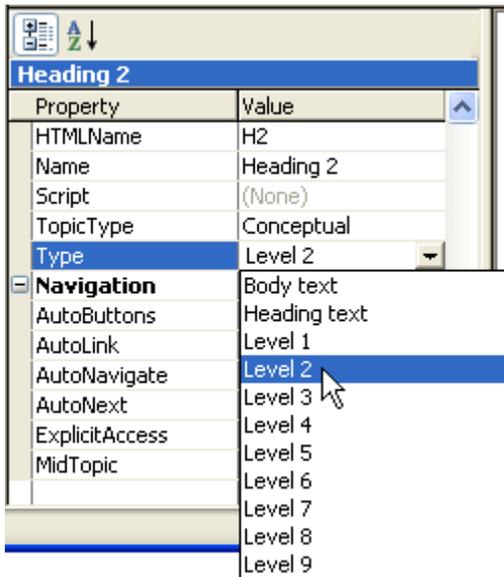
RelatedJumps	Controls the order of automatic and/or custom jumps on a topic by topic basis.
------------------------------	--

Making a Paragraph Style Active

In order for a paragraph style used in a source document to define a topic during compilation, it must be marked as active within the Doc-To-Help project file as follows:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. In the right pane, select the name of the style to be edited.

- In the property pane, scroll down to the [Type](#) property and set the heading type to a value other than **Body text**.



When you create a new paragraph style, the default outline level is *Body text*. The *Body text* setting is the only one that does not mark a paragraph style as active. Like the numeric level settings, it is analogous to its counterpart in Microsoft Word, FrontPage or Dreamweaver, and is provided so that you can easily import styles from a Word or HTML document, template or style sheet without turning every paragraph into a separate topic. You can also use this setting to disable an active style. For example, if your Help system does not need three levels of topics, you can set the outline level of *Heading 3* to *Body text*.

The *Heading text* style makes the style active but excludes associated topics from the Help contents. This setting is typically used to implement ancillary pop-up topics such as figures, glossary terms, and code samples.

The settings *Level 1* through *Level 9* also make the style active. In addition, these settings determine the hierarchy of associated topics within the Help contents. The predefined Doc-To-Help styles *Heading 1*, *Heading 2*, and *Heading 3* are set to *Level 1*, *Level 2*, and *Level 3*, respectively.

For more information on paragraph styles, see [Using Paragraph Styles](#) (page 211).

Specifying the Help Contents

Initially, Doc-To-Help derives the Help contents automatically based upon the hierarchy of topics in the project. Therefore, in order for a topic to be included in the Help contents, the paragraph style that defines it must have a numeric outline level.

To assign a numeric outline level to a paragraph style:

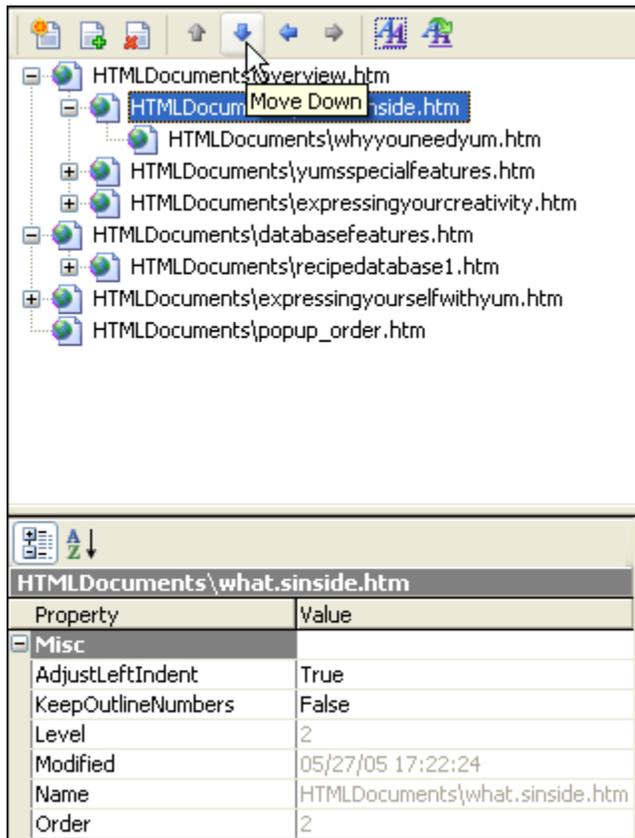
- On the Icon bar, click **Project**.
- Select the **Paragraph Styles** item from the left pane.
- Select the paragraph style to be edited from the right pane.
- In the property pane, change the [Type](#) property to a value between *Level 1* and *Level 9*.

To modify the Help contents using the Project Editor, see [Modifying the Help Contents](#) (page 291).

If a project contains multiple source documents, the order in which they were added to the project determines the default order in which they are listed in the Help contents. Since this may not be the order you intended, you can rearrange it. Note that the order of topics in the table of contents can be customized at any time using the [Contents toolbar](#).

To rearrange the order of the documents and therefore the top level Help topics contents:

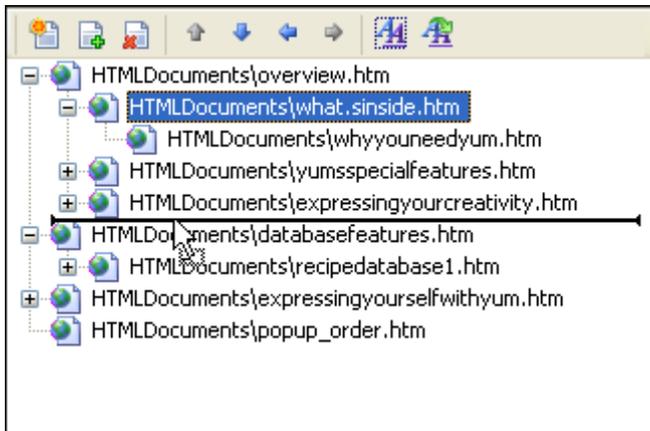
1. On the Icon bar, click **Project**.
2. Choose **Documents** from the left pane.
3. Choose the document you wish to move from the right pane.
4. Change the [Order](#) property by pressing the **Move Up** or **Move Down** arrow located at the top of the documents property pane in the **Documents** toolbar.



You have two other options in addition to using the **Move Up** and **Move Down** arrows when changing the order of your documents:

Drag and Drop

1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Select the source document you wish to move.
4. While holding the left mouse key, drag the document to the desired position within the document tree.



Cut and Paste

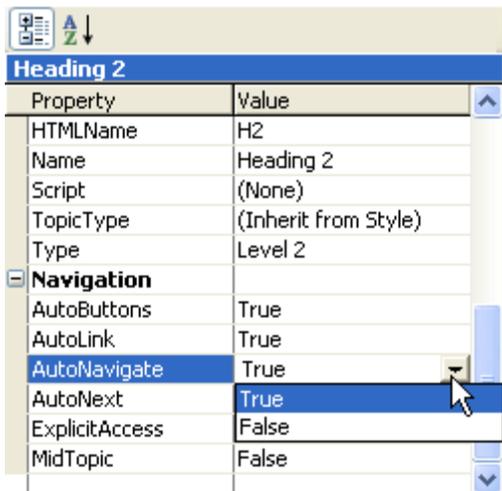
1. On the Icon bar, click **Project**.
2. Select the **Documents** item from the left pane.
3. Right-click the source document you wish to move and select **Cut** from the context menu.
4. Right-click the document to precede the document you are moving and select **Paste** from the context menu. The document you cut now follows this document in the tree.

Specifying the Navigation Sequence

Doc-To-Help does not provide an interface for defining arbitrary topic navigation sequences, but instead derives them automatically based upon the order of topics in each source document. By default, only paragraph styles with a numeric outline level participate in the navigation sequence.

To control whether topics created with an active style are navigable:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. Select the paragraph style to be edited from the right pane.
4. From the property pane, modify the [AutoNavigate](#) property value as desired. Setting the value to **True** allows the topic with the selected style to be navigable.

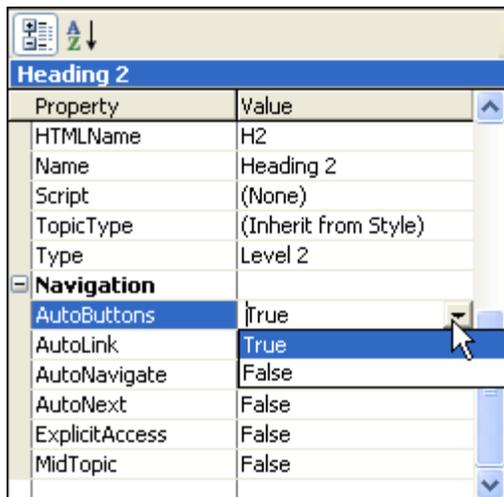


Providing Links to Subtopics

For paragraph styles with numeric outline levels greater than *Level 1*, Doc-To-Help can automatically display a list of subtopic buttons at the end of each parent topic.

To enable subtopic buttons for a paragraph style:

1. On the Icon bar, click **Project**.
2. From the left pane, select **Paragraph Styles**.
3. From the right pane, select the paragraph style to be edited.
4. From the property pane, modify the [AutoButtons](#) property value as desired. Setting the value to **True** allows subtopic buttons to be displayed at the end of the parent topic.

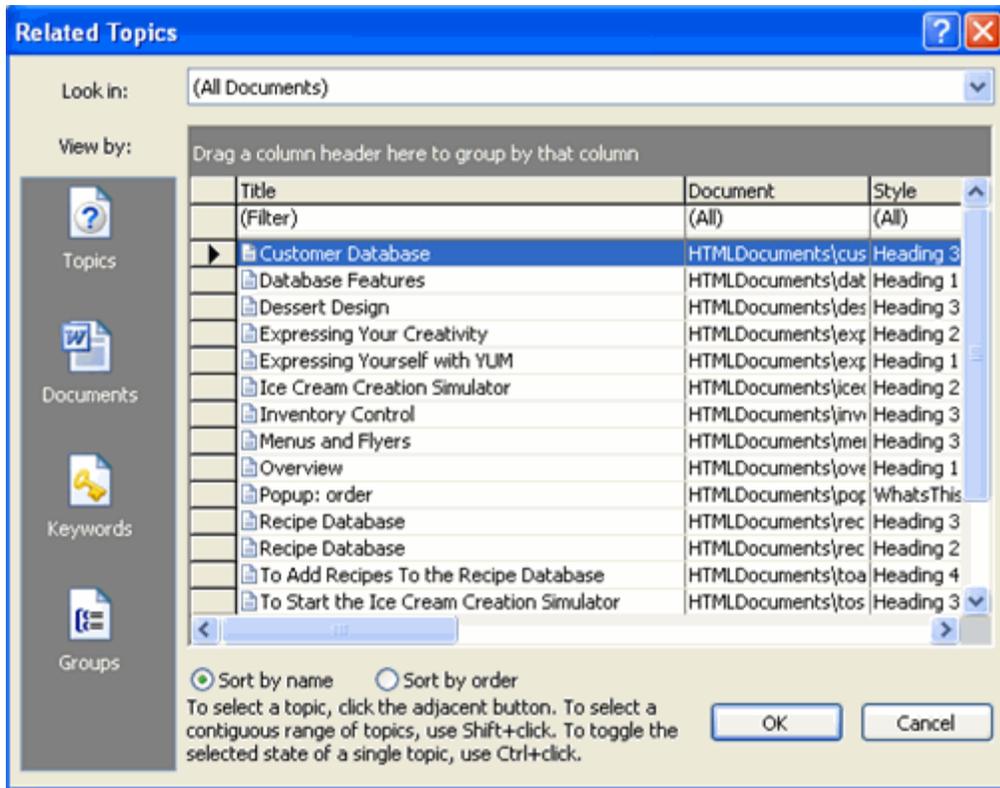


Providing Links to Specific Topics

Along with providing automatic links to subtopics at the end of a parent topic, Doc-To-Help allows you to customize the list of links by using the **Related Topics** command. This command adds additional, author selected topic links, to the end of a parent topic.

To add author selected topic links to the end of a topic:

1. On the Icon bar, click **Topics**.
2. From the right pane, select and right-click the parent topic where you want to add a topic link.
3. From the shortcut pane, click **Related Topics**.
4. From the **Related Topics** dialog box, select the topic you wish to jump to by clicking on one or more of the record selector buttons to the left of the topic titles.



Note: Doc-To-Help supports multiple topic selection by using the **Ctrl** keys. Simply press and hold the **Ctrl** key, for nonconsecutive selections, and choose the record selector buttons to the left of the topic titles.

- Once you have made your selection, click **OK**.
- Upon compilation, the selected topic link is added to the end of the parent topic.

Customizing the Related Topic Jumps

Doc-To-Help provides you with the option to customize which topic links are available at the end of a parent topic. By using the [RelatedJumps](#) property, you can determine whether your automatic and author specified topics are available and in what order they are presented.

To customize the topics:

- On the Icon bar, click **Topics**.
- From the right pane, select the parent topic you wish to edit.
- From the topics properties pane, select [RelatedJumps](#) and click the option you want from the drop-down menu.

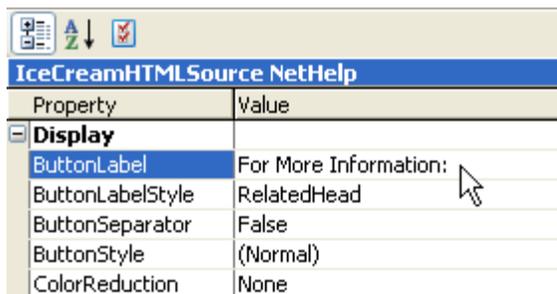
[-] Modular	
MergeContents	
MergeFile	
MergeTitle	None
	Automatic
[-] Natural Search	Automatic+Custom
SearchEnabled	Custom+Automatic
[-] Navigation	Custom
RelatedJumps	Custom+Automatic

Value	Description
None	No Jumps are displayed
Automatic	Only Automatic jumps are displayed
Automatic + Custom	Automatic then Custom jumps are displayed
Custom + Automatic	Custom then Automatic jumps are displayed
Custom	Only Custom jumps are displayed

Labeling Subtopic Links

In cases where you want to insert a descriptive heading for the subtopic buttons, Doc-To-Help provides the [ButtonLabel](#) property. The default setting for this property is **More:**, but you can modify the default easily. To modify the default topic:

1. On the Icon bar, select **Project**.
2. In the left pane, click **Help Targets**.
3. In the right pane, select the Help Target you want to modify.
4. In the property pane, locate the [ButtonLabel](#) property and enter the text you wish to use as your label to the right of the property name.



The changes you made to the [ButtonLabel](#) property produce results as shown in the illustration below.

For More Information:

- Inventory Control
- Customer Database

Modifying the Subtopic Label for a Specific Topic

In some instances, you may wish to override the Help target's existing [ButtonLabel](#) property, for a single topic.

1. On the Icon bar, select **Topics**.
2. In the right pane, select the topic you want to modify.
3. In the property pane, locate the [ButtonLabel](#) property and enter the text you wish to use as your label to the right of the property name.

The screenshot shows the 'Customer Database' property pane. The 'Display' section is expanded, and the 'ButtonLabel' property is selected. The value for 'ButtonLabel' is 'To create your ice cream treat:'. Other properties include 'CancelTag' (False), 'TopicType' (Conceptual), 'ContentsTitle' (Customer Database), 'ContextString', and 'DisplayTitle' (Customer Database).

Property	Value
CancelTag	False
TopicType	Conceptual
Display	
ButtonLabel	To create your ice cream treat:
ContentsTitle	Customer Database
ContextString	
DisplayTitle	Customer Database

The changes you made to the [ButtonLabel](#) property produce results as shown in the illustration below.

To create your ice cream treat:

- Select a flavor
- Add multiple scoops
- Add toppings

Note: When modifying the [ButtonLabel](#) property for a topic, the changes are limited to that topic only. The remaining Subtopic labels will contain the text specified in the Help target [ButtonLabel](#) Property.

Providing Links to the Next Topic

For any active paragraph style, Doc-To-Help can display a button at the end of each associated topic that links to the next topic in the source document, even if the next topic does not participate in the navigation sequence.

To enable the next topic button for a paragraph style:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. Select the paragraph style to be edited from the right pane.
4. From the property pane, modify the [AutoNext](#) property value as desired. Setting the value to **True** allows a button that links to the next topic to be displayed.

The screenshot shows the 'Heading 2' property pane. The 'Navigation' section is expanded, and the 'AutoNext' property is selected. The value for 'AutoNext' is 'False'. Other properties include 'HTMLName' (H2), 'Name' (Heading 2), 'Script' ((None)), 'TopicType' ((Inherit from Style)), 'Type' (Level 2), 'AutoButtons' (True), 'AutoLink' (True), 'AutoNavigate' (True), 'ExplicitAccess' (True), and 'MidTopic' (False).

Property	Value
HTMLName	H2
Name	Heading 2
Script	(None)
TopicType	(Inherit from Style)
Type	Level 2
Navigation	
AutoButtons	True
AutoLink	True
AutoNavigate	True
AutoNext	False
ExplicitAccess	True
MidTopic	False

You can use this setting to chain together a series of topics that are meant to be read in sequence. Although the reader can generally achieve the same effect with a built-in navigation button, providing an explicit link at the end of each topic makes your intentions clear.

You can also use this setting to link master topics with detail topics without affecting the navigation sequence. For example, suppose that a source document contains a set of function descriptions at the *Heading 2* level, each of which is immediately followed by a code example at the *Heading 3* level. If you enable [AutoNext](#) for *Heading 2*, to create links to the examples, disable [AutoNavigate](#) for *Heading 3*, to remove it from the navigation sequence, and assign a secondary window to *Heading 3*, then the reader can step through the function descriptions and click the generated button to view a particular code sample in a separate window.

Note: In HTML Help the [BinaryTOC](#) property must be set to **True** for topic to topic navigation.
In WinHelp the [OneBrowseSequence](#) property must be set to **True** for topic to topic navigation.

Editing Topic Properties

1. On the Icon bar, click **Topics**.
2. Select **All Topics** or any of the other topic choices available from the left pane.
3. In the right pane, select the name of the topic to be edited.

Note that the property pane changes to show the properties of the selected topic.

4. Change the properties as desired.

Note that your changes affect the selected topic instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

For more information, see [Topic Properties](#) (page 280).

Adding a Link Tag to a Topic

In order for a topic to be a destination for a jump or pop-up link, it must have a unique link tag that identifies it. Link tags can be generated automatically by paragraph styles, programmatically by scripts, or manually by the author.

To enable automatic link tags for a paragraph style:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. In the right pane, choose the paragraph style to be edited.
4. In the property pane, set the [AutoLink](#) property value to **True**.

Link tags created in this manner are identical to the topic title except that spaces, hyphens, and periods are converted to underscores; letters, numbers, and underscores are unchanged; and all other characters are removed. For example, the topic title:

BackCol or Property (TextBox)

yields the link tag:

BackCol or_Property_TextBox

Examining Link Tags

In order for a link to a topic to work, one of the following must be true:

- A link tag of the destination topic in Doc-to-Help must exactly match the text of the hot spot in the source document. For example, if you created a hot spot in your source document with the text **help topics**, then the link tag of the destination topic must be **help_topics**.

OR

- A link tag of the destination topic must exactly match the *tag* property of the hot spot if you are using D2HML to create it. For example, if you created a topic link hot spot, **help topics**, with the *tag* property set to *help topics*, then the link tag of the destination topic must be **help_topics**.

When using link tags, spaces, hyphens and periods must be converted to underscores; letters, numbers and underscores are unchanged; and all other characters must be removed. For example, the topic title:

6th St. Exit

yields the link tag:

6th_St__Exit

In this example the topic title contains a period next to a space, so you must provide an underscore for both the period and the space.

To determine the link tag of your destination topic:

1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane.
3. In the right pane, choose the destination topic.
4. Click the **Choose Collection** button on the **Topics Property** toolbar and choose **Link Tag** from the **Collection Type** drop-down.

The link tags are listed next to the **Link Tags** collection.

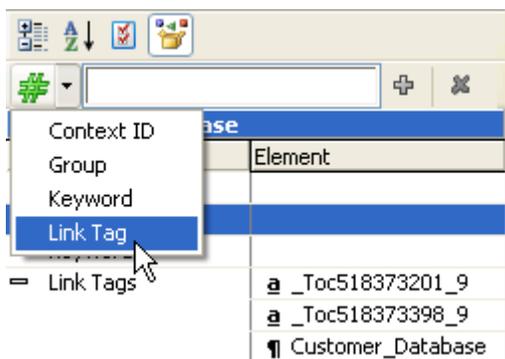
5. Make sure that one of the link tags exactly matches the text of your hot spot. If none does, you must create a matching link tag.

For more information, see [Link Tag and Invisible Link Tag Hot Spots](#) (page 266) or [Manually Inserting Link Tags](#) (page 290).

Manually Inserting Link Tags

To manually insert a link tag:

1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane and select the desired topic for editing.
3. Click the **Choose Collection** button on the **Topics Property** toolbar and choose **Link Tag** from the **Collection Type** drop-down.



4. Type the Link Tag into the **Element Textbox** and press **Enter**.

Note that the Link Tag has been associated with the highlighted topic and is displayed to the right of the Link Tag collection.

When using link tags, spaces, hyphens and periods must be converted to underscores; letters, numbers and underscores are unchanged; and all other characters must be removed. For example, the topic title:

6th St. Exi t

yields the link tag:

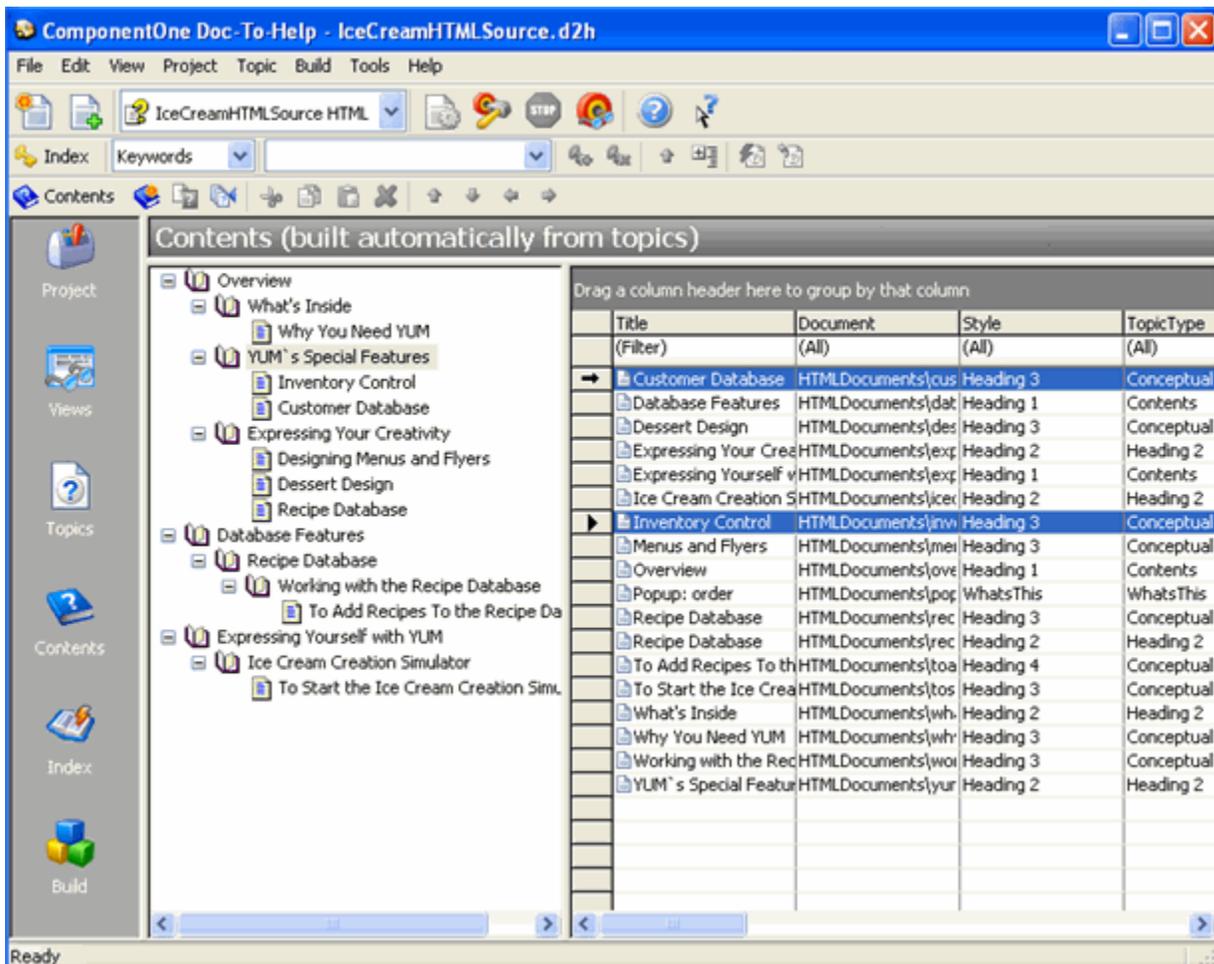
6th_St__Exi t

In this example the topic title contains a period next to a space, so you must provide an underscore for both the period and the space.

Modifying the Help Contents

The **Contents** Icon displays the hierarchy of Help topics as determined by the outline level of the defining styles. This hierarchy is used to build the contents for all online Help targets. Note that the contents is labeled (*built automatically from topics*).

The left pane contains the same books and topics as the Contents section in the compiled Help file, and the right pane displays a sorted list of topics as in the Topics window. The topics can be sorted and filtered in the same manner as in the Topic window. Doc-To-Help supports multiple topic selection by using the *Ctrl* key. Simply press and hold the *Ctrl* key for nonconsecutive selections, and choose the record selector buttons to the left of the topic titles.



Doc-To-Help's Contents Window is an interface in which the final Help system's contents can be fully customized. The ability to redefine the default topic hierarchy, create new books, insert and delete topics, and perform common editing functions is available through this view. The interface utilizes toolbars, drag-and drop editing, and right-click shortcut menus to make the editing of contents easier and you more productive.

Note: Once you have customized the contents, it is no longer automatically updated when you change the topics in your source documents and rebuild the target. The label (*built automatically from topics*) becomes (*customized*).

Using the Contents Toolbar

On the **View** menu, point to **Toolbars**, then click **Contents**. A check mark next to the command indicates that the toolbar is displayed.

After a Doc-To-Help project has been compiled, you can use the **Contents** toolbar to cut, copy, paste, delete, and rearrange contents and the topics.

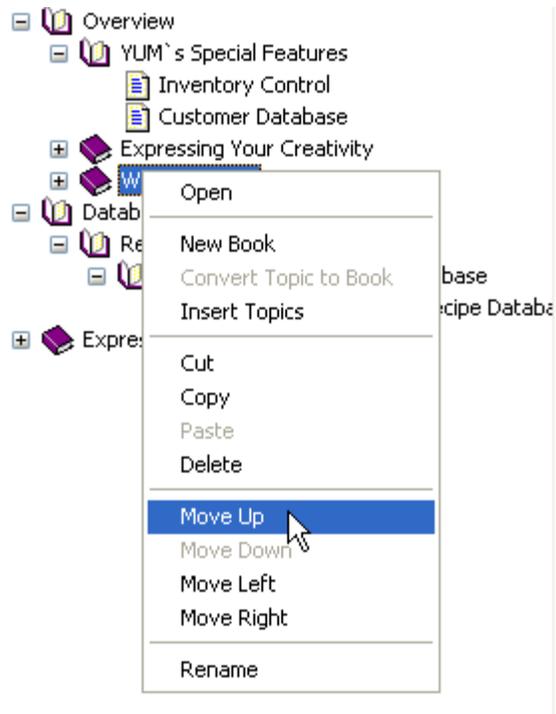


From left to right, the **Contents** toolbar contains the following commands:

New Book	Adds a new book to the Help contents
Convert Topic	Changes the selected topic into a book
Insert Topics	Inserts the selected topic(s) from the right pane into the Help contents
Cut	Removes the selection and places it on the clipboard
Copy	Copies the selection to the clipboard.
Paste	Inserts the contents of the clipboard at the insertion point, replacing any selection
Delete	Deletes the selection
Move Up	Moves the selection up within the outline
Move Down	Moves the selection down within the outline
Move Left	Moves the selection left within the outline
Move Right	Moves the selection right within the outline

Editing the Help Contents

Even though the Help contents can be fully edited in the Contents window through the use of the toolbar, it can also be edited through Doc-To-Help's right-click shortcut menu. Right-clicking a book or a topic brings up the shortcut menu. This menu conveniently opens and provides all of the functions of the toolbar. Using the right-click shortcut menu helps to speed up the time it takes to rearrange and edit your contents.



Rearranging the Help Contents

Upon compilation, the outline for the final Help system is displayed in the left pane of the Contents window. This outline is completely customizable in respect to its structure. The four directional commands on the toolbar or the “move” commands in the shortcut menu are used to alter the outline structure.

To move a topic or book up or down in the outline:

1. Select the topic or book to be moved in the left pane.
2. Use the move up or move down commands either from the **Contents** toolbar or the shortcut menu to change the position of the topic or book.

To move a topic or book right or left in the outline:

1. Select the topic or book to be moved in the left pane.
2. Use the move left or move right commands either from the **Contents** toolbar or the shortcut menu to change the position of the topic or book.

Notice that these commands cannot change a book into a topic or vice versa, but that they merely move the item around the outline.

Creating a New Book

1. In the Contents window, right-click anywhere in the left pane.
2. Choose **New Book** from the menu or select the new book command from the **Contents** toolbar.

A blinking cursor appears and allows you to customize the name of the new contents item.

Notice that if you right-click the dead area, choose new book from the shortcut menu, and no book is selected in the left pane, then Doc-To-Help creates a new book after the last book in the outline.

Converting a Subtopic into a Book

Doc-To-Help also allows you to convert an existing subtopic into a main topic. The topic remains in the same position in the Help system’s outline, but is converted from a topic to a book.

1. In the Contents window, select a subtopic that you would like to be converted into a book.
2. Click the **Convert Topic** command in the toolbar.

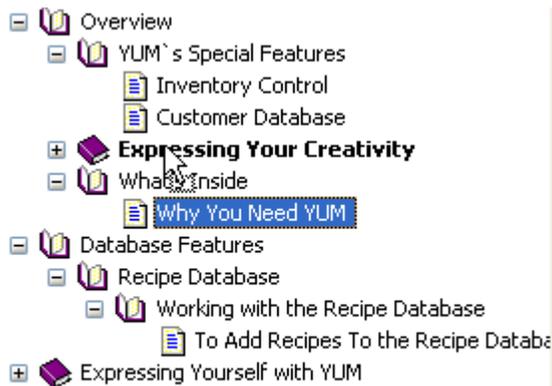
Alternatively, you can right-click the topic you want to convert into a book and select **Convert Topic to Book** from the shortcut menu.

Adding Topics to the Help Contents

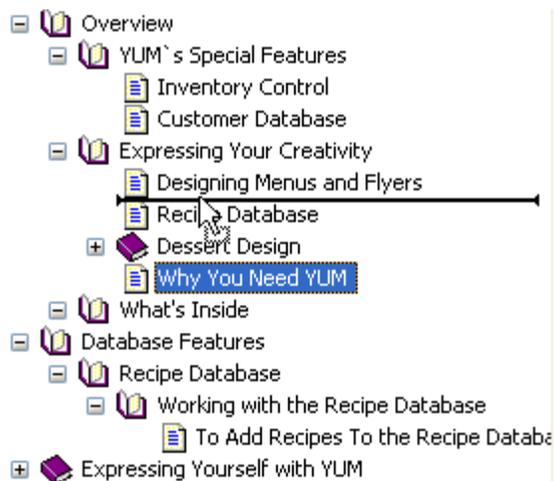
Upon compilation, all of the available topics that Doc-To-Help derives from the source documents are placed into the right pane of the Contents window. This is an inclusive list, so it includes topics already associated with books or main topics. The Contents window interface allows you to add these topics to different books in the left pane with ease.

Adding topics to the Help contents using the drag-and-drop method

1. In the Contents window, click the topic record selector in the right pane that you would like to associate with a topic in the left pane.
2. Drag the topic to the left pane.
3. If you drag the topic over an existing book, the book becomes bold indicating that the topic will be associated with that book if you drop it there.



4. If you drag it between books or topics, a bold line appears indicating the location where the new topic will be placed, if dropped.



5. Drop the topic in the left pane, or over the book with which you would like it associated.

Note: Doc-To-Help supports multiple topic selection by using the CTRL key. Simply press and hold the CTRL key for nonconsecutive selections, and choose the record selector buttons to the left of the topic titles.

Adding topics to the Help contents using the contents toolbar

1. Select a book in the left pane.
2. Select the topic in the right pane to be associated with that book.
3. Click the **Insert Topics** command on the toolbar or select **Insert** from the shortcut menu in the right pane.

The new topic is now associated under the book that was selected in the left pane. Note that if you selected a topic in the left pane, the new topic is inserted above that topic.

Renaming Books and Topics

1. Click the book or topic to be renamed in the left pane of the Contents window.
2. Right-click in the left pane and bring up the shortcut menu.
3. Select **Rename** from the shortcut menu.

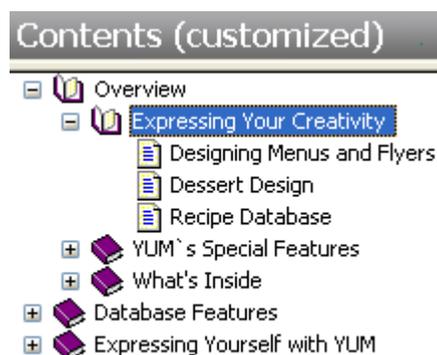
A blinking cursor appears and allow you to customize the name of the new book or topic.

Customizing the Table of Contents for Each Help Target

Doc-To-Help allows you to customize the table of contents (TOC) separately for each of the help targets so that different targets have different TOCs. By default, when you customize the TOC using the Contents tab, this single, customized TOC is applied to all help targets. If you want to customize a TOC for each help target independently of the others, though, you can use the HelpTarget.[OwnTOC](#) property. For example, you may want to change the order of topics in a certain help target without affecting the other targets. Set the [OwnTOC](#) property to **True** for the target, and modify the TOC. The modified TOC is applied to this target only, and it does not appear in the other help targets. If the [OwnTOC](#) property is set to **False**, the modified TOC is applied to all help targets.

To customize the TOC independently for a help target:

1. On the **Project** toolbar, select one of the available items from the Help Target drop-down.
2. On the Icon bar, click **Project**.
3. Select the **Help Targets** item from the left pane.
4. In the property pane, set the [OwnTOC](#) property to **True**.
5. On the Icon bar, click **Contents**.
6. Edit the contents as desired and select **File | Save Contents**. Doc-To-Help labels the contents as customized.



7. Click the **Make Target** button to build the selected help target.
8. Click the **View Target** button to view the help file. The customized TOC appears in the selected help target only.

Note: When using the **Manual** target, topics belonging to a multiple topic document always appear in the TOC in the same order as they appear in the source document. If the order is changed, including when topics from other documents are interspersed with multiple topic document topics, Doc-To-Help lists an error in the build log and ignores the customization. You can change the order of multiple topic documents in the Manual TOC and change the hierarchical level, for example, you can make the main topic of a multi-topic document a child of a topic from another document, but all topics belonging to a multi-topic document must always be kept together and in the same order as in the document.

This applies only to multiple topic documents when using the **Manual** target; topic order in single topic documents and other targets can be modified.

Saving the Help Contents

Doc-To-Help allows for the saving of the current contents without recompilation. The changes to the contents are stored within the active project file itself. The **Save Contents** menu item is available and can be used to save only while working in the Contents window.

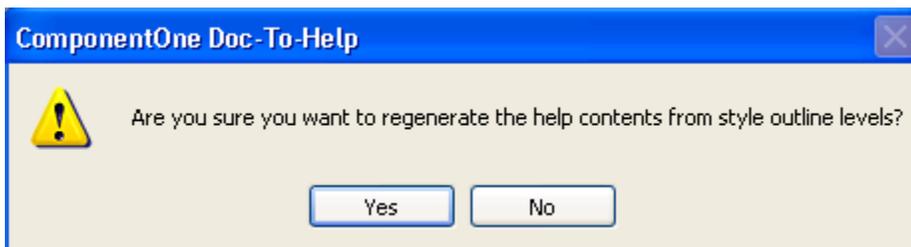
1. Open the **File** menu from the menu bar.
2. Select the menu option to **Save Contents**.

In addition, while working in the Contents window, if you attempt to close Doc-To-Help without saving, you are asked if you would like to save the Contents before exiting. This ensures that any changes to your contents are deliberate and properly retained.

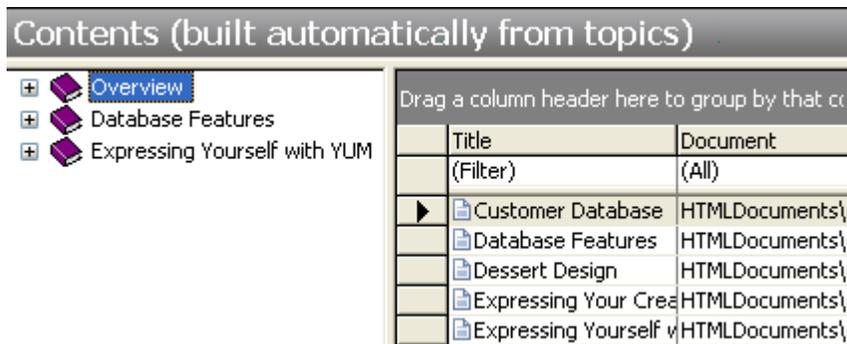
Rebuilding the Automatic Contents

If you have customized the table of contents and would like to revert back to the original contents order based on the topics, you can have Doc-To-Help rebuild the contents.

1. Open the **File** menu from the menu bar.
2. Select the menu option to **Rebuild Automatic Contents**. Doc-To-Help prompts you to make sure you want to regenerate the contents.



3. Click **Yes**. Your customization is discarded, and the table of contents is rebuilt automatically from the topics. The *(customized)* label becomes *(built automatically from topics)*.



Context Sensitive Help

Context-sensitive Help is Help that is sensitive to the context from which it is called—that is, it “knows” what you’re doing when you ask for Help and offers assistance on performing that specific task.

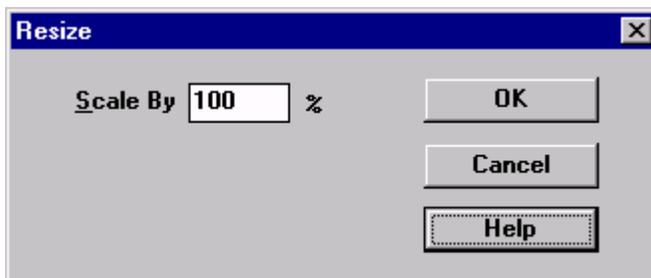
Before context-sensitive Help, integrating Help into an application was easy. The Help author wrote Help topics about an application, and when a user needed to access Help, she went to the Help menu, brought up the contents screen and navigated through the Help file until she found what she wanted.

For very simple applications, this model of Help authoring works well. Many users and programmers, however, prefer and demand a higher level of online Help. Help files, like the programs they document, have grown larger over the years. It is no longer convenient for a user to navigate through an entire Help file to find the section she needs. In a larger application, she may well never find it.

Enter context-sensitive Help. By reacting to the state of the application from which it is called, context-sensitive Help eliminates most of the navigation through the Help, searching for the relevant topics.

The Help Button

At its most basic level, context-sensitive Help can be a Help button on a dialog box.



To view the Help topic for the entire dialog box, the user can simply click the Help button. There’s no longer any need to search through an entire Help file. The jump to the correct topic is automatic and painless.

This method works well in most situations, but it has one drawback. Suppose the dialog box has a large number of text boxes, drop-down list boxes, command buttons, etc. The Help for that screen can rapidly exceed the space provided for by one Help screen (and Help topics that fill more than one Help screen are harder to read).

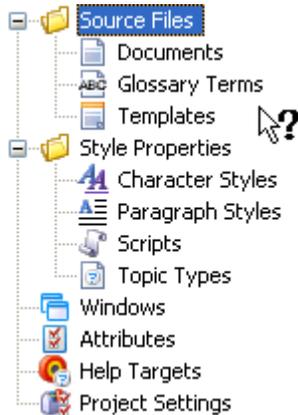
What’s This Help

A more finely tuned approach to context-sensitive Help allows the user to access pop-up Help topics for individual text boxes, drop-down list boxes, command buttons, etc. This sort of field-level Help is known as “What’s This?” help.

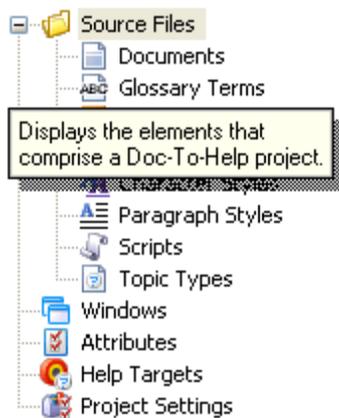
For most applications, there are three ways to access “What’s This” help in Windows:

1. The user can press **F1** to see pop-up Help for the currently selected item in the window.

2. The user can right-click the window item and click the "What's This" menu item in the pop-up menu that opens.
3. The user can click the "What's This" button in the title bar. The cursor becomes a special context-sensitive Help cursor: an arrow with a question mark attached to it.



When you click an item in the dialog box, Windows displays a pop-up Help topic for that item.



Creating Context-Sensitive Help

Context-sensitive Help involves associating topics in your Help project with specific items in a software application. The two most popular methods of making this association are to either have the Help author supply the application developer with topic ID numbers, which he uses as part of his application, or to have the developer supply the Help author with context ID numbers. In this second method, the Help author matches the context ID numbers with the topics in the Doc-To-Help project editor. For more information, see [Adding Context IDs With the Topic Properties Toolbar](#) (page 300).

Generating Context ID Numbers

To expedite the first method where the Help author needs to supply the applications developer with topic ID numbers, Doc-To-Help assigns a unique number to each Help topic. In this way, the developer can use these topic ID numbers as the handle through which an application calls the topic in the Help file. Please note that for this method to be functional, the [MapNumbers](#) property of the current Help target must be set to **Automatic** or **Automatic+Custom** and the [AutoContextID](#) of the related paragraph headings must be set to **True**.

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. Select the name of the desired Help target.

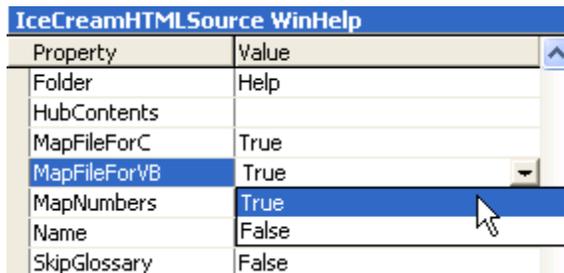
- In the property pane, set the [MapNumbers](#) property value to **Automatic**.



Property	Value
Folder	Help
HubContents	
MapFileForC	True
MapFileForVB	True
MapNumbers	Automatic
Name	Automatic
SkipGlossary	Automatic+Custom
StyleSheet	Custom

To facilitate the transfer of topic ID numbers to developers, you can generate Visual Basic or C/C++ header files in Doc-To-Help by toggling the [MapFileForVB](#) and [MapFileForC](#) properties in the Help target's property pane to **True**.

- On the Icon bar, click **Project**.
- Select the **Help Targets** item from the left pane.
- Select the name of the desired Help target.
- In the property pane, set the [MapFileForC](#) and the [MapFileForVB](#) property values to **True**.

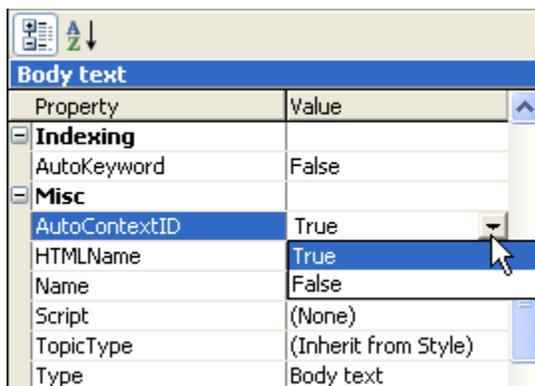


Property	Value
Folder	Help
HubContents	
MapFileForC	True
MapFileForVB	True
MapNumbers	True
Name	False
SkipGlossary	False

Note: Not all Help targets have the option to generate both VB and C header files.

To generate a context ID number:

- On the Icon bar, click **Project**.
- Select the **Paragraph Styles** item from the left pane.
- In the right pane, select the heading for which you want to generate context IDs.
- In the property pane, set the [AutoContextID](#) property value to **True**.

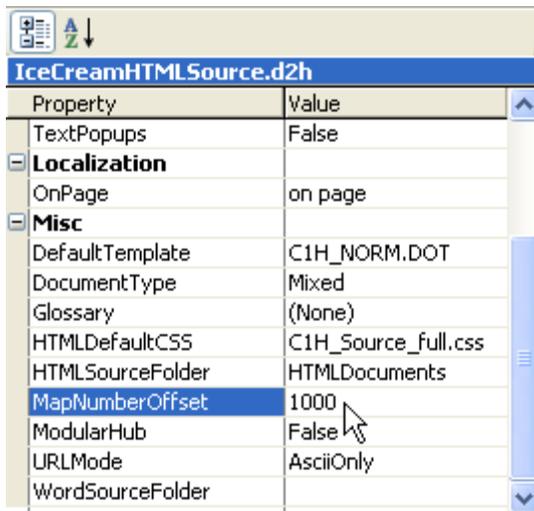


Property	Value
Indexing	
AutoKeyword	False
Misc	
AutoContextID	True
HTMLName	True
Name	False
Script	(None)
TopicType	(Inherit from Style)
Type	Body text

Customizing your Automatically Generated Context ID Numbers

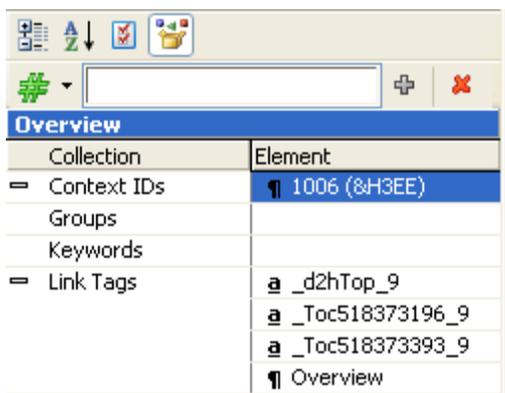
In some cases, like modular Help applications, you may want to customize your Context IDs to start at a specific number. You can do this using the [MapNumberOffset](#) property. When set, Doc-To-Help adds the value associated with this property with the automatically generated Context ID to create a customized number series.

1. On the Icon bar, click **Project**.
2. Select **Project Settings** from the left pane.
3. From the property pane, set the [MapNumberOffset](#) property to the value you want to use for your automatic Context IDs.



4. Rebuild your Help Project.

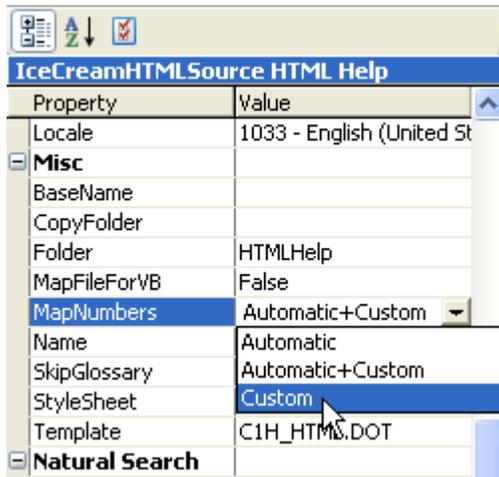
Note that your automatically generated context IDs have the [MapNumberOffset](#) value added to the context ID number.



Adding Context IDs With the Topic Properties Toolbar

After a Doc-To-Help project has been compiled, you can use the **Topic Properties** toolbar to insert context IDs. To associate a developer-supplied Help context ID with a topic:

1. On the Icon bar, click **Project**.
2. Select the **Help Targets** item from the left pane.
3. From the right pane, choose the Help target to be edited.
4. From the property pane, set the [MapNumbers](#) property value to **Custom**.



- Click the **Topics** icon on the **Icon** toolbar, select the **All Topics** item and select the desired topic in the right pane.
- Click the **Topic Collections** button  on the **Topics Property** toolbar.
- Select **Context ID** from the **Choose Collection** drop-down.



- Type the context ID, provided by the developer, into the **Element Textbox** and press **Enter**.

Note that the context ID has been associated with the highlighted topic and is displayed to the right of the **Context IDs** collection.

Note: You can assign multiple context IDs to a single topic, but you cannot assign a single context ID to multiple topics.

Context-Sensitive Help in HTML Help and WinHelp

To enable context-sensitive help in HTML Help and WinHelp, you must assign context IDs to the topics that you want to appear in the context-sensitive help. See [Using the Context String Editor](#) (page 569) for information on specifying context IDs in the Doc-To-Help project editor. For additional information on using HTML Help and WinHelp content-sensitive help in your applications, see the Microsoft HTML Help and WinHelp documentation.

Context-Sensitive Help in JavaHelp

The JavaHelp context-sensitive help model does not require context IDs, so you can show JavaHelp topics in your application without specifying context IDs. Instead, JavaHelp context-sensitive help uses topic map IDs. Topic map IDs are strings specified for every topic in the .jhm file (JavaHelp map file) created by Doc-To-Help in the JavaHelp target directory. Doc-To-Help uses the [AsciiName](#) topic property value to generate map IDs. The map IDs are automatically generated when you build a JavaHelp target. If you need specific map IDs for JavaHelp context-sensitive help in your project, assign them to the [AsciiName](#) topic property.

To assign topic map IDs in your Doc-To-Help project:

- In the **Icon Bar**, click **Topics**.

2. In the left pane, select **All Topics**.
3. Select a topic in the right pane.
4. In the property pane, enter the map ID for the [AsciiName](#) property.

For additional information on JavaHelp context-sensitive help, see the JavaHelp documentation.

Context-Sensitive Help in Microsoft Help 2.0

The Microsoft Help 2.0 help format is used only in help systems integrated with Visual Studio .NET. Visual Studio .NET uses context strings to identify topics shown in its context-sensitive help. To specify context strings for topics, use the [ContextString](#) property for each topic in the Doc-To-Help project that you want to appear in the context-sensitive help. See [Using the Context String Editor](#) (page 569) for more information.

Note: Documenter for .NET automatically generates [ContextString](#) values for the reference topics it creates.

Context-Sensitive Help in NetHelp

NetHelp, the ComponentOne browser-independent help format, supports a versatile context-sensitive help facility that can be used virtually anywhere: on web pages, in web applications and in client applications on any platform written in any programming language. You can show NetHelp topics in a frame inside your application window or in a separate browser window. You also have control over what parts of the help system you want to expose to the user. You can show topic text without navigational frames, or you may want to include full NetHelp navigation with the Context, Index and Search tabs. NetHelp context-sensitive help uses the same context IDs assigned to topics in your project as do the HTML Help and WinHelp targets.

To call NetHelp from your application or web pages, use one of the D2H_ctxt.* source files provided with Doc-To-Help; they define the D2H_ShowHelp function needed to enable context-sensitive help.

Note: The D2H_ctxt.* files are installed in the C:\Program Files\ComponentOne\DocToHelp\Context-Sensitive Help folder by default. The location may be different if you installed Doc-To-Help somewhere else on your machine.

Unless you need to change the functionality, do not modify the D2H_ctxt.* file. Simply add it to your application or web pages as it is. It is necessary to call the D2H_ShowHelp function in all environments. This function is all you need to enable context-sensitive NetHelp. It is implemented in different programming languages for different environments:

Application	Language	File
Web applications and web pages	JavaScript	D2H_ctxt.js
Windows client applications	C#	D2H_ctxt.cs
	VB.NET	D2H_ctxt.vb
	Visual Basic 6.0	D2H_ctxt.bas
Java client applications	Java	D2H_ctxt.java

Samples demonstrating how to use context-sensitive help with NetHelp in each of these environments can be found in the C:\Program Files\ComponentOne\DocToHelp\Context-Sensitive Help\Samples directory. Please note that the location of this folder may be different if you installed Doc-To-Help somewhere else on your machine.

For each sample, start by opening the files below, then run the program, if necessary, and follow the instructions listed on the page:

CSharp	CSharp.csproj
Java	start_sample.bat

VB6	VB6.vbp
VB.NET	VB.NET.vbproj
WebPages	ctxt_sample.htm

If your programming language is not listed above, you can still use NetHelp context-sensitive help if you translate the code of the D2H_ShowHelp function to the language you are using.

The function is described below in more detail:

Function D2H_ShowHelp in JavaScript (web applications and web pages)

Syntax

```
function D2H_ShowHelp(contextID, mainURL, wndName, uCommand)
```

Parameters:

ContextID (integer number):	Context ID of the topic to display. See Context Sensitive Help (page 297) for more information on assigning context IDs to topics.
mainURL (string):	URL of the main web page of the help. For example, the URL of a locally deployed NetHelp target can be a file path, although it still needs to be encoded as a URL with the prefix "file://", such as file:///C:/Program Files/ComponentOne/DocToHelp/Samples/StyleGuide/NetHelp/default.htm ; or, for a server-deployed NetHelp target it can be http://www.mycompany.com/myhelp/default.htm . This parameter is used to identify the help system containing the topic. It is necessary because you may use multiple help systems in the same application or even on the same web page.
wndName (string):	A frame or a browser window to display the topic in. Using this parameter you can show the topic in any frame or in a separate browser window. This parameter has the same possible values as the TARGET attribute of the A (anchor) tag in HTML. If you want to display help in a certain frame or in a separate browser window, set this parameter to that frame or window name. You can also use the standard names supported by the TARGET attribute of the A tag: _self, _blank, _parent, etc.
uCommand (integer number, one of the two possible values: 1 or 2):	This parameter specifies one of the two options (Note: you can use the symbolic names or the numbers 1,2): CTXT_DISPLAY_FULLHELP (=1): Display the complete help system with Contents, Index and Search. The topic will be current in the Table of Contents, and the user will be able to navigate through the links in the topic and through the Context, Index and Search. CTXT_DISPLAY_TOPICONLY (=2): Display the topic without the Contents, Index and Search. The user will still be able to navigate to other topics if this topic contains links to other topics, but the Contents, Index and Search will be hidden.

To enable the function:

Include the D2H_ctxt.js file with your web pages and add the following tag to the web pages where you want to use context-sensitive help:

```
<script language="JavaScript" src="D2H_ctxt.js">
</script>
```

Example:

```
<A HREF=' JavaScript: D2H_ShowHelp(91, helpURL, "helpFrame",
CTXT_DISPLAY_TOPICONLY)' >Topic Only</A>
```

Function D2H_ShowHelp in Windows client applications

In Windows client applications, you can use NetHelp context-sensitive help in two different ways:

- Use the Microsoft WebBrowser control to show help inside one of your application windows. In this case you need to add a reference to the WebBrowser control to your application.
- Show help in a separate browser window. In this case you don't need to use the WebBrowser control.

Syntax:

C#

```
D2H_ctxt.D2H_ShowHelp(int contextID, string mainURL, object wnd, Display
uCommand)
```

Visual Basic .NET

```
D2H_ctxt.D2H_ShowHelp(ByVal contextID As Integer, ByVal mainURL As String,
ByVal wnd As Object, ByVal uCommand As Display) As Boolean
```

Visual Basic 6.0

```
D2H_ShowHelp(ByVal ContextID As Integer, ByVal mainURL As String, ByVal wnd
As Object, ByVal uCommand As Display) As Boolean
```

Parameters:

ContextID (integer number):	Context ID of the topic to display. See Context Sensitive Help (page 297) for more information on assigning context IDs to topics.
mainURL (string):	URL of the main web page of the help. For example, the URL of a locally deployed NetHelp target can be a file path, although it still needs to be encoded as URL with the prefix "file:/// ", such as file:///C:/Program Files/ComponentOne/DocToHelp/Samples/StyleGuide/NetHelp/default.htm ; or, for a server-deployed NetHelp target it can be http://www.mycompany.com/myhelp/default.htm . This parameter is used to identify the help system containing the topic. It is necessary because you may use multiple help systems in the same application.
wnd (WebBrowser object):	If this parameter is null (Nothing in Visual Basic .NET and Visual Basic 6.0), the help is shown in a separate browser window, as if you were opening an HTML file with a double-click. If this parameter is set to a WebBrowser component, the help is shown in that component. Using the WebBrowser component, you can show help inside your application windows.
uCommand (integer number, one of the two possible values: 1 or 2):	This parameter specifies one of the two options (Note: you can use the symbolic names or the numbers 1,2): CTXT_DISPLAY_FULLHELP (=1): Display the complete help system with Contents, Index and Search. The topic will be current in the Table of Contents, and the user will be able to navigate through the links in the topic and through the Context, Index and Search. CTXT_DISPLAY_TOPICONLY (=2): Display the topic without the Contents, Index and Search. The user will still be able to navigate to other topics if this topic contains links to other topics, but the Contents, Index and Search will be hidden.

To enable the function:

- Add one of the D2H_ctxt.* files to your project; use D2H_ctxt.cs for C#, D2H_ctxt.vb for Visual Basic .NET, or D2H_ctxt.bas for Visual Basic 6.0.
- To show context-sensitive help in one of your application windows, add the WebBrowser control reference to your application. If you choose to show help in a separate browser window, you don't need a reference to the WebBrowser control.

Example:*C# and Visual Basic .NET*

```
C1. D2H . D2H_ctxt. D2H_ShowHelp(91, helpURL, WebBrowser1,
CTXT_DISPLAY_TOPICONLY)
```

Visual Basic 6.0

```
D2H_ctxt. D2H_ShowHelp(91, helpURL, WebBrowser1, CTXT_DISPLAY_TOPICONLY)
```

Function D2H_ShowHelp in Java client applications

In Java client applications (usually created with Swing components), the D2H_ShowHelp function uses JDIC (JDesktop Integration Components), in particular, the WebBrowser component. The WebBrowser class allows Java applications to use full browser HTML-rendering capabilities inside a Java component or in a separate browser window, just like the Microsoft WebBrowser control in Windows applications.

Syntax:

```
D2H_ctxt. D2H_ShowHelp(int contextID, String mainURL, WebBrowser wnd, int
uCommand)
```

Note: D2H_ShowHelp is a method of the D2H_ctxt class

Parameters:

ContextID (integer number):	Context ID of the topic to display. See Context Sensitive Help (page 297) for more information on assigning context IDs to topics.
mainURL (string):	URL of the main web page of the help. For example, the URL of a locally deployed NetHelp target can be a file path, although it still needs to be encoded as URL with the prefix "file:///", such as file:///C:/Program Files/ComponentOne/DocToHelp/Samples/StyleGuide/NetHelp/default.htm ; or, for a server-deployed NetHelp it can be http://www.mycompany.com/myhelp/default.htm . This parameter is used to identify the help system containing the topic. It is necessary because you may use multiple help systems in the same application.
wnd (WebBrowser object):	If this parameter is null, the help is shown in a separate browser window (as if you were opening an HTML file with a double-click). If this parameter is set to a WebBrowser component, the help is shown in that component. Using the WebBrowser component, you can show help inside your application's windows.
uCommand (integer number, one of the two possible values: 1 or 2):	This parameter specifies one of the two options (Note: you can use the symbolic names or the numbers 1,2): CTXT_DISPLAY_FULLHELP (=1): Display the complete help system with the Contents, Index and Search. The topic will be current in the Table of Contents, and the user will be able to navigate through links in the topic and through the Context, Index and Search. CTXT_DISPLAY_TOPICONLY (=2): Display the topic without the Contents, Index and Search. The user will still be able to navigate to other topics if this topic contains links to other topics, but the Contents, Index and Search will be hidden.

To enable the function:

To use `D2H_ShowHelp` in Java client applications, you need to install JDIC. JDIC is available at <http://jdic.dev.java.net>. You also need to add `D2H_ctxt.java` to your project's source files. Building your application, add `jdic.jar` to your classpath. You can use the batch files in the sample directory located at `C:\Program Files\ComponentOne\DocToHelp\Context-Sensitive Help\Samples\Java`. Note that the location may be different if you installed Doc-To-Help somewhere else on your machine. In those batch files, change the values of the `JDK_HOME` and `JDIC_HOME` variables to point to your JDK and JDIC locations.

Without JDIC:

If you do not want to use JDIC, you can still show NetHelp context-sensitive help in Java client applications. Modify `D2H_ctxt.java`, excluding the parts using the WebBrowser component. Replace `Desktop.browse(u)` with a method you choose to open a URL in a browser. JDIC is required for showing help inside your application windows.

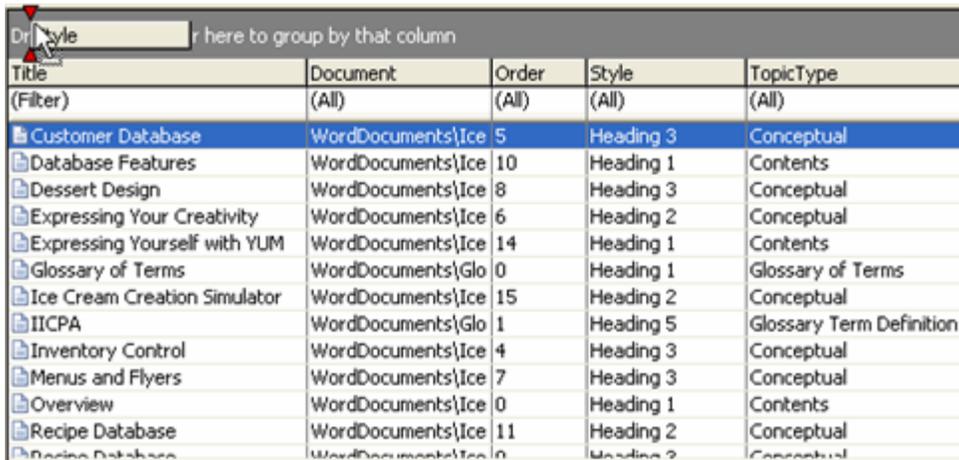
Example:

```
D2H_ctxt.D2H_ShowHelp(91, helpURL, browser, D2H_ctxt.CTXT_DISPLAY_TOPICONLY);
```

Sorting Topics For Easy Viewing With Outlook Style Grouping

Doc-To-Help provides the Help author with enhanced topic display and sorting capabilities in the topic list pane on the right by using a convenient outlook style grouping feature. By simply dragging any of the column headings (Title, Document or Style) into the grouping area located at the top of the topic list pane, Doc-To-Help sorts automatically by that heading. To sort by headings:

1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane.
3. Select the desired topic from the right pane.
4. In the right pane, select the "Style" column from the header bar, and while holding down the mouse button, drag its header to the column grouping area as shown below.



Drag here to group by that column				
Title	Document	Order	Style	TopicType
(Filter)	(All)	(All)	(All)	(All)
Customer Database	WordDocuments\Ice	5	Heading 3	Conceptual
Database Features	WordDocuments\Ice	10	Heading 1	Contents
Dessert Design	WordDocuments\Ice	8	Heading 3	Conceptual
Expressing Your Creativity	WordDocuments\Ice	6	Heading 2	Conceptual
Expressing Yourself with YUM	WordDocuments\Ice	14	Heading 1	Contents
Glossary of Terms	WordDocuments\Glo	0	Heading 1	Glossary of Terms
Ice Cream Creation Simulator	WordDocuments\Ice	15	Heading 2	Conceptual
IICPA	WordDocuments\Glo	1	Heading 5	Glossary Term Definition
Inventory Control	WordDocuments\Ice	4	Heading 3	Conceptual
Menus and Flyers	WordDocuments\Ice	7	Heading 3	Conceptual
Overview	WordDocuments\Ice	0	Heading 1	Contents
Recipe Database	WordDocuments\Ice	11	Heading 2	Conceptual
Recipe Database	WordDocuments\Ice	0	Heading 2	Conceptual

5. Release the mouse button and note the change in the Doc-To-Help topic list pane. The rows are now sorted by the "Style" heading.

Style				
Style	Title	Document	Order	TopicType
(All)	(Filter)	(All)	(All)	(All)
Heading 1	Database Features	WordDocuments\Ice	10	Contents
	Expressing Yourself with YUM	WordDocuments\Ice	14	Contents
	Glossary of Terms	WordDocuments\Glo	0	Glossary of Terms
Heading 2	Overview	WordDocuments\Ice	0	Contents
	Expressing Your Creativity	WordDocuments\Ice	6	Conceptual
	Ice Cream Creation Simulator	WordDocuments\Ice	15	Conceptual
	Recipe Database	WordDocuments\Ice	11	Conceptual
	What's Inside	WordDocuments\Ice	1	Conceptual
Heading 3	YUM's Special Features	WordDocuments\Ice	3	Conceptual
	Customer Database	WordDocuments\Ice	5	Conceptual
	Dessert Design	WordDocuments\Ice	8	Conceptual
	Inventory Control	WordDocuments\Ice	4	Conceptual
	Menus and Flyers	WordDocuments\Ice	7	Conceptual
	Recipe Database	WordDocuments\Ice	9	Conceptual
	To Start the Ice Cream Creation	WordDocuments\Ice	16	Conceptual

6. Select the "Document" column and drag its header to the column group area. Note that the topics are sorted by Style, then Document as shown below.

Style				
Style	Document	Title	Order	TopicType
(All)	(All)	(Filter)	(All)	(All)
Heading 1	WordDocuments\Ice	Expressing Yourself with YUM	14	Contents
		Overview	0	Contents
Heading 2		Expressing Your Creativity	6	Conceptual
		Ice Cream Creation Simulator	15	Conceptual
Heading 3		Recipe Database	11	Conceptual
		What's Inside	1	Conceptual
		YUM's Special Features	3	Conceptual
		Customer Database	5	Conceptual
		Dessert Design	8	Conceptual
Heading 3		Inventory Control	4	Conceptual
	Menus and Flyers	7	Conceptual	
	Recipe Database	9	Conceptual	
	To Start the Ice Cream Creation	16	Conceptual	
		Why You Need YUM	2	Conceptual
		Working with the Recipe Databa	12	Conceptual

7. By simply dragging the headers back down into the header bar you can return to the default topic view.

Sorting to View Order by Document

In some cases, you may need to view the order of the topics within each document. Sorting by the document, then order, columns provides you with this view.

1. To do this, select the document column from the header bar, and while holding down the mouse button, drag its header to the column grouping area.
2. Next, select the order column from the header bar, and while holding down the mouse button, drag its header to the column grouping area.

Note that the topics are sorted by document, then order as shown below.

Document	Order	Title	Style	TopicType
(All)	(All)	(Filter)	(All)	(All)
WordDocuments\Glo	0	Glossary of Terms	Heading 1	Glossary of Terms
	1	IICPA	Heading 5	Glossary Term Definition
WordDocuments\Ice	0	Overview	Heading 1	Contents
	1	What's Inside	Heading 2	Conceptual
	2	Why You Need YUM	Heading 3	Conceptual
	3	YUM's Special Features	Heading 2	Conceptual
	4	Inventory Control	Heading 3	Conceptual
	5	Customer Database	Heading 3	Conceptual
	6	Expressing Your Creativity	Heading 2	Conceptual
	7	Menus and Flyers	Heading 3	Conceptual
	8	Dessert Design	Heading 3	Conceptual
	9	Recipe Database	Heading 3	Conceptual
	10	Database Features	Heading 1	Contents
	11	Recipe Database	Heading 2	Conceptual
12	Working with the Recipe Databa	Heading 3	Conceptual	

Filtering Topics in Doc-To-Help

Doc-To-Help provides the author with topic filtering capabilities by using the filter boxes located beneath the headers in the topic list in the right pane. By simply choosing one or more of the drop-down filtering options, or by entering text in the topic textbox, you can filter the topics in any fashion you like. To filter the topics:

1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane.
3. To filter by Topic, simply type the text you want to use as your filter into the textbox under the Title header and press **Enter**.

Drag a column header here to group by that column				
Title	Document	Order	Style	TopicType
Database	(All)	(All)	(All)	(All)
Customer Database	WordDocuments\Ice	5	Heading 3	Conceptual
Database Features	WordDocuments\Ice	10	Heading 1	Contents
Recipe Database	WordDocuments\Ice	11	Heading 2	Conceptual
Recipe Database	WordDocuments\Ice	9	Heading 3	Conceptual
To Add Recipes To the Recipe D	WordDocuments\Ice	13	Heading 4	Conceptual
Working with the Recipe Databa	WordDocuments\Ice	12	Heading 3	Conceptual

4. To filter by Document or Style, click the arrow under the respective header and choose the item you want to use as your filter.

Drag a column header here to group by that column				
Title	Document	Order	Style	TopicType
Database	(All)	(All)	(All)	(All)
Customer Database	WordDocuments\Ice	5	(All)	Conceptual
Database Features	WordDocuments\Ice	10	Glossary Heading	Contents
Recipe Database	WordDocuments\Ice	11	Heading 1	Conceptual
Recipe Database	WordDocuments\Ice	9	Heading 2	Conceptual
To Add Recipes To the Rec	WordDocuments\Ice	13	Heading 3	Conceptual
Working with the Recipe D	WordDocuments\Ice	12	Heading 4	Conceptual
			Heading 5	Conceptual
			MidTopic	
			WhatsThis	

Note that by combining the Title, Document and Style filtering options, you can sort by almost any combination you like as shown above.

Links and Hot Spots

Links and hot spots are created within your source document by formatting them with character styles using [Doc-To-Help Markup Language \(D2HML\)](#) (page 233). Using Doc-To-Help's wide variety of predefined D2HML styles is the recommended way to create links and hot spots, but, if necessary, you also have the more advanced option of creating your own character styles for D2HML. There are also alternative ways to create links; these will be discussed later in this section.

In order for a hot spot in your source document to work correctly in the help target, it must have a corresponding character style with an active behavior in the Doc-to-Help project. The [Behavior](#) property of the style tells the hot spot what to do. For example, C1H Jump is a predefined character style in Doc-to-Help. The [Behavior](#) property of C1H Jump is **Topic Link**, so Doc-To-Help knows to treat the hot spot formatted with the C1H Jump style as a link to the topic.

A style is considered active when its [Behavior](#) property is set to a value other than None.

Creating a Jump to a Topic in Another Document

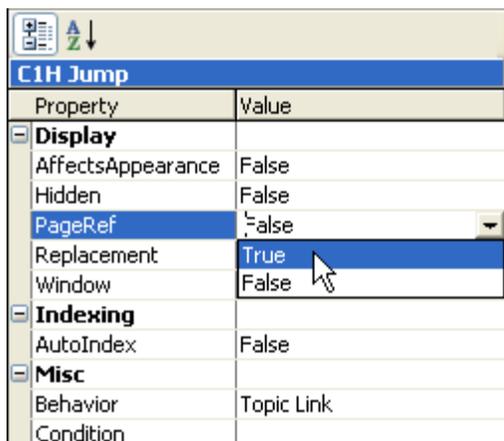
In online Help targets, a jump is a hyperlink that displays another topic in the current Help window. In printed manual targets, a jump can refer to another topic by its physical page number.

Use the built-in character style C1HJump to link to a topic in any document within your project.

To create a jump to a topic in another document, select the link text and apply the **C1HJump** character style. Use the Topic Link D2HML dialog box to specify a topic to link to, or add a secondary part to the hot spot, setting its *tag* property to the link tag of the topic to which you are jumping. If one of the link tags of the destination topic exactly matches the text of the hot spot or the *tag* property, the jump is created. See [Creating a Hot Spot](#) (page 249) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information on applying D2HML styles and using D2HML.

To enable automatic page number references for jumps in printed manual targets:

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane and select a character style whose [Behavior](#) property is set to **Topic Link** and [LinkType](#) is set to **Jump**.
3. In the property pane, set the [PageRef](#) property value to **True**.



Creating Popup Topics

In online Help targets, a pop-up is a hyperlink that displays another topic in a separate pop-up window. In printed manual targets, a pop-up can refer to another topic by its physical page number.

To create a pop-up topic, select the link text and apply the **C1HPopup** character style. Use the Popup Link D2HML dialog box to specify a topic to link to, or add a secondary part to the hot spot, setting its *tag* property to the link tag of the topic to which you are jumping. If one of the link tags of the destination topic exactly matches the text of the hot spot or the *tag* property, the jump is created. See [Creating a Hot Spot](#) (page 249) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information on applying D2HML styles and using D2HML.

Note: When adding a pop-up in WinHelp, set the [NonScrolling](#) property of the “target” paragraph style to **False**.

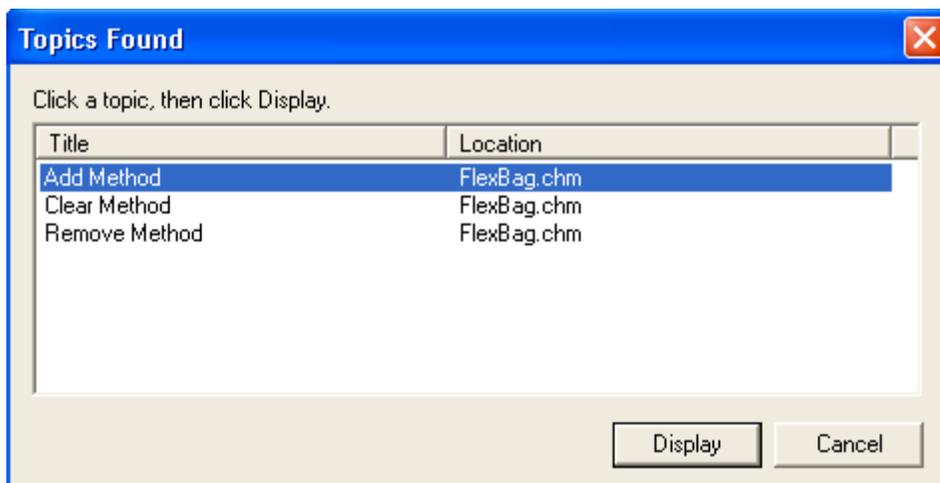
To enable automatic page number references for pop-ups in printed manual targets:

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane and select a character style whose [Behavior](#) property is set to **Topic Link** and [LinkType](#) is set to **Popup**.
3. In the property pane, set the [PageRef](#) property value to **True**.

Creating Keyword Links (KLinks)

In online Help targets, a keyword link displays a list of available topics that have been associated with an index keyword.

For example, if you were describing the FlexBag ActiveX Object and created the keyword link *FlexBag* in your document, a user would be able to click *FlexBag* in the Help file, and a dialog box would appear, allowing the user to choose any of the topics that have been associated with *FlexBag*, as shown below.



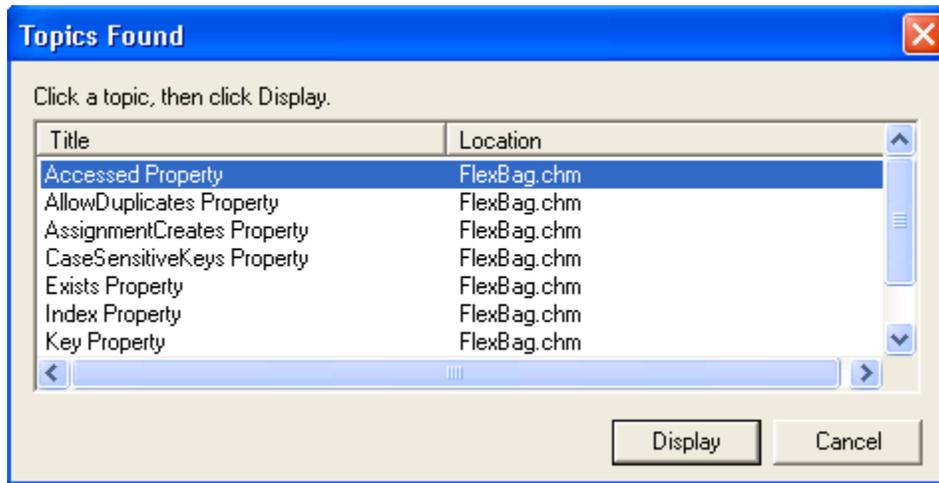
To create a keyword link, select the link text and apply the **C1HKeywordLink** character style. Use the Keyword Link D2HML dialog box to specify a topic to link to, or add a secondary part to the hot spot, setting its *keyword* property to the topic to which you want to link. See [Creating a Hot Spot](#) (page 249) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information on applying D2HML styles and using D2HML.

Note: Keyword links have no effect in printed manual targets.

Creating Group Links (ALinks)

In online Help targets, a group link (ALink) displays a list of available topics associated with a named group, which is private to the Help project and not visible to the reader.

For example, if you were describing the benefits of the FlexBag ActiveX Object in your document, you might write, “The FlexBag ActiveX Object has *additional properties* which make it possible to manipulate keys more easily.” If you created the group link *additional properties*, a user would be able to click *additional properties* in the Help file, and a dialog box would appear, allowing the user to choose any of the topics that have been associated with *additional properties*, as shown below.



Note: Since *additional properties* is a group, there is NOT a visible section in your document or Help file called *additional properties*.

To create a group link, select the link text and apply the **C1HGroupLink** character style. Use the Group Link D2HML dialog box to specify the group that is associated with the topics to which you want to link. If you are not using the dialog box, add a secondary part to the hot spot, setting its *group* property to the group associated with the topics. See [Creating a Hot Spot](#) (page 249) and [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information on applying D2HML styles and using D2HML.

Note: Group links have no effect in printed manual targets.

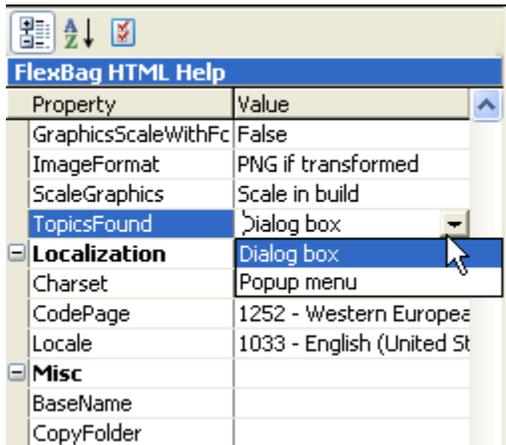
Modifying the Topics Found Display

When creating group links for HTML Help, Doc-To-Help provides you with the option of modifying how your “Topics Found” message is displayed. By setting the [TopicsFound](#) property, your topics found message can display in the standard dialog or in a pop-up.

To set the TopicsFound property:

1. Click the **Project** icon.
2. Select **Help Targets** from the left pane.
3. From the right pane, select the HTML Help target you want to modify.

- From the property pane, select the [TopicsFound](#) property and choose the display option from the drop-down menu.



Creating Mid-Topic Jumps Using D2HML

The **MidTopic** paragraph style defines a topic as an “in topic” jump location similar to an HTML anchor. By formatting text with the **MidTopic** paragraph style and creating a topic link hot spot, end users have jump functionality within a lengthy topic.

Inserting a Mid Topic Jump

- Open your source document, select or insert your mid topic heading and define it with the **MidTopic** paragraph style. The mid topic heading must be formatted with a “heading” paragraph on a separate line (not part of the body text).
- Save the document and open or return to the project editor.
- In the left pane, select the **Help Targets** item.
- Choose a Help target from the right pane, and build the Help target. At this point you have created your mid topic jump location.

Create a hotspot to this location using D2HML:

- Open the source document where you defined the mid topic location heading.
- Select your hot spot text.
- Click the **Jump Link** button  on the **D2HML Styles** toolbar. The **Topic Link** dialog box opens.



4. Click the **Browse** button. The **Select Topic** window appears.
5. Select the topic you defined as your mid topic jump location and click **OK** to close the **Select Topic** window.
6. Click **OK** again to close the dialog box. The hot spot is formatted with the **C1HJump** character style.
7. Close the source document.
8. In the left pane of the project editor, select the **Help Targets** item.
9. Choose a Help target from the right pane, and build the Help target. The mid-topic jump is created.

For detailed information on using D2HML, see [Doc-To-Help Markup Language \(D2HML\)](#) (page 233).

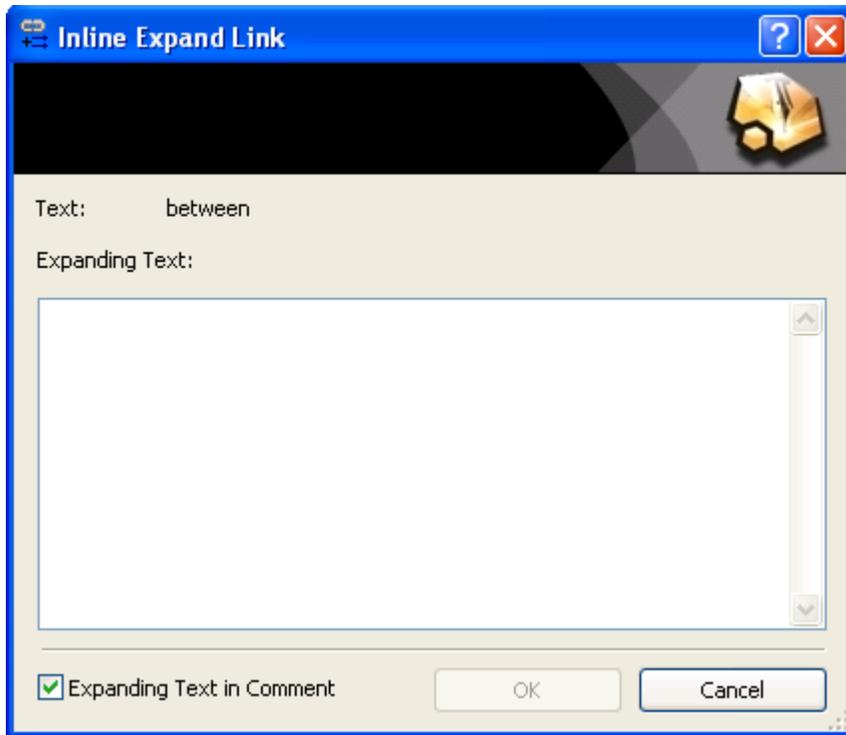
Expanding, Dropdown and Popup Text

In online Help targets, you can create Inline Expand, Inline Dropdown and Inline Popup hot spots, allowing you different options for displaying text. The Inline Expand hot spot, when clicked, exposes the specified expanding text. The Inline Dropdown hot spot displays drop-down text below the link, while the Inline Popup hot spot displays your text in a pop-up box.

Creating Expanding Text

Using the D2HML Dialog Box in Word

1. In your source document, select the link text that the user will click to view the expanding text.
2. Click the **Inline Expand** button on the **D2HML Styles** toolbar. The **Inline Expand Link** dialog box appears.



Note: Expanding text may require style formatting. Microsoft Word does not allow you to apply style formatting within expanding text because it is already formatted with the **C1H Expand Text** style. By default, Doc-To-Help applies expanding text as a comment rather than a D2HML style in your Word documents so that style formatting may be used within the comment. To format expanding text as a D2HML style, uncheck the **Expanding Text in Comment** checkbox.

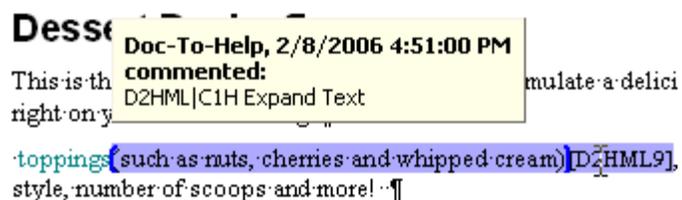
3. Enter the expanding text in the **Expanding Text** window and click **OK**. The link text is formatted with the **C1HInlineExpand** character style. The expanding text is formatted with a comment: **Page: 314 D2HML | C1H Expand Text**.

Using Expand Text Comments in Word

1. In your Word source document, select the link text that the user will click to view the drop-down text.
2. Select the **C1H Inline Dropdown** character style from the Word styles drop-down list.
3. Select the text, including graphics and formatted text, to be converted into expanding text.
4. Select **Insert | Comment** and add the following text, which will become your comment text:

D2HML | C1H Expand Text

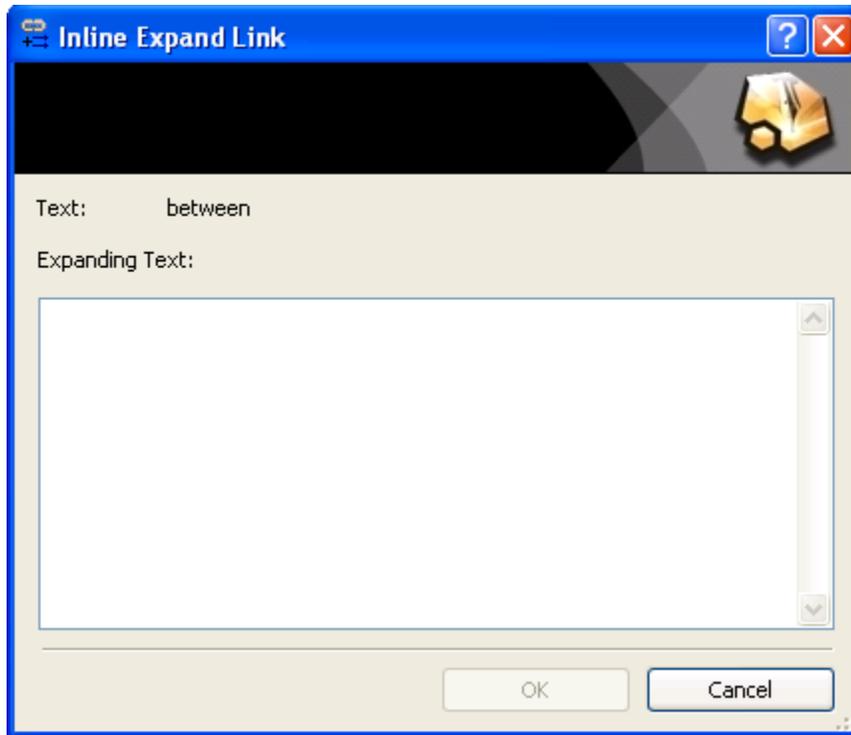
Your comment should look like the following image.



Using the D2HML Dialog Box in FrontPage and Dreamweaver

1. In your source document, select the link text that the user will click to view the drop-down text.

- Click the **Inline Expand** button on the **D2HML Styles** toolbar. The **Inline Expand Link** dialog box appears.



- Enter the drop-down text in the **Expanding Text** window and click **OK**.
- The link text is formatted with the **C1HInlineExpand** character style. The expanding text is formatted with the **C1HExpandText** character style.

Note: You can add additional items, such as graphics and formatted text, to the expanding text. Once you have created the initial expanding text and closed the D2HML dialog box, simply insert the graphics and/or formatted text in your document and format it with the corresponding **C1HExpandText** style.

Using D2HML Styles without D2HML Dialog Boxes

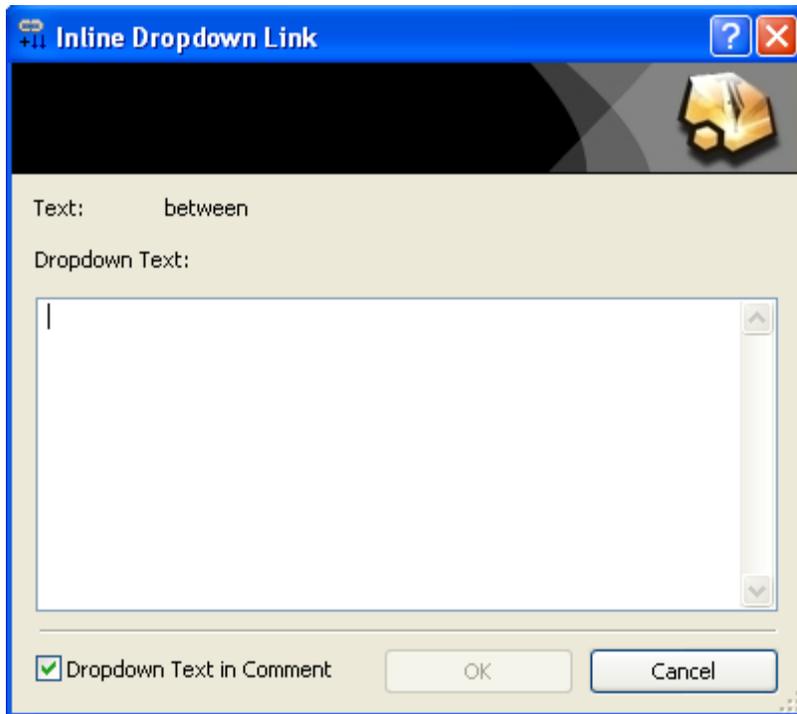
You can apply D2HML styles without using D2HML dialog boxes in your Word and HTML source documents.

- In your source document, select the link text and apply the **C1HInlineExpand** character style.
- Select the expanding text and apply the **C1HExpandText** character style. If using HTML source documents, you can include graphics and formatted text in your drop-down text. Just make sure they are formatted with the **C1HExpandText** style. If using Word documents, you can manually create an expand text comment and include the graphics and formatted text within the comment.

Creating Dropdown Text

Using the D2HML Dialog Box in Word

- In your source document, select the link text that the user will click to view the drop-down text.
- Click the **Inline Dropdown** button on the **D2HML Styles** toolbar. The **Inline Dropdown Link** dialog box appears.



3. Enter the drop-down text in the **Dropdown Text** window and click **OK**.
4. The link text is formatted with the **C1HInlineDropdown** character style. The drop-down text is formatted with a comment: **D2HML | C1H Dropdown Text**.

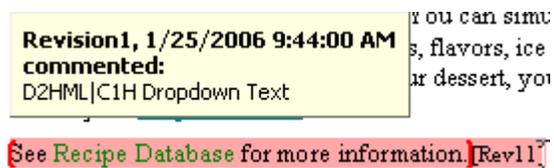
Note: Drop-down text may require style formatting. Microsoft Word does not allow you to apply style formatting within drop-down text because it is already formatted with the **C1H Dropdown Text** style. By default, Doc-To-Help applies drop-down text as a comment rather than a D2HML style in your Word documents so that style formatting may be used within the comment. To format drop-down text as a D2HML style, uncheck the **Dropdown Text in Comment** checkbox.

Using Dropdown Text Comments in Word

1. In your Word source document, select the link text that the user will click to view the drop-down text.
2. Select the **C1H Inline Dropdown** character style from the Word styles drop-down list.
3. Select the text, including graphics and formatted text, to be converted into drop-down text.
4. Select **Insert | Comment** and add the following text, which will become your comment text:

D2HML | C1H Dropdown Text

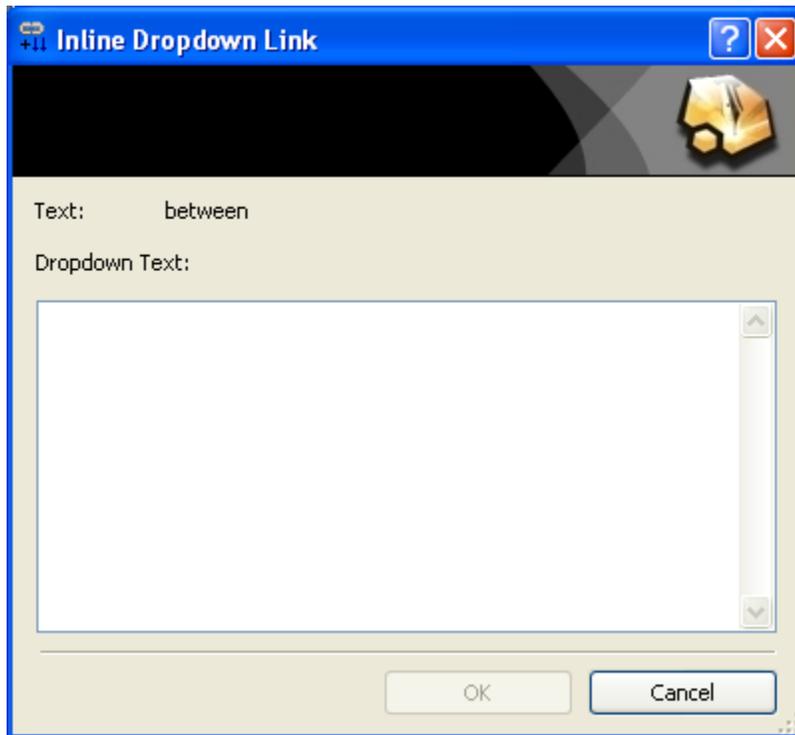
Your comment should look like the following image.



Using the D2HML Dialog Box in FrontPage and Dreamweaver

1. In your source document, select the link text that the user will click to view the drop-down text.

- Click the **Inline Dropdown** button on the **D2HML Styles** toolbar. The **Inline Dropdown Link** dialog box appears.



- Enter the drop-down text in the **Dropdown Text** window and click **OK**.
- The link text is formatted with the **C1HInlineDropdown** character style. The drop-down text is formatted with the **C1HDropdownText** character style.

Note: You can add additional items, such as graphics and formatted text, to the drop-down text. Once you have created the initial drop-down text and closed the D2HML dialog box, simply insert the graphics and/or formatted text in your document and format it with the corresponding **C1HDropdownText** style.

Using D2HML Styles without D2HML Dialog Boxes

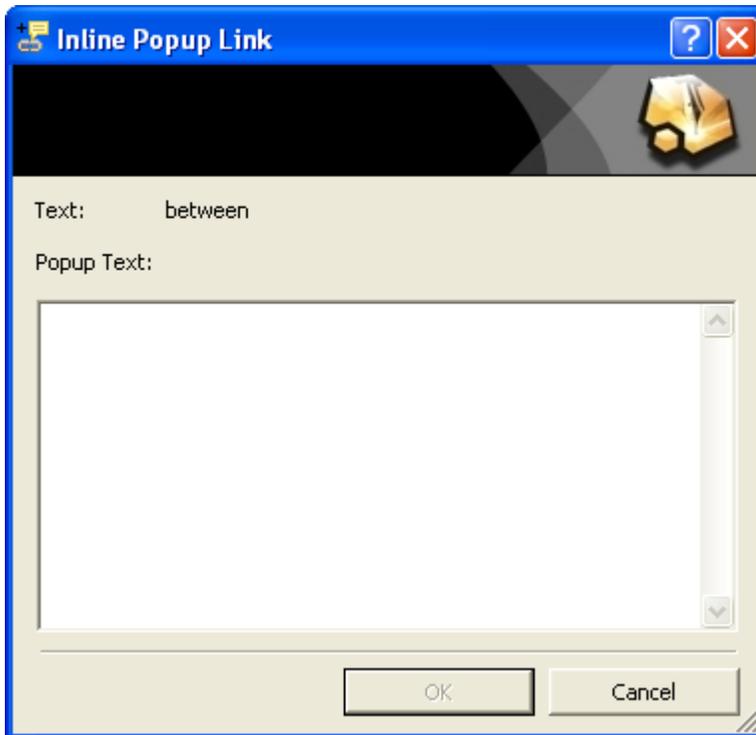
You can apply D2HML styles without using D2HML dialog boxes in your Word and HTML source documents.

- In your source document, select the link text and apply the **C1HInlineDropdown** character style.
- Select the drop-down text and apply the **C1HDropdownText** character style. If using HTML source documents, you can include graphics and formatted text in your drop-down text. Just make sure they are formatted with the **C1HDropdownText** style. If using Word documents, you can manually create a drop-down text comment and include the graphics and formatted text within the comment.

Creating Popup Text

Using the D2HML Dialog Box

- In your source document, select the link text that the user will click to view the pop-up text.
- Click the **Inline Popup** button on the **D2HML Styles** toolbar. The **Inline Popup Link** dialog box appears.



3. Enter the pop-up text in the **Popup Text** window and click **OK**.
4. Click **Yes** or **No** to make all hidden hot spots visible. The link text is formatted with the **C1HInlinePopup** character style. The pop-up text is formatted with the **C1HPopupText** character style.

Using D2HML Styles without D2HML Dialog Boxes

1. Select the link text and apply the **C1HInlinePopup** character style.
2. Select the pop-up text and apply the **C1HPopupText** character style.

Creating Active Character Styles

If you are not able to use one of the many predefined character styles that comes with Doc-To-Help, you can create your own. This is an advanced option and will probably not need to be used often.

1. Create the style in your Word or HTML source document.
2. Create a like-named character style in the Doc-To-Help project:
 - On the Icon bar, click **Project**.
 - Select the **Character Styles** item from the left pane.
 - Right-click anywhere in the character style list in the right pane and select **New** from the shortcut menu.
 - Type the name of the new character style and press **Enter**. The new character style is added to the list.
3. Set the [Behavior](#) property of the new style to one of the D2HML values. In order for the style to be active, it must have a [Behavior](#) property other than **None**.

The active link [Behavior](#) settings are as follows:

Value	Description
None	No hot spot

Value	Description
ContextString	Context string hot spot
Group	Group hot spot
Topic Link	Topic link hot spot
Group Link	Group link hot spot
Keyword Link	Keyword link hot spot
Link Tag	Link tag hot spot
Context ID	Context ID hot spot
Conditional Text	Conditional text hot spot
Contents Title	Contents title hot spot
Topic Properties	Topic properties hot spot
Keyword	Keyword hot spot
Inline Expand	Inline expand hot spot
Inline Popup	Inline pop-up hot spot
Inline Text	Inline text hot spot
Inline Dropdown	Inline drop-down hot spot

Alternative Options for Creating Links and Hot Spots in Word

Using Doc-To-Help Markup Language, although it is the recommended method, is not the only way you can create links and hot spots in your source documents.

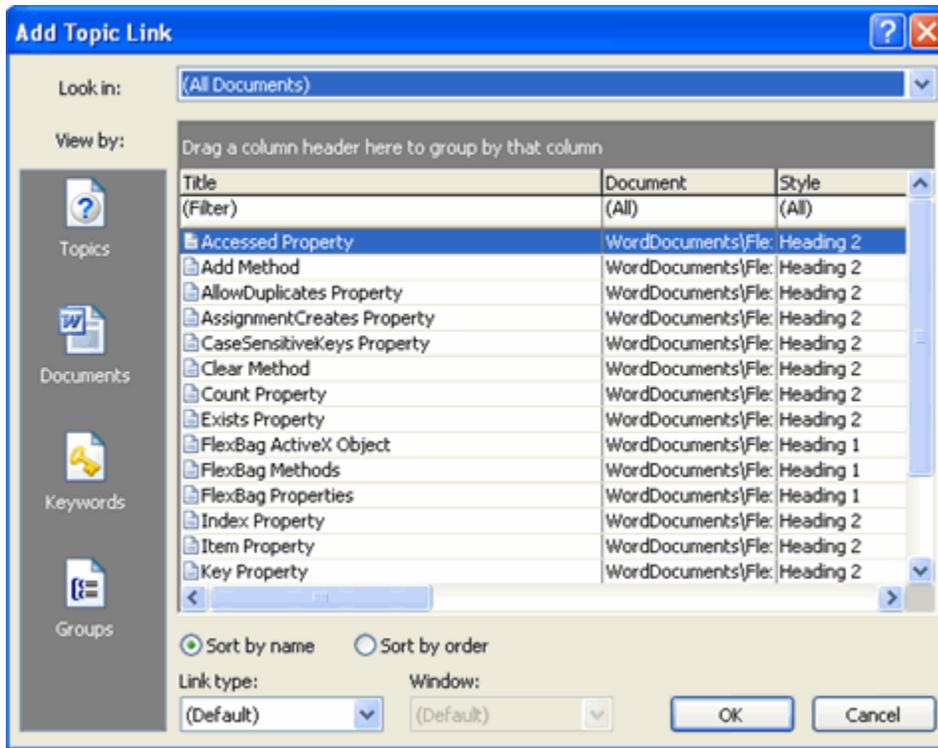
In Microsoft Word, you can use the **Doc-To-Help** toolbar to create topic links, cross-references, margin notes, mid-topic jumps and dynamic links.

Using the Add Topic Link Dialog Box

To add a link to a topic using the **Add Topic Link** dialog box:

1. In Microsoft Word, highlight the word or words you wish to use as your topic hot spot.
2. Click the **Add Topic Link** button .
3. From the **Add Topic Link** dialog box, choose the topic you wish to associate with your hot spot.

Note: To facilitate selecting topics, you can use the radio buttons to sort alphabetically or by order (as they appear in each document).



4. From the **Link Type** drop-down, select the **Jump** or **Popup** option.
5. From the **Window** drop-down, select the window type where you wish to display your topic and click **OK**. For more information on customizing windows, see [Customizing Help Windows](#) (page 353).
6. Save and close the Word document.
7. Compile and view your Help project.

Note that the words you highlighted are now defined as a jump to the associated topic.

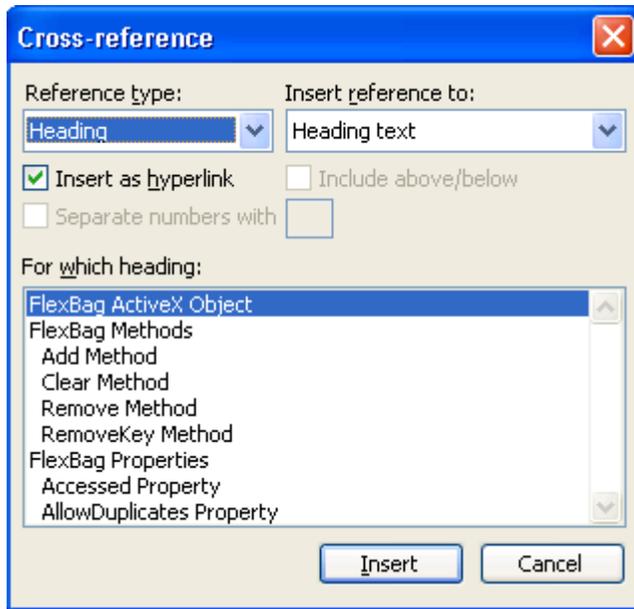
Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Topic Link** menu command.

Creating Cross References

By utilizing Doc-To-Help's **Insert Cross Reference** and **Complete Cross Reference** buttons on the **Doc-To-Help** toolbar in Word, you can automatically add updateable page references in printed manuals and hyperlink jumps in online Help.

Inserting and Completing a Cross Reference

1. In Word, position your cursor where you want the reference to appear.
2. Type the referring text, such as "For more information, see" or "See also" followed by a space.
3. Click the **Insert Cross Reference** button  on the **Doc-To-Help** toolbar. The **Cross Reference** dialog box opens.
4. Set the **Reference Type** set to *Heading*.
5. Set **Insert Reference To** to *Heading Text*.
6. Select the heading you want to refer the reader to.



7. Click the **Insert** button. The heading text is inserted.
8. Click the **Close** button.
9. With the insertion point immediately following the cross-reference (reference field), click the **Complete Cross Reference** button  on the **Doc-To-Help** toolbar.

The page reference is inserted and the heading text is enclosed in quotes.

10. Type a period after the page number, if necessary.

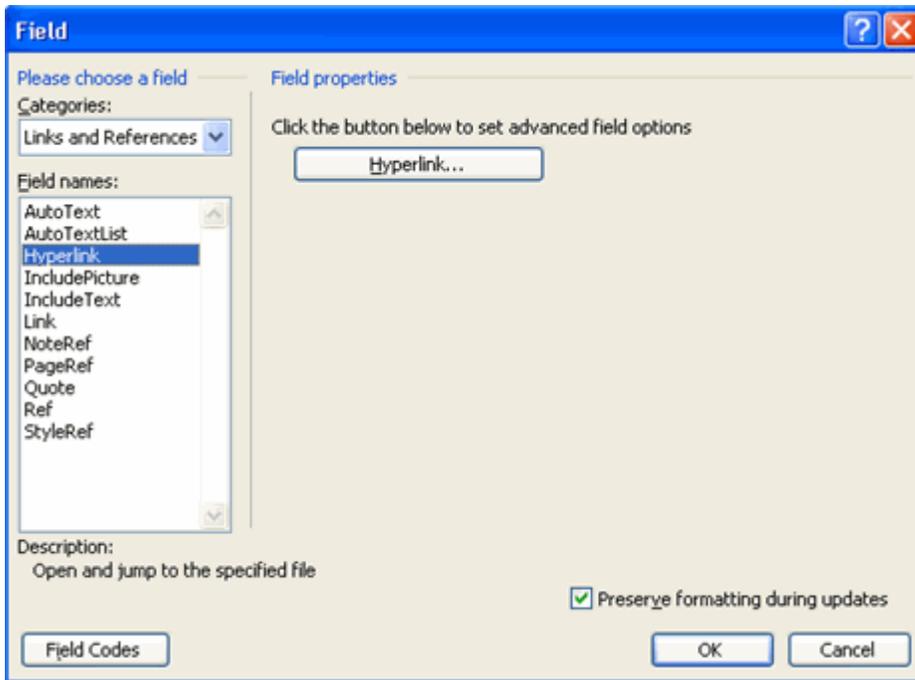
Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Cross Reference** and the **Edit | Complete Cross-reference** menu commands.

Note: The “on page” text, added automatically by Doc-To-Help, can be modified by changing the [OnPage](#) property setting. To locate the [OnPage](#) property, click the **Project** icon, then select **Project Settings** from the left pane.

Using the Word Hyperlink Field to Create Cross Document Links

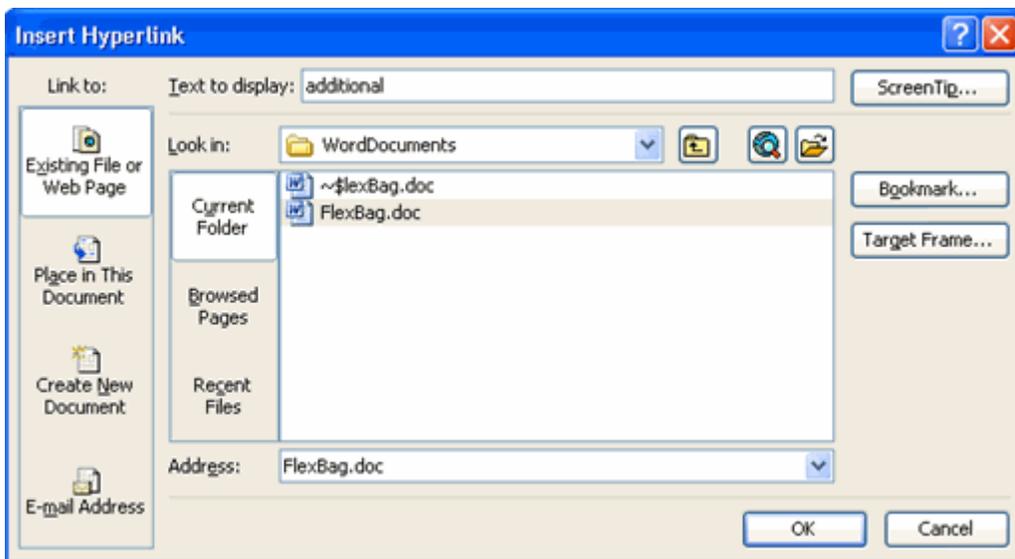
By using the Microsoft Word Hyperlink field you can create cross-references to other documents in your project or to bookmarks within other documents in your project.

1. Select **Insert | Field** from the Word menus.
The **Field** dialog box opens.



2. Select **Links and References** from the categories drop-down.
3. Select **Hyperlink** from the Field names list.
4. Click the **Hyperlink** button.

The **Insert Hyperlink** dialog box opens.



5. Select the document for cross-reference.
6. If you are referencing a bookmark, click the **Bookmark** button.
7. Click **OK**.

FlexBag is a 100%-compatible replacement for VBA/VB5 Collection objects. In addition to all of the features of the standard Collection object, it has the following additional benefits:

- It has `{HYPERLINK "FlexBag.doc" }` which make it possible manipulate keys and their positions more easily, as well as features which make it easy to replace ScriptingDictionaries and achieve the benefits of associative arrays.

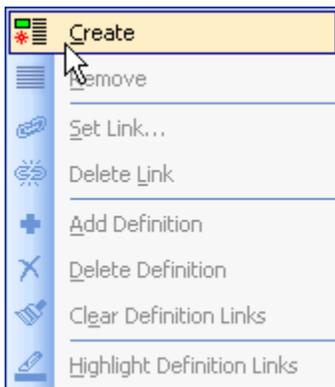
Using Margin Notes in Word

The **Margin Note** drop down makes it easy to place text or graphics in the left margin, next to the main body of the text. Margin notes do not appear in the Help unless you explicitly link them to the text, in which case they become pop-ups. Text is automatically formatted to appear in Margin Note style, smaller than Body Text and italicized.

Adding Margin Notes

The **Create Margin Note** command allows you to place text or graphics in the left margin, next to the main body of the text. Margin notes do not appear in the Help unless you explicitly link them to the text, in which case they become pop-ups. Text is automatically formatted to appear in Margin Note style, smaller than Body Text and italicized.

1. Place your cursor anywhere in the paragraph you want to annotate.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Create** from the drop-down menu.



A two-column table is placed in the document with the body text in the right hand table cell. If Gridlines is selected in the Table menu, you'll see a dotted outline of the table. Note that the gridlines do not print.

4. Type the margin note in the left column of the table.

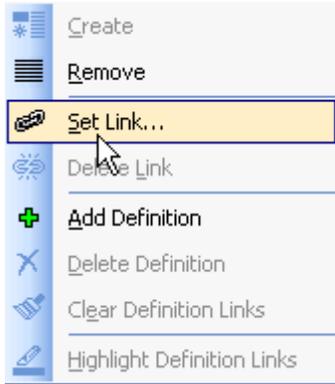
Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Linking to Margin Notes

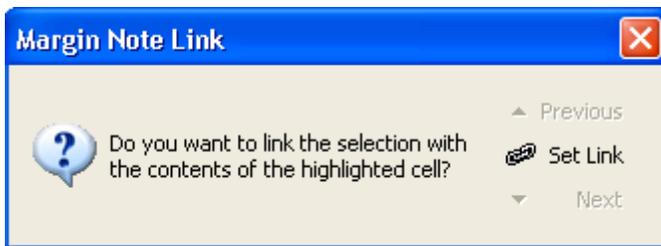
The **Set Link** command creates a hypertext link between text or graphic and the margin note. If you're linking the margin note to text, that text appears underlined and blue in Help. A graphic will look no different as hypertext, so you may wish to add such help-only advice as "Click this picture for more information." When the blue, underlined text or the selected graphic is clicked, the margin note displays in a pop-up window.

1. Click in the right column of the table, that is, in the cell inhabited by the body text.
2. Highlight the text that you want to link to the margin note.
3. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.

4. Select **Set Link** from the drop-down menu.



5. The **Margin Note Link** box opens. If you have more than one margin note, you can select the note you want by using the **Previous** and **Next** icons.

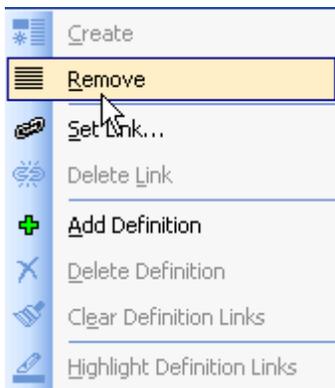


Doc-To-Help places a Word comment next to the text that you have chosen. This comment signals the Doc-To-Help compiler to create a pop-up window for the margin note.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Removing Margin Notes

1. Place your cursor anywhere in the paragraph adjacent to the margin note.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Remove** from the drop-down menu.



Doc-To-Help removes the table and margin note.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Deleting Margin Note Links

1. Place your cursor anywhere inside the margin note link you wish to delete.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Delete Link** from the drop-down menu.

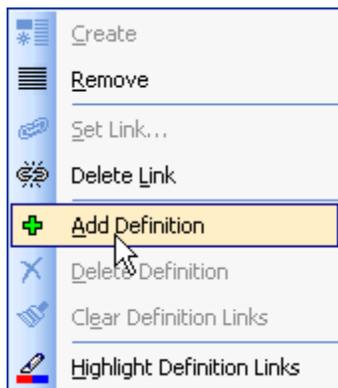


Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Adding Additional Margin Note Definitions

The **Add Definition** command allows you to add multiple margin notes within one paragraph.

1. Place your cursor anywhere in the paragraph you want to annotate.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Add Definition** from the drop-down menu.



Doc-To-Help splits the margin note column adding an additional cell to the left column of the margin note table.

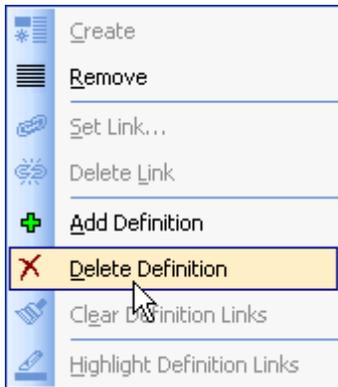
4. Type the margin note in the new cell.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Deleting Margin Note Definitions

1. Place your cursor anywhere in the margin note cell you wish to delete.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.

3. Select **Delete Definition** from the drop-down menu.



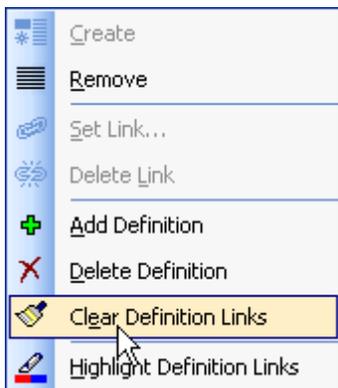
Doc-To-Help removes the margin note cell.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Clearing Margin Note Links

The **Create Margin Note** command allows you to remove all links associated with a specific margin note.

1. Place your cursor anywhere in the margin note cell you wish to modify.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Clear Definition Links** from the drop-down menu.



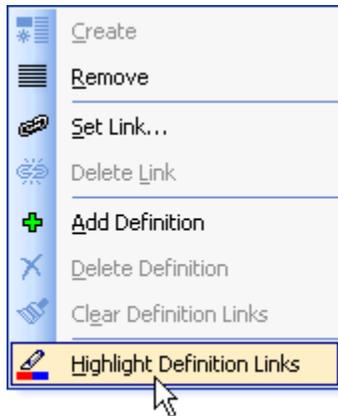
Doc-To-Help removes all links that are associated with that margin note.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Viewing Margin Note Links

In some instances you may have multiple margin notes with multiple links for each note. It can become difficult to distinguish which links are associated with which margin notes. Doc-To-Help allows you to easily view the links for an individual margin note.

1. Place your cursor anywhere in the margin note you wish to view.
2. Click the **Margin Note** button  on the **Doc-To-Help** toolbar.
3. Select **Highlight Definition Links** from the drop-down menu.



Doc-to-Help color codes the margin note cell and the associated links for that margin note.

Alternatively, you can perform the same steps with the **Edit | Margin Note** menu command.

Multiple Paragraphs in Margin Notes

A problem arises with margin notes when you want to have a margin note that contains line breaks. Separating the lines into paragraphs won't work because Doc-To-Help's **Link to Margin Note** command sees just the text up to the first paragraph mark. The easy way to solve this problem is to use line breaks (SHIFT+ENTER) instead of hard returns at the end of each paragraph.

Mid Topic Jumps

The [MidTopic](#) property defines a paragraph style or a topic type as an “in topic” jump location similar to an HTML anchor. By defining a paragraph style or topic type as MidTopic and defining a jump method, end users have jump functionality within a lengthy topic.

Inserting a Mid Topic Jump

1. Open your source document, select or insert your mid topic heading and define it with the MidTopic paragraph style. The mid topic heading must be formatted with a “heading” paragraph on a separate line (not part of the body text).
2. Save the document and open or return to the project editor.
3. In the left pane, select the **Help Targets** item.
4. Choose a Help target from the right pane, and build the Help target.

At this point you have created your mid topic jump location. You can create a hot spot using the **Add Topic Link** button on the **Doc-To-Help** toolbar for Word.

To create a hotspot using the Add Topic Link button:

1. Open the source document where you defined the mid topic location heading.
2. Select your hotspot text.
3. Click the **Add Topic Link** button.

The **Add Topic Link** dialog box opens.

4. Click the **Topics** icon in the **View by:** pane.
5. Select the topic you defined as your mid topic jump location.
6. Click **OK**.

7. Close the source document.
8. In the left pane of the project editor, select the **Help Targets** item.
9. Choose a Help target from the right pane, and build the Help target.

For detailed information on using the Add Topic Links command, see [Using the Add Topic Link Dialog Box](#) (page 319).

Using Dynamic Links

By using Doc-To-Help's Add Dynamic Link feature in Microsoft Word, you can turn any text into a hyperlink that is associated with one or more index keywords or index groups. Simply highlight the desired text, click the **Add Dynamic Link** button on the Doc-To-Help tool bar and select the desired [keywords \(KLinks\)](#) or [keyword groups \(ALinks\)](#) to create your hyperlink association.

For more information on indexing, see [Building an Index](#) (page 341).

Adding a Dynamic Link

To add a link to an index keyword(s) or to an index group using the **Add Dynamic Link** dialog box:

1. In Microsoft Word, highlight the word or words you wish to use as your dynamic link hot spot.
2. Click the **Add Dynamic Link** button .
3. From the **Add Dynamic Link** dialog box, choose the index keyword or index keyword group you wish to associate with your hot spot and click **OK**.

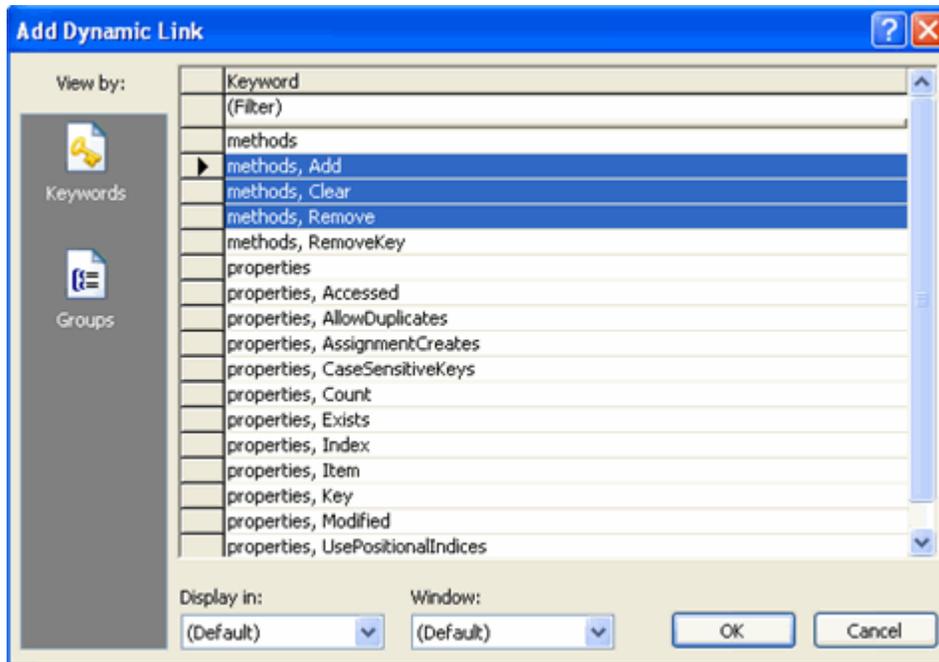
Doc-To-Help supports multiple index keyword selection by using the *Shift* and *Ctrl* keys. Simply press and hold the SHIFT key, for consecutive selections, or CTRL key, for nonconsecutive selections, and choose the record selector buttons to the left of the index keywords as shown below.

4. The **Display in** combo box allows you to override the [TopicsFound](#) property in HTML Help targets on a case-by-case basis.

Note: The default for this option is a pop-up menu.

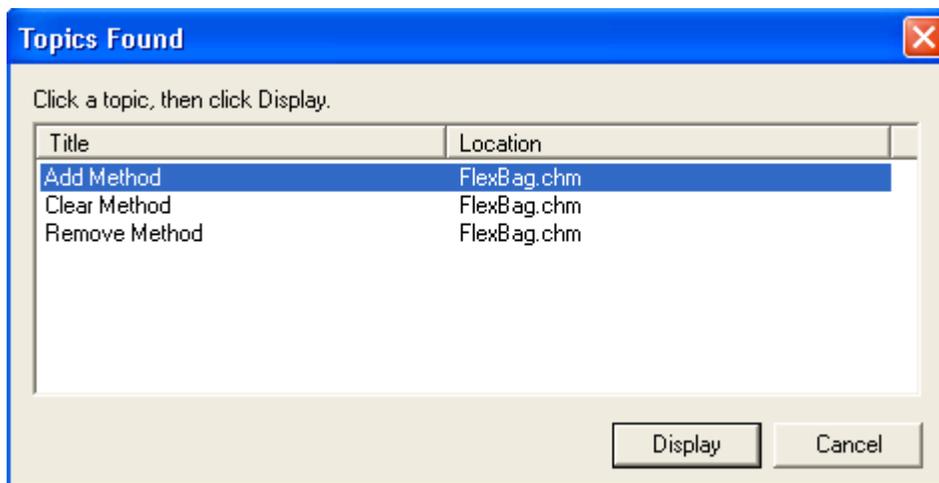
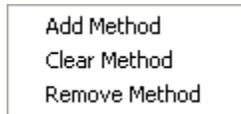
5. If desired, select which window you want your topics to open in from the **Window** combo box.

The window drop-down contains the currently defined help windows. You can modify the contents of this drop-down by creating custom help windows. For more information, see [Customizing Help Windows](#) (page 353).



6. Save and close the Word document.
7. Compile and view your Help project.

Depending on how you set the **Display in** combo box, clicking the hyperlink presents you with either a pop-up or topics found dialog box from which to choose a topic.



Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Dynamic Link** menu command.

Using Image Maps

You can create image maps, or images with hot spot topic links, in both Word and HTML source documents.

Doc-To-Help provides an Image Map Editor for Microsoft Word. See [Using the Image Map Editor in Microsoft Word](#) (page 560) for specific instructions on creating image maps in your Word source documents.

This section explains how to create image maps in HTML source documents.

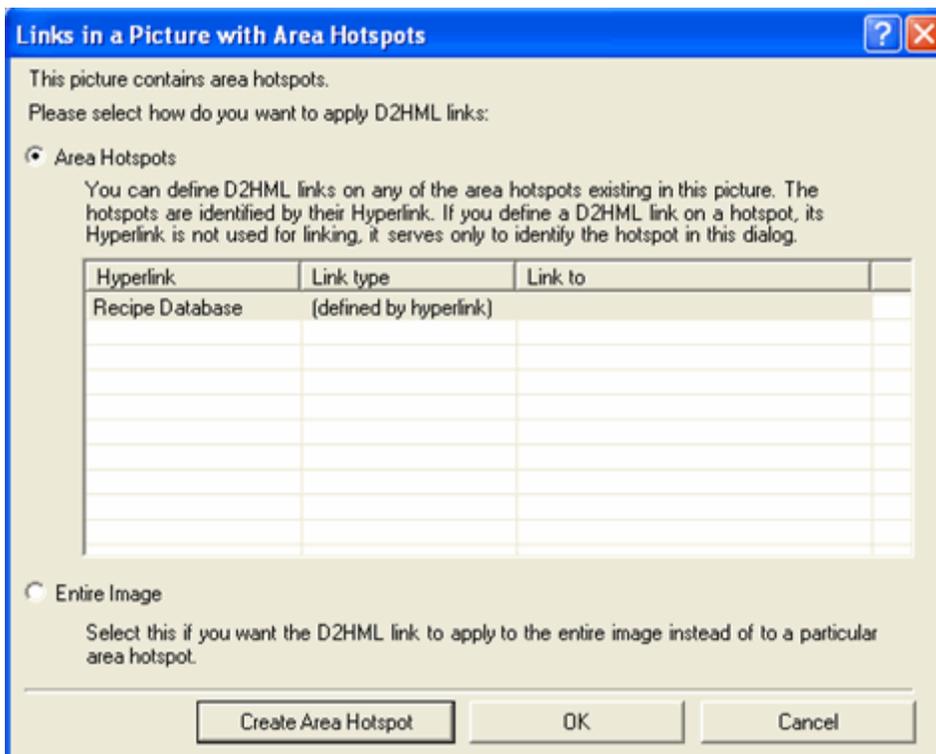
Image maps can be created in HTML source documents by drawing a hot spot and clicking one of the D2HML link buttons on the **D2HML Styles** toolbar in FrontPage and Dreamweaver.

The D2HML attribute of the <AREA> tag is used to specify the topic to which you are linking. If you are using an HTML editor other than FrontPage or Dreamweaver, you can still create an image map and use the <AREA> tag and D2HML attribute to define the links of the hot spots.

A hot spot can be created using any of the five D2HML link types: Jump Link, Popup Link, Keyword Link, Group Link and Glossary Link.

Creating Image Maps with FrontPage

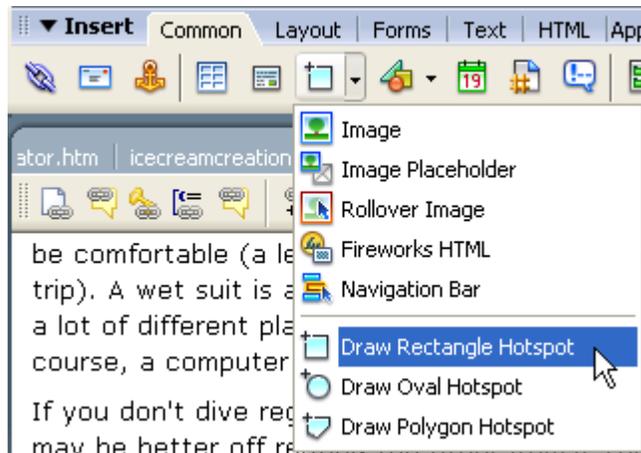
1. Select a graphic in your HTML source document.
2. On the **Pictures** toolbar, click **Rectangular Hotspot**, **Circular Hotspot**, or **Polygonal Hotspot**.
3. Draw the hot spot on your graphic. The **Insert Hyperlink** dialog box appears.
4. Enter a unique, arbitrary name to identify the hotspot in the **Address** text box and click **OK**.
5. Select the hot spot and click one of the D2HML link buttons on the **D2HML Styles** toolbar. You can use any of the five D2HML link types: jump link, pop-up link, keyword link, group link, and glossary link. The **Links in a Picture with Area Hotspots** dialog box appears.



6. Select a hot spot in the **Hyperlink** column, or you can select **Entire image** if you want the D2HML link to apply to the whole image.
7. Click **Create Area Hotspot**. The D2HML dialog box appears.
8. Click the **Browse** button to locate and select the topic to which you want to link.
9. Click **OK** to close the dialog boxes.

Creating Image Maps with Dreamweaver

1. Select a graphic in your HTML source document.
2. On the **Common** tab of the **Insert** toolbar, click the drop-down arrow of the **Images** button and select one of the following: **Draw Rectangle Hotspot**, **Draw Oval Hotspot** or **Draw Polygon Hotspot**.



3. Draw the hotspot on your graphic.
4. With the hotspot still selected, click one of the D2HML link buttons on the **D2HML Styles** toolbar. You can use any of the five D2HML link types: jump link, pop-up link, keyword link, group link, and glossary link.
5. In the D2HML dialog box, click the **Browse** button to locate and select the topic to which you want to link.
6. Click **OK** to close the dialog boxes.

Creating Image Maps Using Other HTML Editors

1. Insert a graphic in your HTML source document and set the *usemap* attribute of the tag to the map name, for example:

```

```

2. Specify the map name using the <MAP> tag:

```
<map name="Map">
```

3. Use the <AREA> tag to specify the hot spot shape, coordinates and the type of D2HML link to use. Additionally, use the D2HML attribute to specify the topic to which you want to link.

```
<area shape="rect" coords="26,5,168,384" class="C1HJump" d2hml="tag=YUM's Special Features">
```

For more information on the D2HML attribute, see [Using the D2HML Attribute in HTML Source Documents](#) (page 244).

Creating a Glossary

A glossary is a list of specialized words with their definitions, often placed at the end of a book. To assist in developing an online and printed glossary, Doc-To-Help automatically creates a glossary document with two formatted glossary entry examples whenever a new project is created. All you need to do is open it and add and format your entries.

Doc-To-Help can convert text in your source documents to hyperlinks which, when clicked, open DHTML pop-ups containing the corresponding glossary definitions. You may also choose not to use this functionality for some or all terms.

Additionally, there may be instances of a word or phrase in your source documents where you would like to link to a glossary term. Doc-To-Help allows you to create pop-up links using the **Glossary Links** button on the **D2HML Styles** toolbar to link specific text to glossary terms.

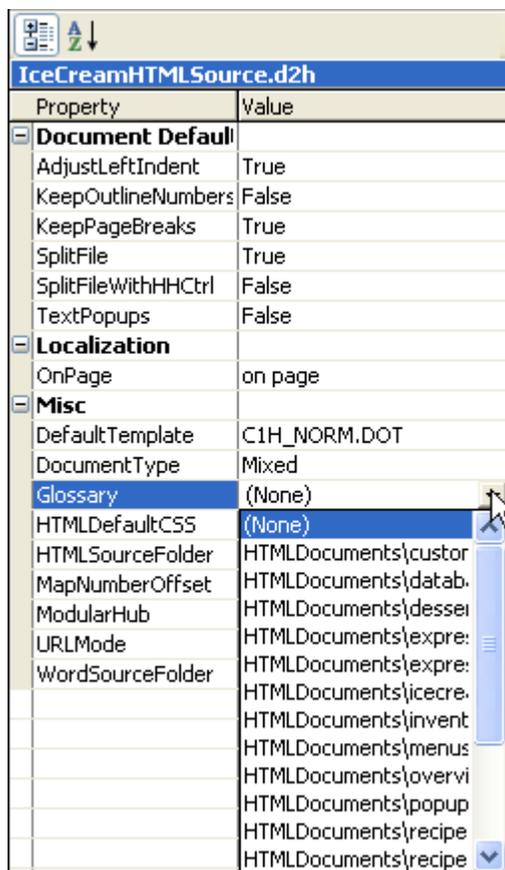
Glossary entries can be viewed by clicking the **Glossary Terms** item in the project editor tree.

The glossary document remains as your final document in the Help project by default.

Specifying a Glossary Document

The glossary document is specified in the Project.[Glossary](#) property. To assign a glossary document to the project:

1. On the Icon bar, click **Project**.
2. Select **Project Settings** from the left pane.
3. From the property pane, click the drop-down arrow next to the [Glossary](#) property and select your glossary document.



Opening the Glossary

1. Highlight the **Document** item on the project editor tree list.
2. Select then right-click the glossary document and choose **Open** from the shortcut menu. The document is opened within Microsoft Word or your default HTML editor.

Editing the Glossary in Doc-To-Help

1. Highlight the **Document** item on the project editor tree list.
2. Select and right-click the glossary document and choose **Open** from the shortcut menu. The document is opened within Word or your default HTML editor and you can begin editing it. Alternatively, if the glossary document is a Word document, you can right-click it and choose **Edit** to open it within the right pane of the project editor.

Adding and Formatting Glossary Entries

Once you open the glossary document, you can begin adding and formatting glossary entries. Add the first glossary heading on the first line following the **Glossary of Terms** heading, and add its definition on the second line.

You may choose to format some or all glossary headings with the **Glossary Heading** paragraph style, which has the [AutoGlossaryLinks](#) property set to **True**. Note that the two sample glossary headings automatically created in the glossary document are formatted with this style. In this case, when you build a Help file, Doc-To-Help examines the project documents for text that matches the glossary headings, converting these matches into hyperlinks. If there are multiple instances in a topic where the text matches a glossary heading, Doc-To-Help creates the pop-up link at the first instance of the text.

Note: The pop-up link can be created at an instance other than the first by using the **D2HNoGloss** character style. See [Filtering Topics from the Glossary](#) (page 335) for more information.

Whenever the user clicks one of the hyperlinks, a DHTML pop-up opens containing the corresponding glossary entry definition.

If you do not want to use this functionality for some or all terms, you can format your glossary headings with the **Glossary Heading (no auto links)** paragraph style in the glossary document. The glossary still appears in your Help file, but text matching the glossary headings in the source documents will not automatically become hyperlinks when the Help is compiled. You can [create glossary links](#) (page 338) manually using D2HML in those instances where you want to link a word or phrase to a glossary term.

If you create your own glossary heading style and do not want to use the automatic glossary link functionality, set its [AutoGlossaryLinks](#) property to **False**.

Note: If one of the predefined glossary heading styles cannot be used, you can create your own glossary heading paragraph style. Set its [TopicType](#) property to **Glossary Term Definition**, which has the [AutoGlossaryLinks](#) property set to **True** by default, to use the automatic glossary link functionality. Set the [AutoGlossaryLinks](#) property to **False** if you do not want to use the automatic glossary link functionality. See [Using Styles in Doc-To-Help](#) (page 211) for more information on creating paragraph styles and using Topic Types.

The glossary definitions are formatted with the **C1H Popup Topic Text** style. This creates the DHTML pop-up that opens when the glossary link is clicked.

To add and format a glossary entry:

1. Open the glossary document.
2. Enter the glossary heading on the first line following the **Glossary of Terms** heading.

- Format this heading with the **Glossary Heading** paragraph style if you want Doc-To-Help to automatically create pop-up links in your source documents for text that matches the glossary heading. Use the **Glossary Heading (no auto links)** paragraph style if you do not want the links created automatically, and then you can format text with a D2HML hot spot to [create glossary links](#) (page 338) manually.
- Enter the glossary definition on the second line and format it with the **C1H Popup Topic Text** style. The entries should look like the following example.

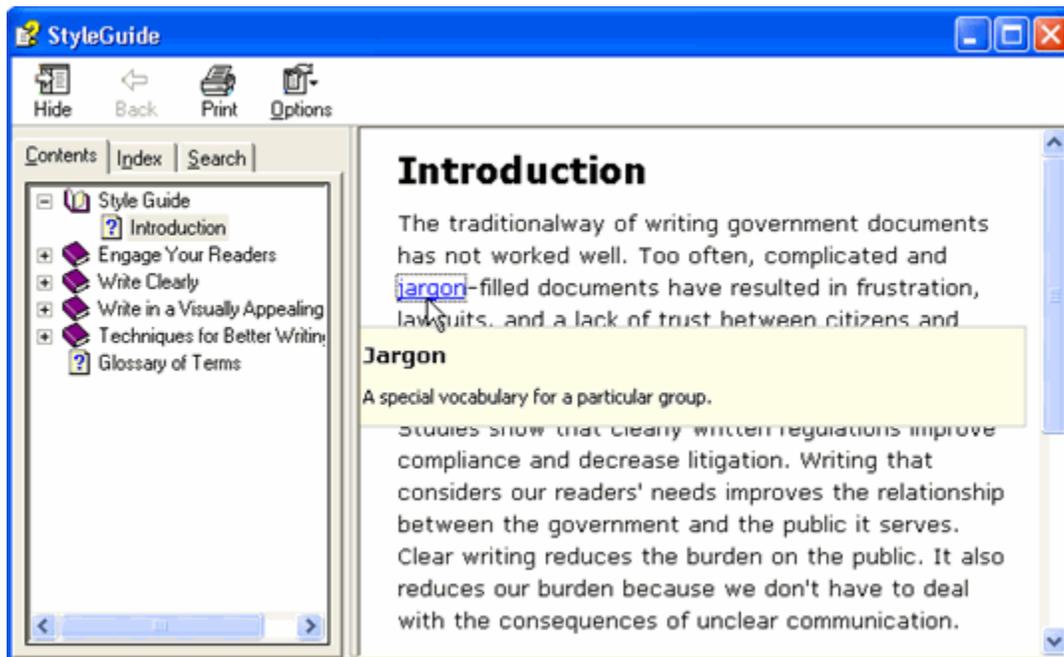
Active voice¶

A style of writing in which the subject of a sentence performs or causes the action expressed by the verb.¶

Jargon¶

A special vocabulary for a particular group.¶

- After you have finished adding and formatting your glossary entries, save and close the **Glossary** document.
- Make sure that the value of the [Glossary](#) property, located under **Project Settings**, is set to your **Glossary** document.
- Select the **Build** menu and click **Rebuild Target**. It is important to rebuild the entire target to ensure that recent changes will be included. After the target is compiled, click the **View Target** button. If you used the **Glossary Heading** paragraph style, wherever the Help text matches your glossary entries (**Glossary Heading**), Doc-To-Help has formatted the text as a Dynamic HTML pop-up. If you used the **Glossary Heading (no auto links)** paragraph style, hyperlinks were not automatically created for the matching text. If you manually created glossary links using a D2HML hot spot, a DHTML pop-up appears when you click them.



Filtering Topics from the Glossary

In some cases, your topics may contain multiple instances of a specific glossary term. If you have Doc-To-Help automatically create pop-up links for text matching glossary terms, it inserts the glossary pop-up at the first instance of the term in each topic, by default. If you want to modify your Doc-To-Help project so that it uses an instance of the glossary term other than the first, you can use the **D2HNoGloss** character style.

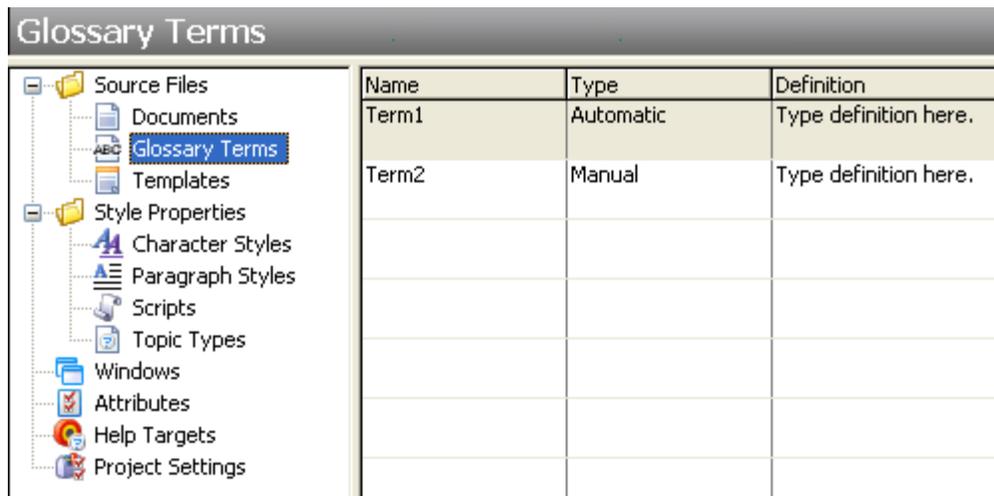
Simply apply the **D2HNoGloss** character style to each instance of a glossary term in a topic until you come to the term you want to designate as the pop-up. Do not apply **D2HNoGloss** to this term. You do not have to apply the **D2HNoGloss** style to subsequent terms because Doc-To-Help creates the pop-up at the first instance of the term that is not formatted with the **D2HNoGloss** style.

Note: If all instances of the glossary term are formatted with the **D2HNoGloss** style, Doc-To-Help does not create a glossary pop-up in that topic.

Viewing Glossary Terms

To view the glossary terms in the Doc-To-Help project editor:

1. Click the **Project** icon.
2. Click the **Glossary Terms** item in the project editor tree.



The screenshot shows the 'Glossary Terms' window in the Doc-To-Help project editor. On the left is a tree view with 'Glossary Terms' selected. On the right is a table with three columns: Name, Type, and Definition.

Name	Type	Definition
Term1	Automatic	Type definition here.
Term2	Manual	Type definition here.

Automatic and Manual Terms

Glossary terms formatted as **Glossary Heading**, or with a paragraph style that has [AutoGlossaryLinks](#) set to **True**, are listed as *Automatic* in the **Type** column since Doc-To-Help automatically creates pop-up links for text matching the term.

Glossary terms formatted as **Glossary Heading (no auto links)**, or with a paragraph style that has [AutoGlossaryLinks](#) set to **False**, are listed as *Manual* in the **Type** column since you must manually create pop-up links to them.

Adding and Sorting Glossary Terms in Word

The **Doc-To-Help** toolbar in Word provides an alternative way for adding glossary terms to a Word glossary document. You can also alphabetize the terms simply by clicking a button.

Using the Add Glossary Terms Tool in Word

The **Add Glossary Terms** tool provides you with an alternative way to add entries to your glossary Word document without having to open the glossary document itself. In this way, you can add new entries, as you see them, from any source document.

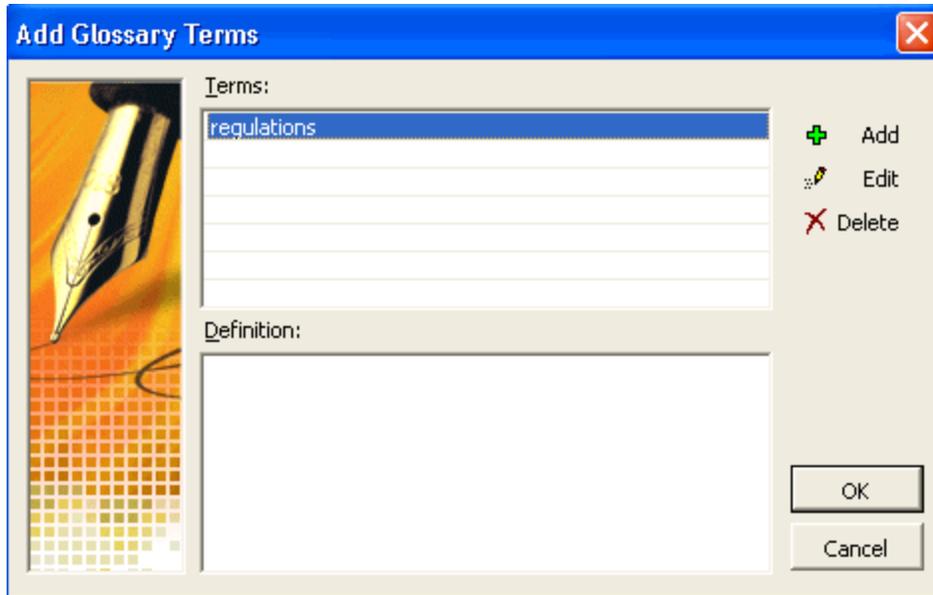
To add glossary entries:

1. Open any of your Word source documents.

- Place your cursor in front of the word that you want to add to the glossary and click the **Add Glossary Terms** button .

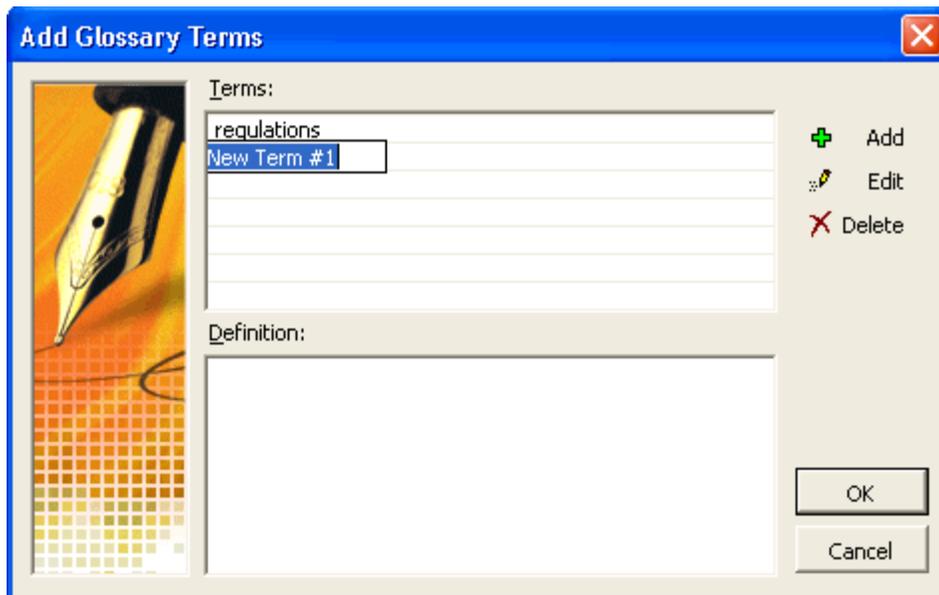
The **Add Glossary Terms** dialog opens with the word you selected in the terms box.

- Click inside the definition box and type the definition of the term.

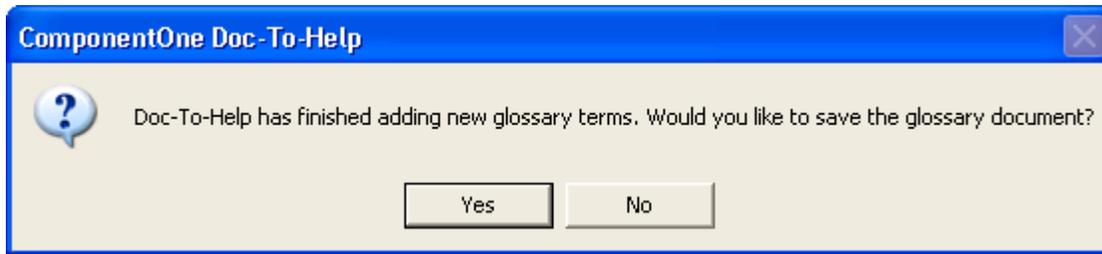


At this point, you can close the **Add Glossary Terms** dialog by clicking **OK**, or you can add additional terms by clicking the **Add** button.

- If you click the **Add** button, you are prompted to add new terms and definitions.



When you click the **OK** button you are prompted with the following message.



Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Glossary Terms** menu command.

Sorting Glossary Entries in Word

To make alphabetizing your glossary terms simple, Doc-To-Help automatically sorts your glossary entries when you click the **Sort Glossary** button .

To sort your glossary entries:

1. Open the Glossary source document.
2. Click the **Sort Glossary** button.

The glossary entries are automatically sorted alphabetically.

3. **Save** and **Close** the glossary document.

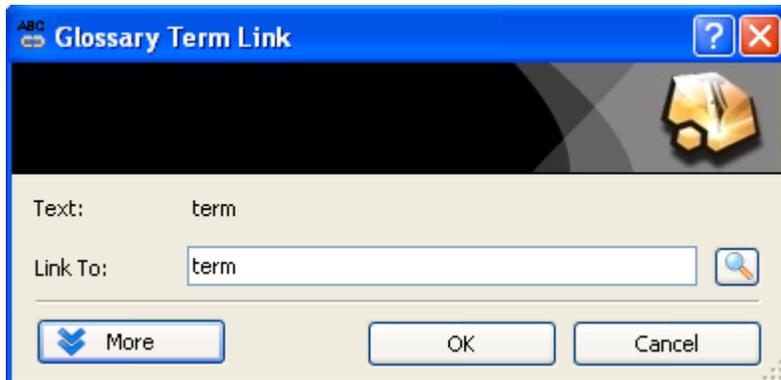
Alternatively, you can perform the same steps with the **Tools | Sort Glossary** menu command.

Note: If you add additional entries after performing the sort, it will be necessary to resort the glossary.

Creating Glossary Links

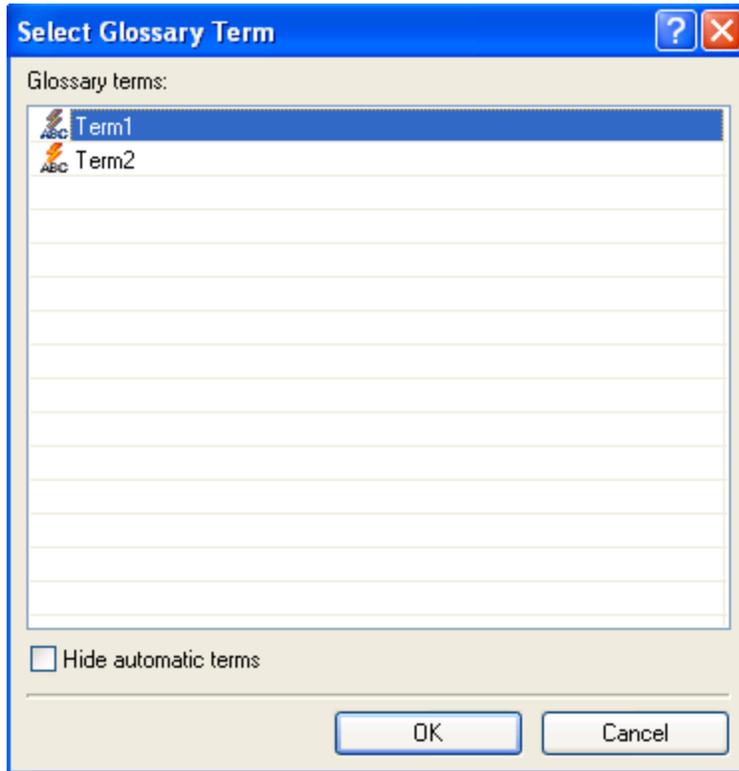
There may be instances in your source documents where you want to link a word or phrase to a glossary term. You can manually create the link using the **Glossary Links** button on the **D2HML Styles** toolbar in Microsoft Word, FrontPage or Macromedia Dreamweaver.

1. Select the word or phrase you want to link to a glossary term.
2. On the **D2HML Styles** toolbar, click the **Glossary Link** button  to open the **Glossary Term Link** dialog box.



Note: If the term you are linking to matches the link text, you can simply click **OK** and the link is created.

3. Click the **Browse** button. The **Select Glossary Term** window appears.
4. Select the glossary term to which you want to link.



Note: The **Hide automatic terms** checkbox is checked by default so only *Manual* terms are visible to choose from. If you need to link to an *Automatic* term, uncheck the **Hide automatic terms** checkbox. See [Viewing Glossary Terms](#) (page 336) for more information on *Automatic* and *Manual* terms.

5. Click **OK** to close the **Select Glossary Term** window, and click **OK** again to close the **Glossary Term Link** dialog box.
6. Save and close the glossary document.
7. In the Doc-To-Help project, select the **Build** menu and click **Rebuild Target**.
8. Once the target is compiled, click the **View Target** button and click the word or phrase where you created the **Glossary Link**. A DHTML pop-up opens containing the corresponding glossary entry description.

Building an Index

To construct an index for the reader, such as the Index tab in HTML Help, you define a set of keywords and associate each with one or more Help topics. There are several ways to create index keywords:

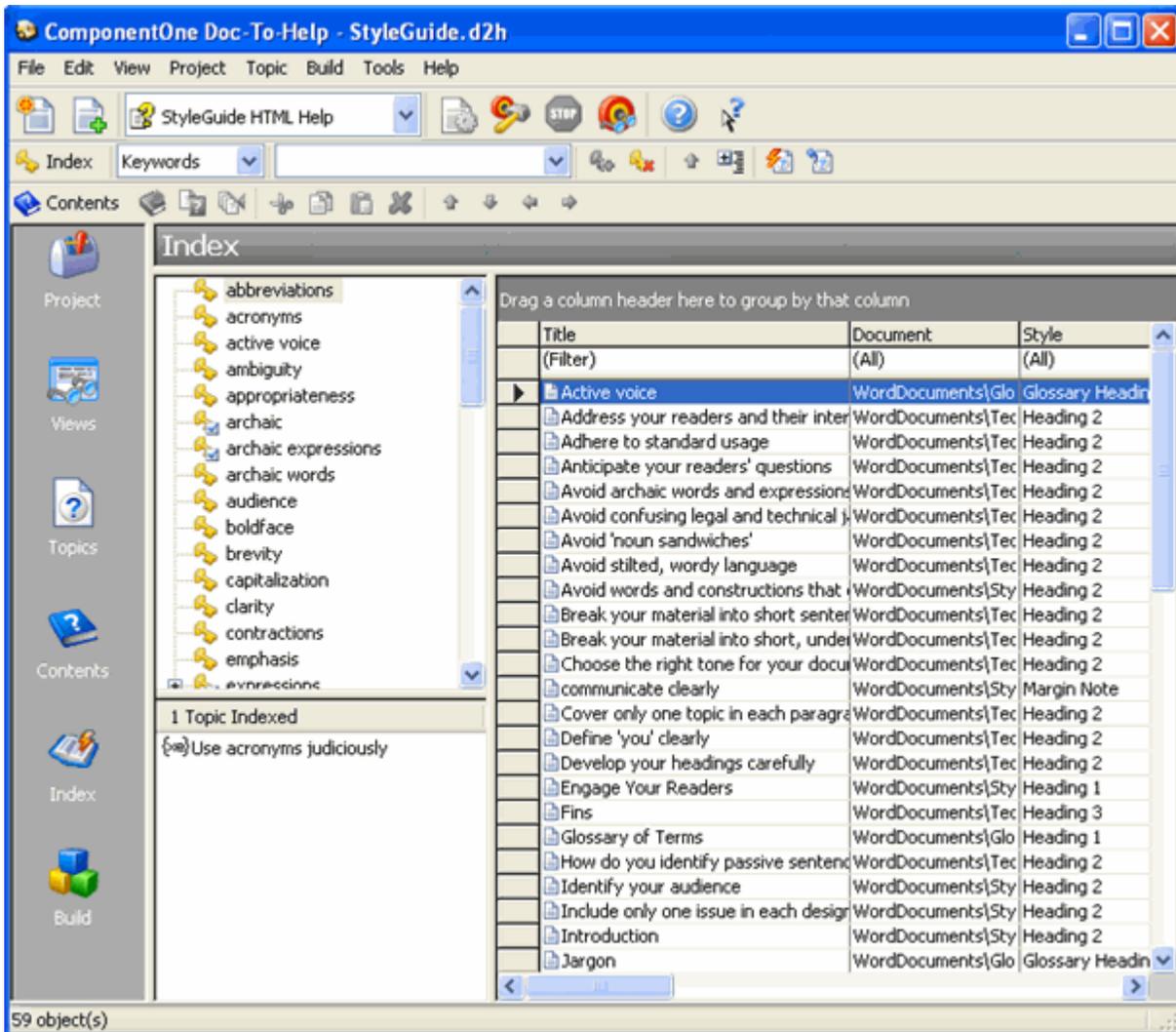
1. Using [Doc-To-Help Markup Language \(D2HML\)](#) (page 233).
2. Automatically, by setting the appropriate property in a paragraph or character style.
3. Manually, by typing them into ComponentOne Doc-To-Help 2006.
4. Natively, by using the inherent capabilities of Microsoft Word (XE fields) when using Word source documents.
5. Programmatically, by writing script code.

Likewise, you can use any of these methods to associate a keyword with a Help topic.

You can also organize Help topics according to a private index that the reader never sees. For example, you may want to mark certain topics as being for advanced users, then provide a hot spot that displays them in a list (like an ALink macro in WinHelp). To construct the private index, you use group names instead of index keywords.

Project Editor Index View

The **Index** view provides an interface for associating index keywords or named groups with specific Help topics. Index keywords are visible to the reader; only the author sees named groups. The upper left pane contains the index keywords that the reader can see while the lower left pane shows which topics are indexed to the keyword selected in the upper left pane. Note that the topics that are indexed by virtue of XE fields using Word are identified with the  symbol in this pane. The right pane displays a sorted list of topics that can be indexed. The topics can be sorted and filtered in the same manner as in the Topic window.



Using the Index Toolbar

On the **View** menu, choose **Toolbars**, then click **Index**. A check mark next to the command indicates that the toolbar is displayed.

After a Doc-To-Help project has been compiled, you can use the **Index** toolbar to view index collections and modify the Help project index.



From left to right, the **Index** toolbar contains the following commands:

Index Collections	Allows you to view the groups or keywords collections
Index Elements	Indicates/selects which index element is currently being viewed
Add	Adds an index element
Delete	Deletes an index element
Up One Level	Moves an index element up one level
Show Secondary	Shows secondary or child index elements

Index Topics	Adds the currently highlighted topic to the current index keyword
Unindex Topics	Removes the currently highlighted topic from the current index keyword

Adding Index Keywords Manually

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Keywords**.
3. Right-click any primary keyword in the upper left pane of the Index window and choose **New** from the shortcut menu.
4. The focus is moved to the **Index Elements** textbox. Replace <new> with the keyword you wish to add to the index and press **Enter** or click the adjacent check mark button.

Note that after pressing **Enter**, the keyword is added to the index list. The focus remains in the **Index Elements** textbox, allowing you to add multiple keywords.

Alternatively, you can add a new keyword by clicking the **Project** menu, choosing **New**, then clicking **Keyword**.

Creating Index Keywords Automatically Using Styles

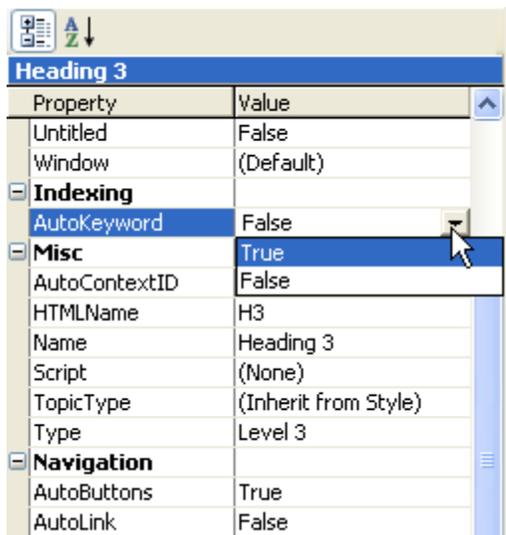
For any active paragraph style, Doc-To-Help can create index keywords from topic titles and automatically associate them with the appropriate topic. Similarly, for character styles, Doc-To-Help can create index keywords from hot spot text and automatically associate them with the topic that contains the hot spot.

Note: Automatically created keywords will not actually be added to your project until you build the project.

For more information on using styles, see [Using Styles in Doc-to-Help](#) (page 211).

To enable automatic indexing for a paragraph style:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** item from the left pane.
3. Select the paragraph style to be edited.
4. In the paragraph style property pane, toggle the [AutoKeyword](#) property to **True**. Index keywords are then created from topic titles formatted with this paragraph style and are associated with the appropriate topic.



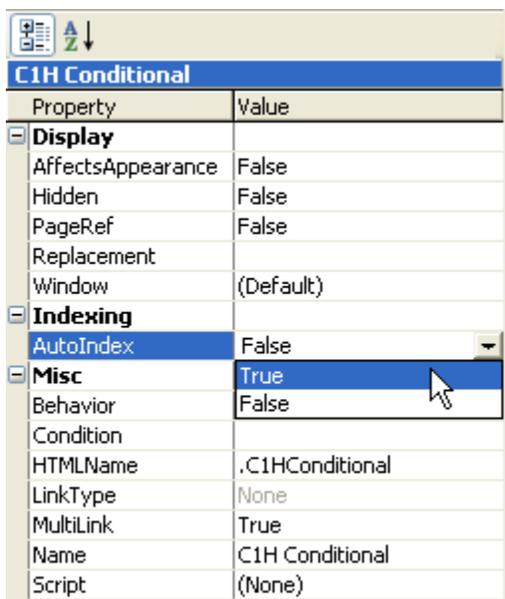
Note: This setting is not available for styles whose **Outline level** is set to *Body text*.

Note that the topics that are indexed in this manner are identified with the  symbol in the lower left pane of the **Topics** icon view.

Once keywords have been automatically added to a project by setting [AutoKeyword](#) to **True** and building the project, subsequently setting [AutoKeyword](#) to **False** will not automatically remove them; you must remove them manually.

To enable automatic indexing for a character style:

1. On the Icon bar, click **Project**.
2. Select the **Character Styles** item from the left pane.
3. Select the character style to be edited.
4. In the character style property pane, toggle the [AutoIndex](#) property to **True**. Index keywords are then created from text formatted with this character style and are associated with the topic that contains the text.



Note that the topics that are indexed in this manner are identified with the  symbol in the lower left pane of the **Topics** icon view.

Adding Secondary Index Keywords Manually

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Keywords**.
3. Select the primary keyword in the upper left pane or in the **Index Elements** list to which you wish to add a secondary keyword.
4. Click the **Show Secondary** button on the **Index** toolbar. The **Index Elements** textbox is cleared.
5. Enter a new secondary keyword, then press **Enter** or click the adjacent check mark button. The new keyword is added to the upper left pane of the Index window.

Alternatively, you can add a new secondary keyword by highlighting the primary keyword, clicking the **Project** menu, choosing **New**, then clicking **Keyword**. Enter the new keyword in the **Index Elements** textbox, and press **Enter**.

Note that after pressing **Enter**, the secondary keyword is added to the index list below the parent primary keyword. The focus remains in the **Index Elements** textbox, allowing you to add multiple secondary keywords to this primary keyword.

Renaming an Index Keyword

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Keywords**, and ensure that keywords (not groups) are displayed in the upper left pane.
3. In the upper left pane, right-click the desired keyword, then click **Rename**.
4. Type the new keyword name, then press **Enter**. Doc-To-Help moves the keyword to preserve alphabetical order, if necessary.

Alternatively you can select the keyword and then click it a second time so that the cursor blinks at the end of it, type the new keyword name, and press **Enter**.

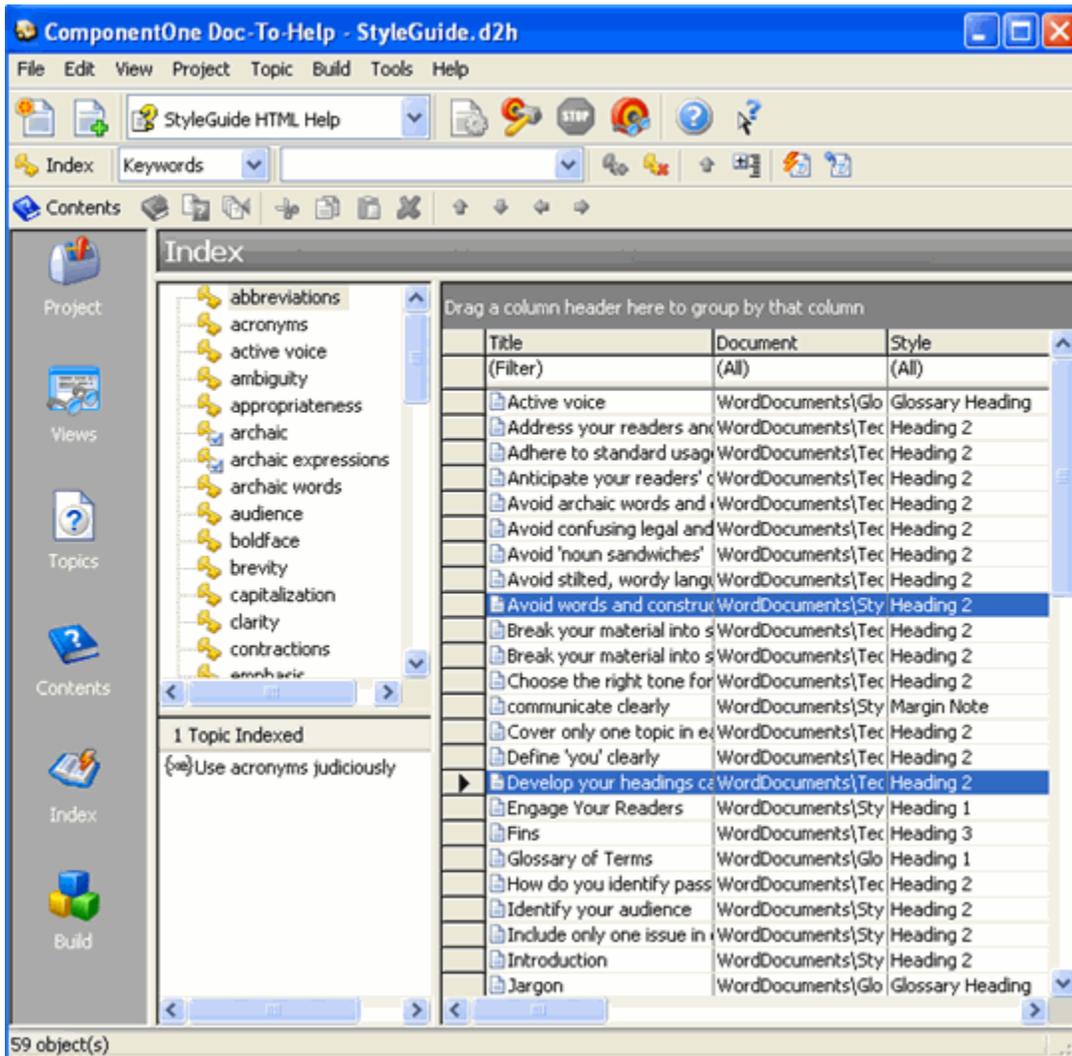
Assigning Topics to an Index Keyword

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Keywords**.
3. Select the desired keyword in the upper left pane of the Index window.

For large indexes, you can save time by using the auto-completion feature of the **Index Elements** list on the **Index** toolbar. For more information, see [Using the Index Elements Auto-Completion Feature](#) (page 350).

4. Select one or more topics from the right pane that you wish to attach to the index keyword.

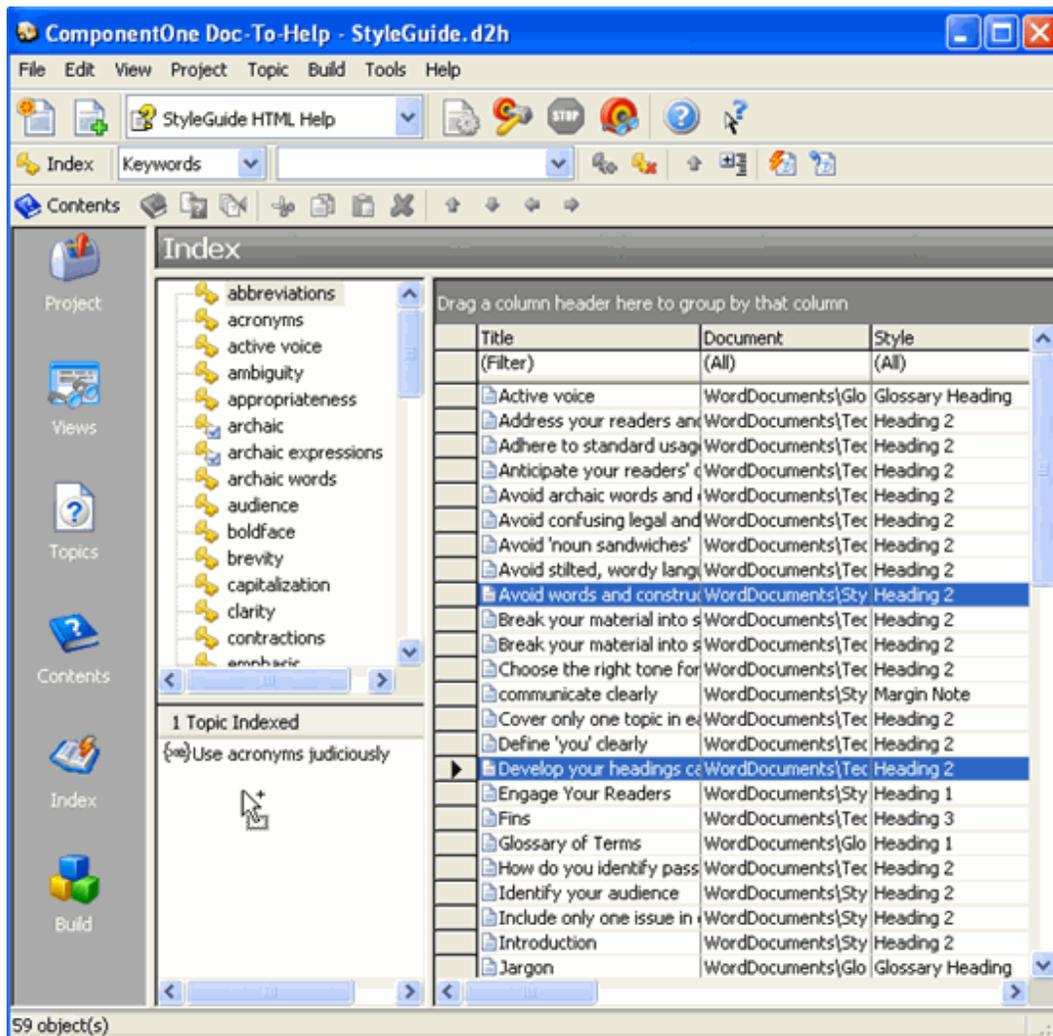
Doc-To-Help supports multiple topic selection by using the *Ctrl* keys. Simply press the **Ctrl** key, for nonconsecutive selections, and choose the record selector buttons to the left of the topic titles as shown below.



5. Select and right-click any of the highlighted topics and choose **Index** from the shortcut menu or click the **Index Topics** button on the **Index** toolbar.

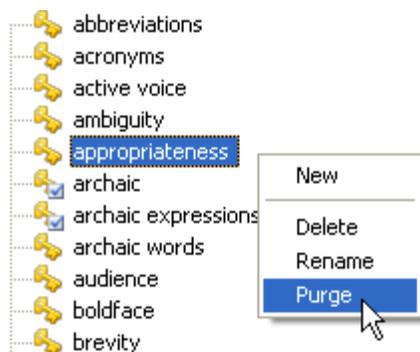
Note that the selected topics are added to the list of indexed topic for those keywords in the bottom left pane. Topics that are indexed in this manner are identified with the  symbol.

Alternatively, you can select the keyword and the topics and “drag and drop” them into the Topics Indexed pane.



Removing Unused Index Keywords

Keeping your project index free of unused index keywords can be time consuming. Even though Doc-to-Help does not use these keywords in your Help targets, it is a good practice to remove unused and out of date keywords. To purge your index of unused keywords, right click anywhere within the keyword pane and select **Purge**.



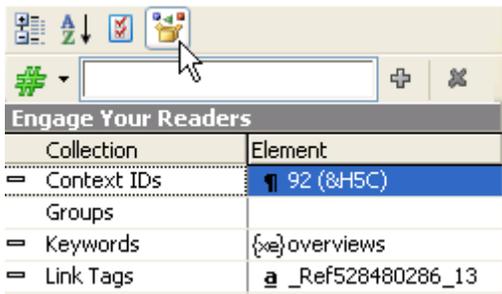
Warning: Index keywords cannot be recovered once they are purged.

Adding an Index Keyword to a Topic

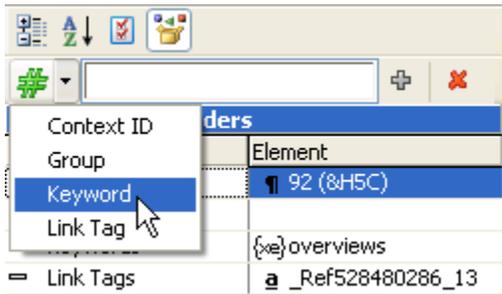
In some cases, choosing a topic and associating an index keyword with it is more efficient than creating index keywords and associating topics to each. This “topic to keyword” workflow makes it easier to create index keywords based on the existing topic titles. After a Doc-To-Help project has been compiled, you can use the **Topics Property** toolbar to insert index keywords.

To associate a topic keyword with a specific topic:

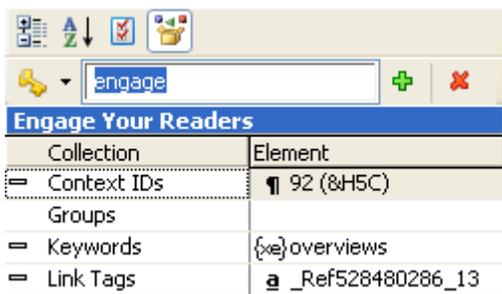
1. On the Icon bar, click **Topics**.
2. Select the **All Topics** item from the left pane.
3. Select the desired topic for editing from the right pane.
4. Click the **Collections** button on the **Topics Property** toolbar.



5. Choose **Keyword** from the **Collection Type** drop-down.



6. Type the keyword you want to associate into the **Element Textbox** and press **Enter**.



Note that the keyword has been associated with the highlighted topic and is displayed to the right of the **Keywords** collection.

Keywords created in this manner are automatically added to the main **Index** tab of Doc-To-Help and associated with the current topic. If you enter a keyword that already exists, Doc-To-Help associates the current topic with the existing keyword.

Note: You cannot specify secondary index keywords in the topic property pane.

Creating Topic Groups

To create a named topic group manually:

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Groups**.
3. Click the **Index Elements** textbox and enter a group name to be added to the index.
4. Press **Enter** or click the adjacent check mark button.

Note that after pressing **Enter** the topic group is added to the topic group list. The focus remains in the **Index Elements** textbox, allowing you to add multiple topic groups.

Alternatively, you can add a new topic group by clicking the **Project** menu, choosing **New**, then clicking **Group**.

To create a named topic group within a script:

See [Creating Topic Groups in a Script](#) (page 446).

Renaming a Group

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Index** toolbar is set to **Groups**.
3. In the upper left pane, right-click the desired group, then choose **Rename** from the shortcut menu.
4. Type the new group name, then press **Enter**.

Note that Doc-To-Help moves the group to preserve alphabetical order, if necessary.

Assigning Topics to a Group

1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Icon** toolbar is set to **Groups**.
3. Select the desired topic group in the upper left pane of the Index window.

For large group lists, you can save time by using the auto-completion feature of the **Index Elements** list on the **Index** toolbar. For more information, see [Using the Index Elements Auto-Completion Feature](#) (page 350).

4. Select one or more topics from the right pane that you wish to add to the topic group.

Doc-To-Help supports multiple topic selection by using the **Ctrl** keys. Simply press the **Ctrl** key for nonconsecutive selections, and choose the record selector buttons to the left of the topic titles.

5. Select and right-click any of the highlighted topics and choose **Index** from the shortcut menu or click the **Index Topics** button on the **Index** toolbar.

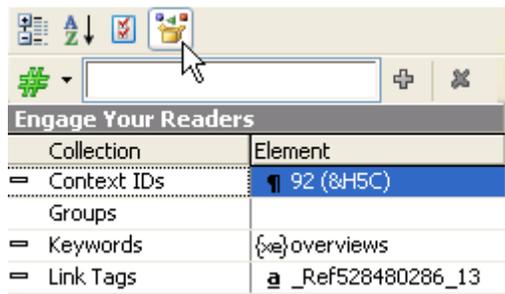
Note that the selected topics are added to the list of indexed topics for that topic group in the lower left pane.

Adding a Group to a Topic

To associate an index group with a specific topic:

1. On the Icon bar, click **Topics**.

2. Select the **All Topics** item from the left pane.
3. Select the desired topic for editing from the right pane.
4. Click the **Collections** button on the **Topics Property** toolbar.



5. Choose **Group** from the **Collection Type** drop-down.
6. Type the group name to be associated with the topic into the **Element Textbox** and press **Enter**.

Note that the group has been associated with the highlighted topic and is displayed to the right of the **Groups** collection.

Groups created in this manner are automatically added to the main **Index** tab of Doc-To-Help and associated with the current topic. If you enter a group that already exists, Doc-To-Help associates the current topic with the existing group.

Removing Topics from Index Keywords or Topic Groups

To remove a topic from an index keyword or group:

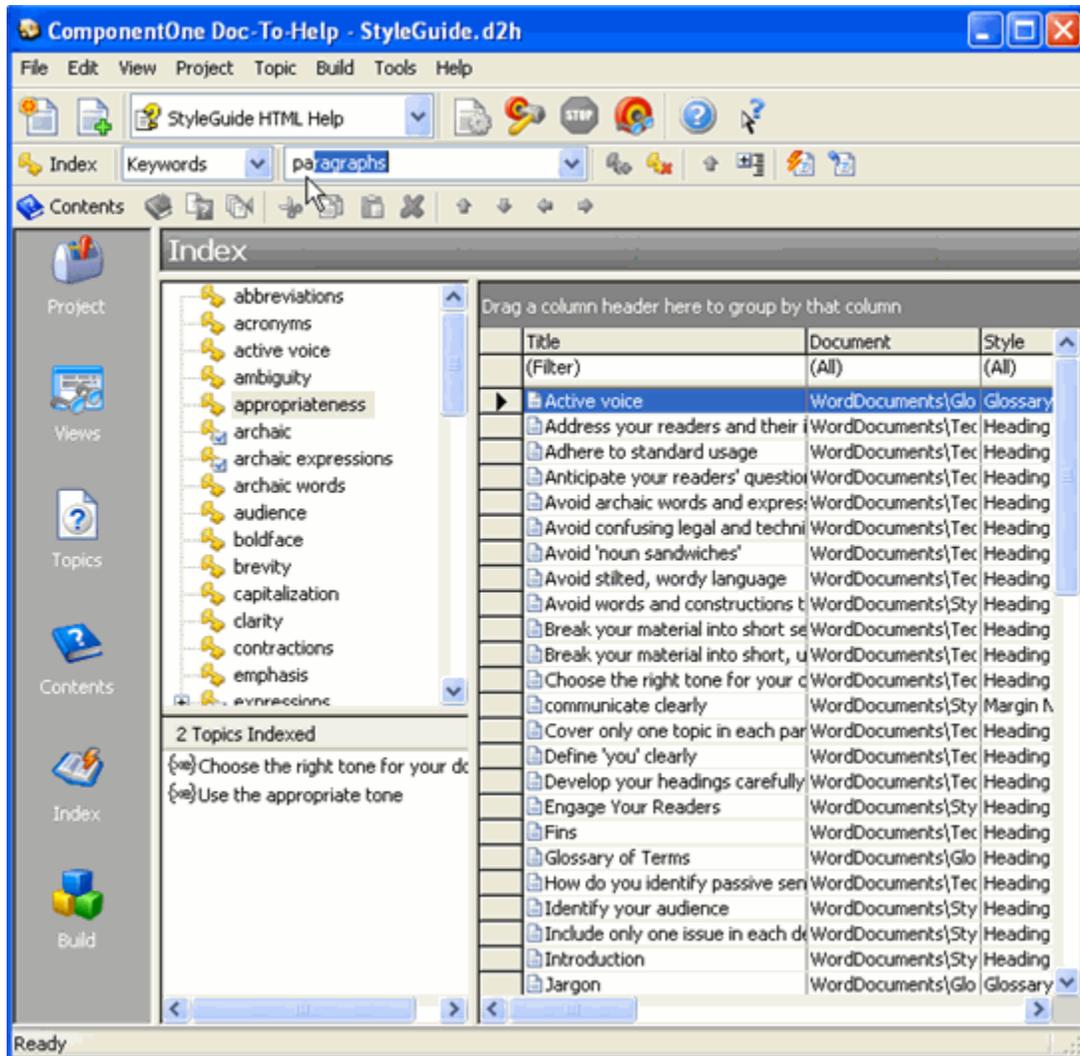
1. On the Icon bar, click **Index**.
2. Verify that the **Index Collections** button on the **Icon** toolbar is set to **Groups** or **Keywords** as required.
3. From the upper left pane, choose the group or keyword to be removed.
4. In the lower left pane, right-click the topic to be removed and choose **Unindex**.

Using the Index Elements Auto-Completion Feature

When you are trying to find a specific keyword or group in a large index, you can save time by using the auto-completion feature of the **Index Elements** list on the **Index** toolbar.

1. On the Icon bar, click **Index**.
2. Set the **Index Collections** button on the **Icon** toolbar to either **Keywords** or **Groups**.
3. Enter a letter in the **Index Elements** list box, and Doc-to-Help fills in the remaining letters of a word in the list of index keywords or groups.

For example, if you are looking for the keyword **paragraphs**, and you enter the letter **p** in the **Index Elements** list box, Doc-to-Help fills in the remaining letters of a word in the list of index keywords. If you enter another letter after the **p**, such as an **a**, Doc-to-Help finds the first word in the list beginning with **pa**.



- Once the remaining letters are filled in, click the arrow and select the topic from the drop-down list of keywords. If you press **Enter** instead of selecting the topic from the list, you will get an error telling you that the keyword already exists.

Customizing Help Windows

The Help targets use the default window, named main, to specify the size, location, appearance, and the user interface presented to the reader. You can change the default characteristics by editing the properties of this window in ComponentOne Doc-To-Help 2006 Pro.

You can also define secondary windows for displaying certain kinds of information. For example, if a topic lists a series of steps for carrying out a procedure, you may want to provide links to other topics with illustrations or additional instructions. By associating a secondary window with the character style that implements the link, you can display multiple topics simultaneously while giving the reader control over the level of detail and the layout of the Help windows.

By selecting the **Windows** item on the project editor tree list, you are presented with a list of the currently defined windows in the right pane. By choosing any one of the windows, the lower left pane displays the property settings for that window.

HTML Help Window Properties

Appearance Properties

[TriPaneWindow](#) When True, enables the standard tripane format for HTML Help.

Buttons Properties

[Jump1Caption](#) Contains the caption for the Jump1 button.

[Jump1URL](#) Contains the URL of the HTML file for the Jump1 button.

[Jump2Caption](#) Contains the caption for the Jump2 button.

[Jump2URL](#) Contains the URL of the HTML file for the Jump2 button.

[ShowBackButton](#) When selected, the Back button is included in the Help window.

[ShowForwardButton](#) When selected, the Forward button is included in the Help window.

[ShowHideShowButton](#) When selected, the Hide and Show buttons are included in the Help window.

[ShowHomeButton](#) When selected, the Home button is included in the Help window.

[ShowJump1Button](#) When selected, the Jump1 button is included in the Help window. Chose the caption and URL for this button with the [Jump1Caption](#) and [Jump1URL](#) properties.

[ShowJump2Button](#) When selected, the Jump2 button is included in the Help window. Chose the caption and URL for this button with the [Jump2Caption](#) and [Jump2URL](#) properties.

[ShowLocateButton](#) When selected, the Locate button is included in the Help window.

[ShowNextButton](#) When selected, the Next button is included in the Help window. The [BinaryTOC](#) property must be set to True for the navigation buttons to function.

[ShowOptionsButton](#) When selected, the Options button is included in the Help window.

[ShowPreviousButton](#) When selected, the Previous button is included in the Help window. The [BinaryTOC](#) property must be set to True for the navigation buttons to function.

[ShowPrintButton](#) When selected, the Print button is included in the Help window.

[ShowRefreshButton](#) When selected, the Refresh button is included in the Help window.

[ShowStopButton](#) When selected, the Stop button is included in the Help window.

Miscellaneous Properties

Caption	Represents the string to be displayed in the caption bar of the Help window.
Global	When True, \$global_ is prepended to the window name in generated files and <OBJECT> tags.
Name	Returns a string containing the name of a specified object (read-only).

Navigation Properties

AutoHideNavigationPane	Minimizes the HTML Help navigation pane when HTML Help is not the active window.
AutoSyncNavigationPane	Allows the heading or topic in the Navigation pane to automatically synchronize with whatever topic appears in the Topic pane.
HideNavigationPane	When True, the navigation pane for the Help file is hidden by default.
NavigationPaneWidth	Defines the width of the navigation pane of the Help file in pixels.

Position Properties

AlwaysOnTop	When True, the Help window remains on top of all other windows in the desktop.
Height	Determines the height of the Help window.
Left	Determines the location of the left edge of the Help window. In conjunction with the Top property, determines the position of the Help window on the screen.
Maximized	If True, the Help window is automatically maximized when displayed.
SaveUserPosition	When True, stores the size and position of the Help window in the registry when the user modifies it.
Top	Determines the location of the top edge of the Help window. In conjunction with the Left property, determines the position of the Help window on the screen.
Width	Determines the width of the Help window.

Tab Properties

FavoritesTab	When enabled, the favorites tab is added to the Help window.
SearchTab	Adds a search tab to the Help window.
SearchTabAdvanced	Adds additional functionality to the search tab. Note this property is only available when the SearchTab property is True.

NetHelp Window Properties

Miscellaneous Properties

Name	Returns a string containing the name of a specified object (read-only).
----------------------	---

Position Properties

Height	Determines the height of the Help window.
Left	Determines the location of the left edge of the Help window. In conjunction with the Top property, determines the position of the Help window on the screen.
Top	Determines the location of the top edge of the Help window. In conjunction with the Left property, determines the position of the Help window on the screen.
Width	Determines the width of the Help window.

WinHelp 4.0 Window Properties

Appearance Properties

NonScrollingColor	Displays the current color setting for the non-scrolling help window. To change the color, click the adjacent change button to open a standard color selection dialog.
TopicColor	Displays the current color setting for the topic Help window. To change the color, click the adjacent change button to open a standard color selection dialog box.

Buttons Properties

ShowBackButton	When selected, the Back button is included in the Help window.
ShowBrowseButtons	When selected, the Browse buttons are included in the Help window.
ShowContentsButton	When selected, the Contents button is included in the Help window.
ShowFindButton	When selected, the Find button is included in the Help window.
ShowHelpTopicsButton	When selected, the Help Topics button is included in the Help window.
ShowIndexButton	When selected, the Index button is included in the Help window.
ShowOptionsButton	When selected, the Options button is included in the Help window.
ShowPrintButton	When selected, the Print button is included in the Help window.

Miscellaneous Properties

Caption	Represents the string to be displayed in the caption bar of the Help window.
Name	Returns a string containing the name of a specified object (read-only).
WinHelpMacro	Specifies a macro to run when a Help file, topic, or secondary window is opened.

Position Properties

AdjustForScreenSize	Determines whether the Help window conforms to the resolution of the reader's display.
AlwaysOnTop	When True, the Help window remains on top of all other windows in the desktop.
AutoSizeHeight	When True, the height of secondary Help windows are automatically resized to fit the length of the current topic.
Height	Determines the height of the Help window.
Left	Determines the location of the left edge of the Help window. In conjunction with the Top property, determines the position of the Help window on the screen.
Maximized	If True, the Help window is automatically maximized when displayed.
Top	Determines the location of the top edge of the Help window. In conjunction with the Left property, determines the position of the Help window on the screen.
Width	Determines the width of the Help window.

Creating a New Help Window

To create a new window:

1. On the Icon bar, click **Project**.
2. Select the **Windows** item from the left pane.
3. Right-click anywhere in the right pane and choose **New** from the shortcut menu.

Note that the focus of your cursor has moved to the top of the window list pane.

4. Type the name of the new window and press **Enter**.

Note that the new window has been added to the list.

Alternatively, you can add a new window by clicking the **Project** menu, choosing **New**, then clicking **Window**.

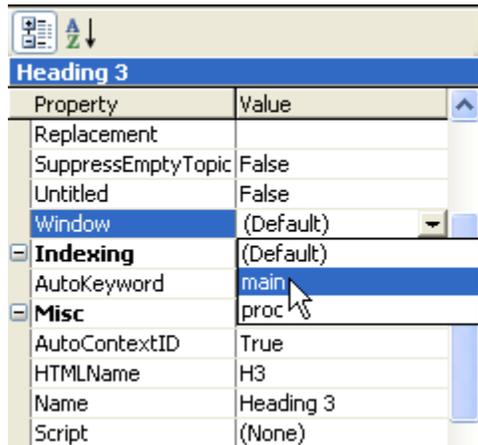
Assigning a Window to a Style

In order for a secondary window to be used, you must assign it to a style as follows:

1. On the Icon bar, click **Project**.
2. Select the **Paragraph Styles** or **Character Styles** item from the left pane.
3. In the right pane, select the name of the style to be edited.

Note that the property pane changes to show the properties of the selected style.

4. In the property pane, set the [Window](#) property to the window type you would like to be assigned to this style.



You can assign one window to multiple styles. You can also assign the *main* window to one or more styles to prevent the associated topics from being displayed in a secondary window.

Editing Help Window Properties

1. On the Icon bar, click **Project**.
2. Select the **Windows** item from the left pane.
3. Select the name of the window to be edited.

Note that the lower left pane changes to show the properties of the selected window.

4. Change the window properties as desired.

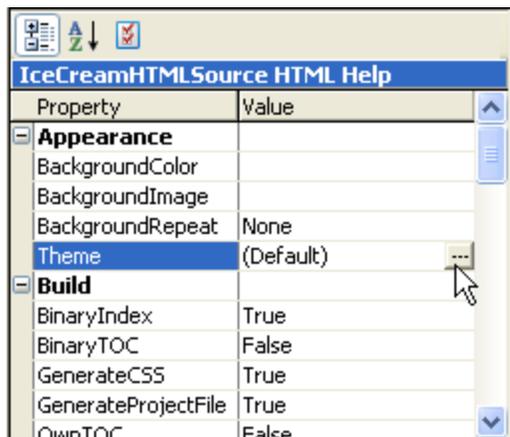
Note that your changes affect the selected window instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

Using Themes to Modify the Help Window

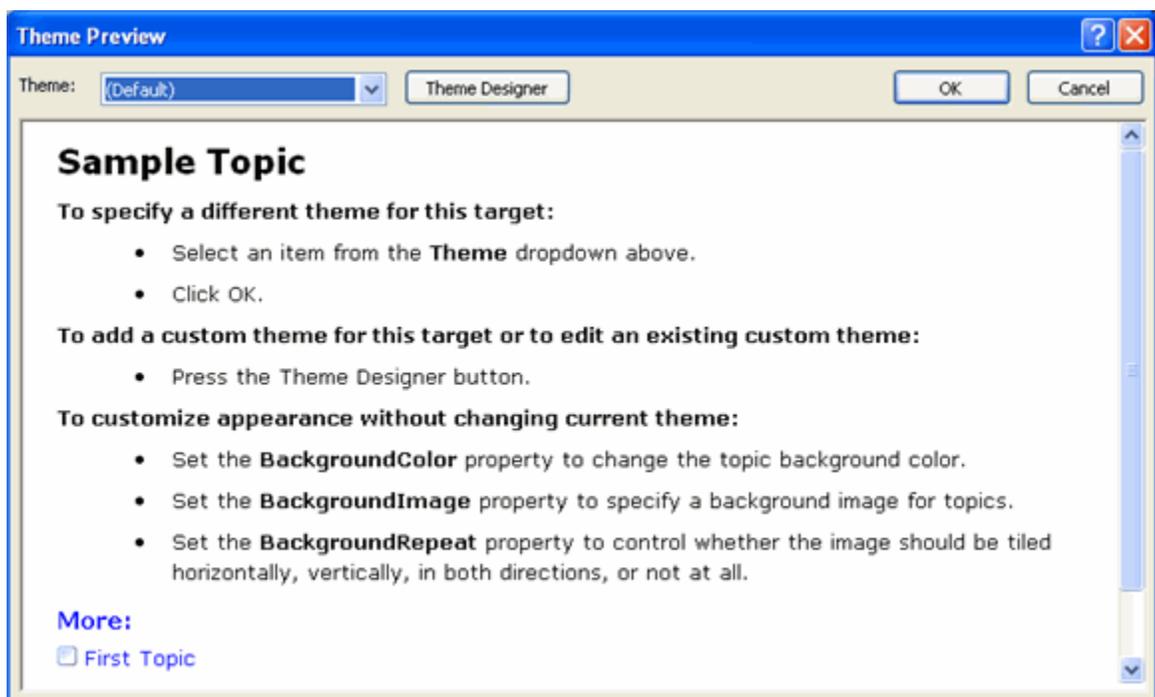
Doc-To-Help allows you to modify the style of your target windows for all HTML-based targets by using the [Theme](#) property. From the **Theme Preview** dialog, you can view each of the themes provided with Doc-To-Help and select the one that best fits your purposes. To customize your themes, see [Using the Theme Designer](#) (page 572).

To use the Theme property:

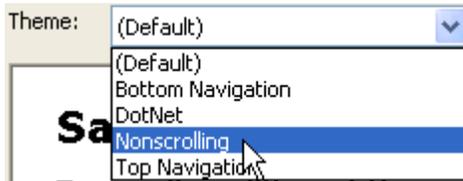
1. Select the **Projects** icon.
2. Select **Help Targets** from the left pane.
3. Select the Help target you wish to modify from the right pane This option is available for Help 2.0, HTML Help, JavaHelp and NetHelp.
4. From the property pane, select the [Theme](#) property and click the button in the box to the right of the property name.



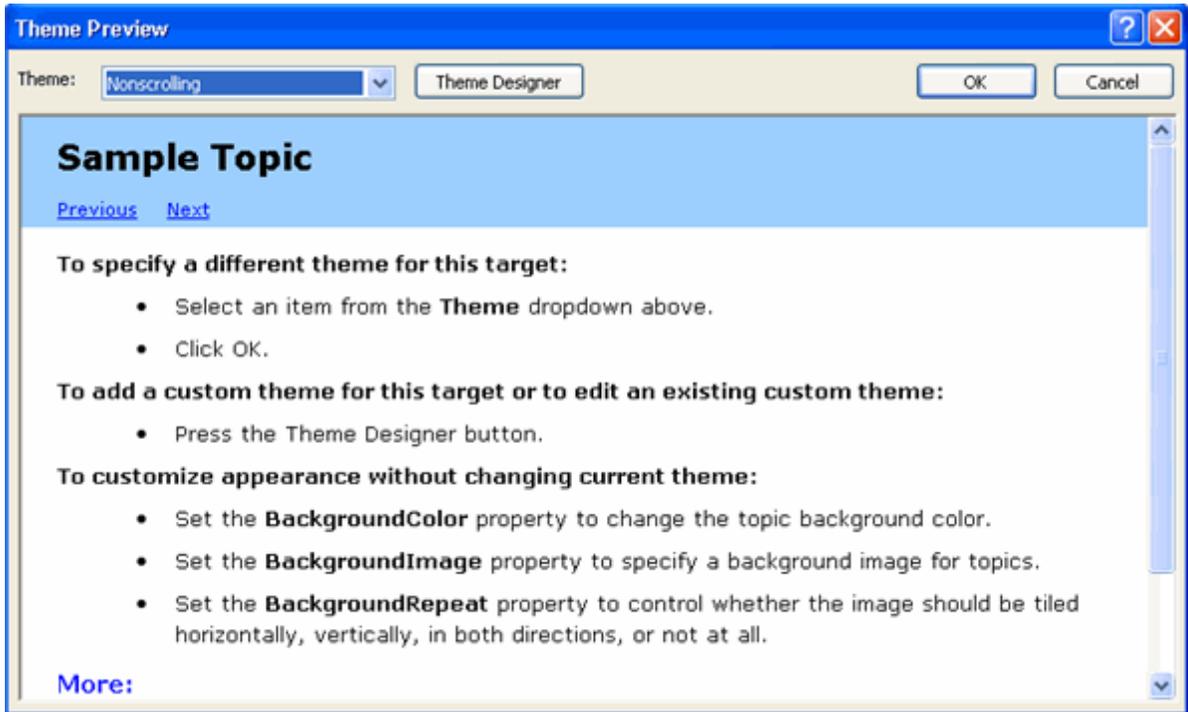
The **Theme Preview** dialog opens.



5. From the **Theme** drop-down list, select the theme style you wish to preview.



6. After you have viewed the themes and selected the one that suits your needs, click **OK** to accept the theme.



7. Build the Help target to reformat your Help window.

Note: To modify your themes, you can access the **Theme Designer** by clicking the **Theme Designer** button to the right of the theme drop-down. For more information, see [Using the Theme Designer](#) (page 572).

NetHelp Theme Descriptions

Default	The default NetHelp window includes a contents pane along the left side of the window and navigational hyperlinks above the Help text.
DotNet	The DotNet window includes a contents pane along the left side of the window and a colored non-scrolling area at the top of the Help pane. The non-scrolling area contains the topic heading and navigational buttons similar to the ones used in the MSDN Library in Microsoft Visual Studio .NET.
Nonscrolling	The Nonscrolling window includes a contents pane along the left side of the window and a colored non-scrolling area at the top of the Help pane. The non-scrolling area contains the topic heading and navigational hyperlinks.
TextOnly	The TextOnly window includes a contents pane along the left side of the window and navigational hyperlinks above the Help text. All navigational hyperlinks appear as text rather than buttons.
Tripane Classic	The Tripane HTML window includes a contents pane along the left side of the window and a navigation pane along the top of the window.

TripaneXP	The TripaneXP window includes a contents pane along the left side of the window and a navigation pane along the top of the window, but it has the XP look and feel.
------------------	---

HTML Help and Help 2.0 Theme Descriptions

Default	The default HTML Help window includes a table of contents pane along the left side of the window and the Help content in the right pane.
Bottom Navigation	The bottom navigation HTML Help window includes a table of contents pane along the left side of the window. The right hand Help content pane includes previous/next icons along the bottom of the pane.
DotNet	The DotNet HTML Help window includes a contents pane along the left side of the window and a colored non-scrolling area at the top of the Help pane. The non-scrolling area contains the topic heading and navigational hyperlinks similar to the ones used in the MSDN Library in Microsoft Visual Studio .NET.
Nonscrolling	The Nonscrolling HTML Help window includes a contents pane along the left side of the window and a colored non-scrolling area at the top of the Help pane. The non-scrolling area contains the topic heading and navigational hyperlinks.
Top Navigation	The top navigation HTML Help window includes a table of contents pane along the left side of the window. The right hand Help content pane includes previous/next icons along the top of the pane.

JavaHelp Theme Descriptions

Default	The default JavaHelp window includes navigational hyperlinks above the Help text.
----------------	---

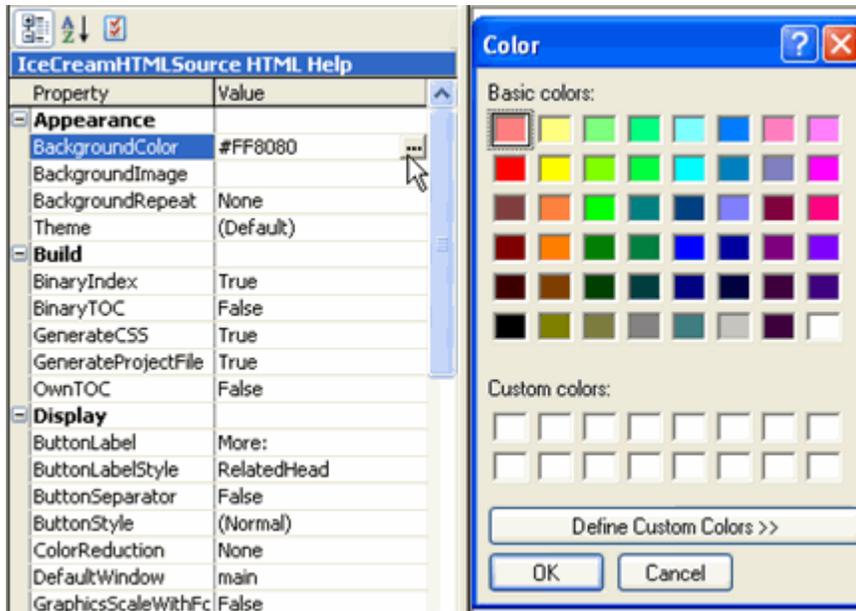
Modifying the Background of the Window Pane

By modifying the [BackgroundColor](#), [BackgroundImage](#) and [BackgroundRepeat](#) properties, you can customize the background of your Help contents pane for NetHelp, Help 2.0, JavaHelp and HTML Help. The [BackgroundColor](#) property allows you to define the background color of the pane, while using a combination of the [BackgroundImage](#) and [BackgroundRepeat](#) properties allows you to select a graphic and position it as part of your background.

Changing the Background Color

To change the background color of your Help contents:

1. Select the **Projects** icon.
2. Select **Help Targets** from the left pane.
3. Select the Help target you wish to modify from the right pane (not available in Manual or WinHelp targets).
4. From the property pane, select the [BackgroundColor](#) property and click the button in the box to the right of the property name.



The standard Windows **Color** dialog opens.

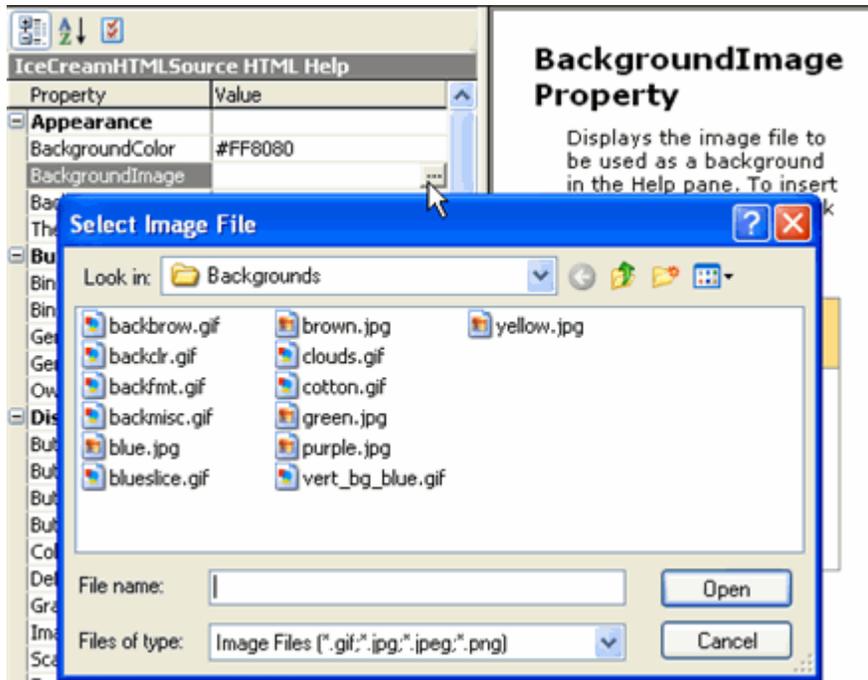
5. Select a color from the dialog and click **OK**.
6. Build the Help target to reformat your Help contents pane with the new background color.

Note: To preview your color selection before building your Help target, you can open the **Theme Preview** dialog box. For more information, see [Using Themes to Modify the Help Window](#) (page 357).

Inserting and Positioning a Background Image

To insert and position a background image in your Help contents pane:

1. Select the **Projects** icon.
2. Select **Help Targets** from the left pane.
3. Select the Help target you wish to modify from the right pane (not available in Manual or WinHelp targets).
4. From the property pane, select the [BackgroundImage](#) property and click the button in the box to the right of the property name.



The **Select Image File** dialog opens.

Note, that the Select Image File dialog opens by default to C:\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds.

This directory contains the default background images included with Doc-To-Help. Any customized background images you plan to use should also be placed in this directory prior to inserting them into your Help project. **Doc-To-Help retrieves background images from this directory, images contained in alternate directories will not be used by Doc-To-Help during the build process.**

5. Select a background image from the dialog and click **OK**.

By default, the image will be placed in the upper left corner of the Help contents pane. Your options for image positioning are:

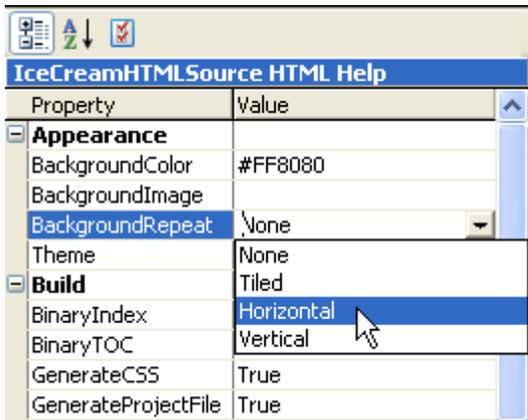
None The background image is placed in the upper left corner of the Help contents pane.

Tiled The background image is tiled, covering the entire area of the Help contents pane.

Horizontal The background image is repeated horizontally along the top edge of the Help contents pane.

Vertical The background image is repeated vertically along the left edge of the Help contents pane.

6. Select your background image preference from the drop-down to the right of the [BackgroundRepeat](#) property in the Help target property pane.



7. Build the Help target to reformat your Help contents pane with the new background image.

Note: To preview your background image selection before building your Help target, you can open the **Theme Preview** dialog box. For more information, see [Using Themes to Modify the Help Window](#) (page 357).

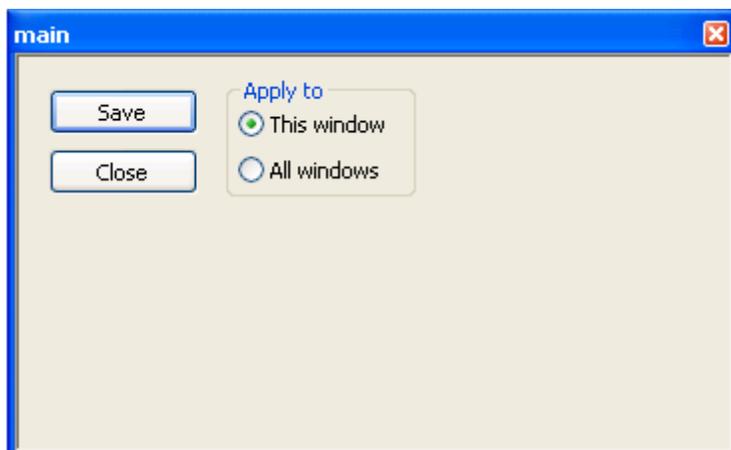
Modifying the Window Size Using the Size Tool

Doc-To-Help provides you with an easy-to-use utility that allows you to modify the size and position of your Help windows.

1. On the Icon bar, click **Project**.
2. Select the **Windows** item from the left pane.
3. In the right pane, select and then right-click the window name to be edited and choose **Size Tool** from the shortcut menu.

The size tool dialog box opens.

4. Drag any side or corner of the window to adjust it to the size you want.



5. Drag the window to the desired position on the screen and click **OK**.

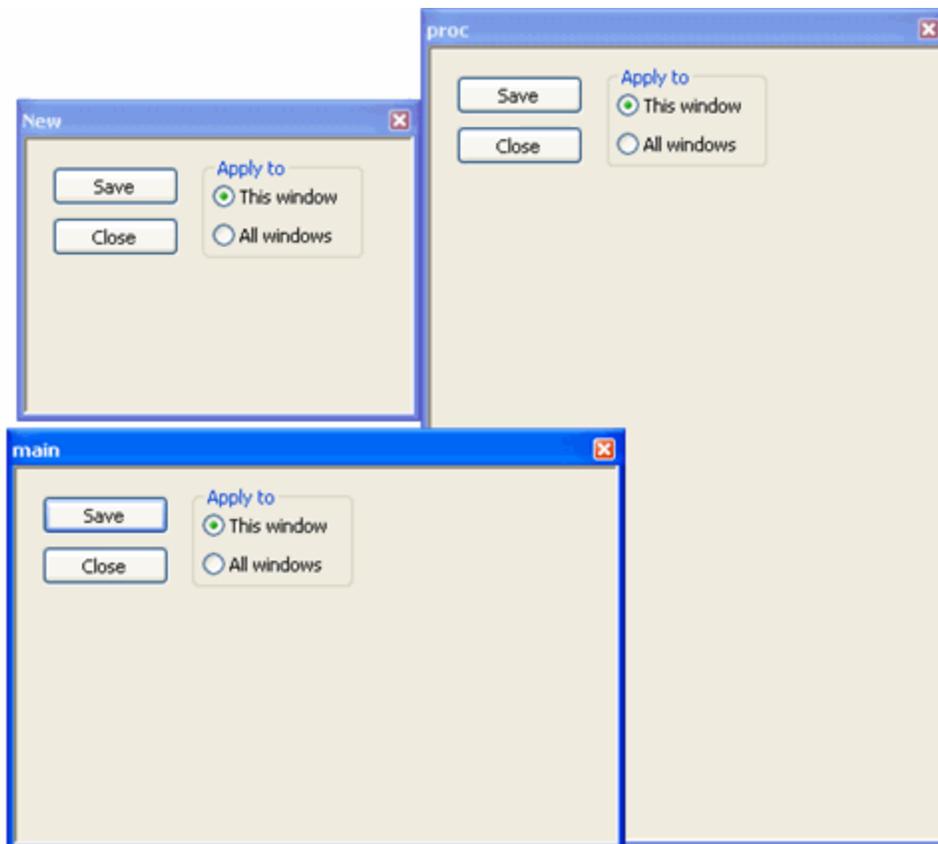
Note: A practical method of using the tool is to open your compiled Help file window and reposition and adjust the size of that window as needed. Then open the **Size Tool** and stretch the size tool dialog box to cover your Help file window and click **OK**. In this way, you have set the new window properties as the default for that window.

Modifying Multiple Windows

Doc-To-Help allows you to modify all windows in your Help system at one time. By opening one Window Size Tool for each defined window you wish to modify, you can reposition them until you are satisfied with the relative locations, then save them all simultaneously.

1. On the Icon bar, click **Project**.
2. Select the **Windows** item from the left pane.
3. In the right pane, select and then right-click the window name to be edited and choose **Size Tool** from the shortcut menu.
4. Drag any side or corner of the window to adjust it to the size you want.
5. Repeat this process for each window you wish to resize leaving all windows open.
6. Click **All Windows** in any of the size tool windows and click **Save**.

All open size tools close and the size and location settings for all windows is saved.



Tips and Techniques for Word

This chapter shows how you can enhance your Help projects with graphics, macros and Internet links. It also describes the printing, reporting, and project organizer features available through ComponentOne Doc-To-Help 2006.

Working with Graphics in Word

When preparing your source documents for Doc-To-Help, you can include static graphics just as you would in a "normal" document. You do not need to use any nonstandard commands or specify target-specific escape sequences.

However, some of the graphics features introduced in older versions of Word are not suitable for Help authoring, and you should avoid using them. Here are some guidelines to follow when inserting pictures into Word source documents:

- Don't use floating pictures. Since online Help systems don't have the concept of Z-order, the placement of the picture may not be what you expect.
- Instead of pasting a picture into a document directly from the Clipboard, save the picture to a file and insert it as a linked object. That is, use the **Link to file** option in the **Insert Picture** dialog box. This will save you time if the pictures are likely to change.
- If a picture is too large, you can crop it or scale it in Word, and Doc-To-Help applies the appropriate transformations to the image file it generates during compilation. However, if you crop or scale a linked picture, use the **Insert and Link** option in the **Insert Picture** dialog box. Otherwise, Doc-To-Help reads the picture file from disk as is.

Inserting Graphics Files

1. On the **Insert** menu in Microsoft Word, point to **Picture**, then click **From File**. The **Insert Picture** dialog box opens.
2. Specify a folder and file name, then choose **Insert**, **Link to file** or **Insert and Link**.

Scaling Inserted Graphics

1. Select the graphic you inserted, right-click, and select **Format Picture** from the shortcut menu.

Note: To scale the graphic using the features contained within Microsoft Word, you must have "inserted", not "linked" the graphic.

2. Select the **Size** tab.
3. In the Scale area, change **Height** and **Width** as desired.
4. Click **OK**.
5. Save and close the Word document.
6. In the Project Editor, click the **Project** icon.
7. Select **Help Targets** from the left pane.
8. Select the correct Help Target from the right pane. In the property pane, verify that the [ScaleGraphics](#) property is set to **True**.

Note: The [ScaleGraphics](#) property allows you to determine on a target-by-target basis if the Word scaling will be applied.

9. Click the **Make Target** button to compile the Help target.

Inserting Objects to Create Special Effects in Word Documents

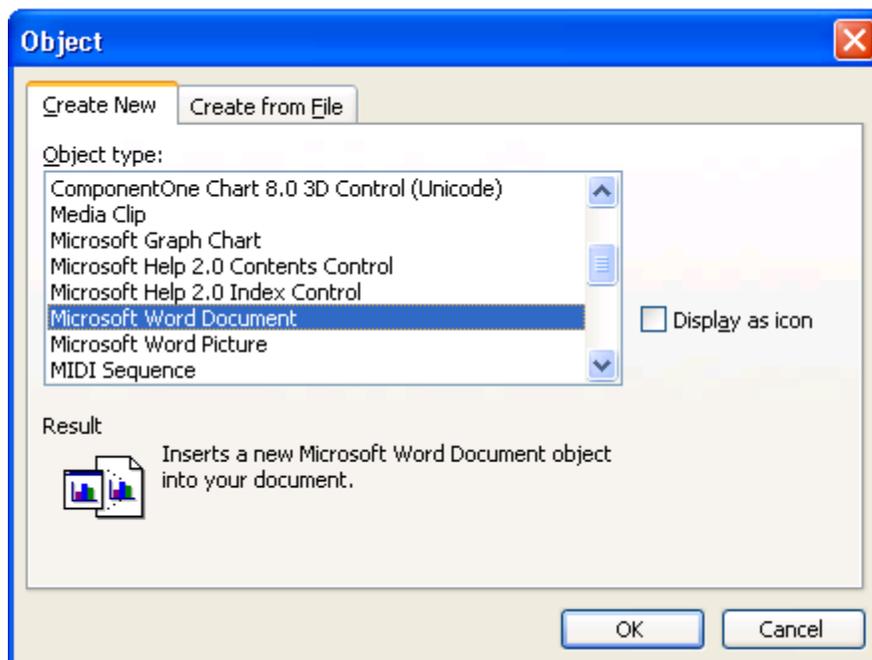
Some of Microsoft Word's drawing and text layout features are ideal for producing hard copy output, but ill-suited for creating online Help. AutoShapes, text boxes, and WordArt effects can only reside in the drawing layer; you cannot insert them as inline elements. Since you have no control over the placement of floating objects in the RTF files passed to the WinHelp compiler, you really can't take advantage of these features.

Although the WinHelp compiler supports Word tables, it does not support table borders or shaded cell backgrounds. If you want to see these table features in a Help file, you have to create a bitmap that looks like a table or take a screen shot of a table in Word.

Fortunately, you can circumvent these restrictions on tables by embedding another Word document within your source document as follows:

1. In your source document, click where you want to insert a table or drawing object.
2. On the **Insert** menu, click **Object**.

The following dialog box opens.



3. In the **Object type** list, select **Microsoft Word Document**.
4. If checked, clear the **Display as icon** check box, then click **OK** to create a new document.
5. Insert the table or drawing object as you would in a "normal" Word document.
6. When you are finished editing, click **Close & Return to** on the **File** menu to return to the original document.

When Doc-To-Help compiles the source document, it treats the embedded Word document as a graphic and creates an appropriate image file for the current Help target.

Example

The following table was inserted into the source document as an embedded Word document.

Wine Tasting Chart	
intensity	delicate powerful
dry or sweet	bone drydessert
body	light bodyvery full body
acidity	soft, gentle very crisp
tannin	none heavy tannins
oak	none heavy oak
complexity	direct very complex

Like tables, Doc-To-Help cannot translate floating pictures into the final Help system. Converting a floating picture to an Inline shape will ensure that this image is included in the final document.

Converting Floating Pictures to Inline Shapes

1. Select the picture to be converted.

Note that floating pictures are only visible in online layout or page layout view.

2. Right-click the picture, then choose **Format Picture** from the shortcut menu.
3. On the **Layout** tab, select the **In line with text** wrapping style, then click **OK**.

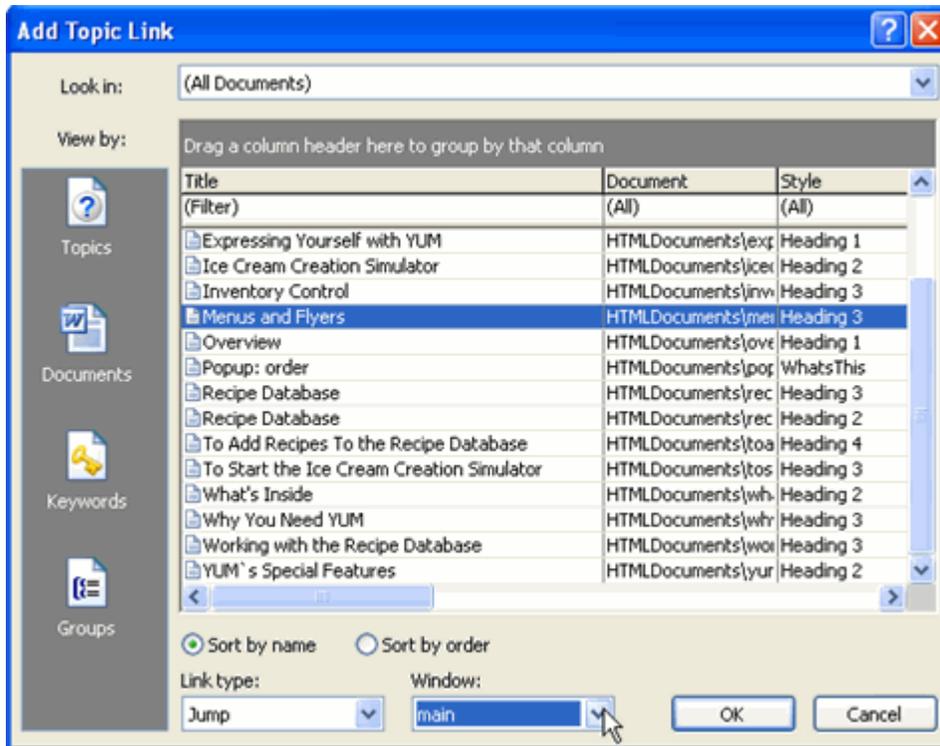
Creating Hot Spots With Graphics

Typically, you implement links and hot spots as text formatted with an active character style. Doc-To-Help derives a link tag from the text during compilation, this link tag links this text to the destination hot spot. Depending upon the style's link type, Doc-To-Help looks for a matching topic, index keyword, or group name. You can also override or cancel the default link tag by writing a script and associating it with the character style. Please review the section [Links and Hot Spots](#) (page 309) if you are not already familiar with the text to link tag process.

You can also implement links and hot spots using graphics instead of text. As with textual links, Doc-To-Help uses active character styles to determine both the link type and the destination, and you can override or cancel the default behavior with scripts. Doc-To-Help still needs some text from which it can determine the destination and create the link tag, however. Since Microsoft Word does not provide an intrinsic way to attach an arbitrary string to a picture, Doc-To-Help provides a command that does just that.

To create a graphical hot spot:

1. Insert the picture you want to use as the hot spot.
For specific instructions, see [Inserting Graphics Files](#) (page 365).
2. Select the hot spot graphic then click the [Add Topic Link](#) dialog box button.
3. In the **Add Topic Link** dialog box, choose the topic you wish to associate with your hot spot.
4. From the **Link Type** drop-down, select the **Jump** or **Popup** option.
5. From the **Window** drop-down, select the window type where you wish to display your topic and click **OK**.

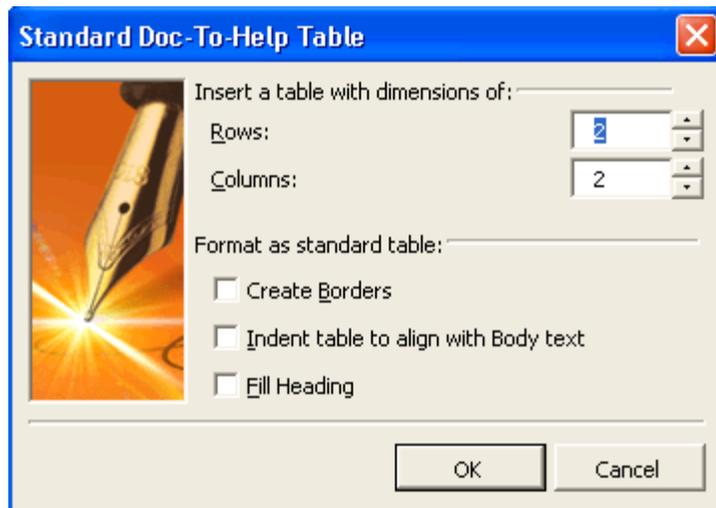


For more information on customizing windows, see [Customizing Help Windows](#) (page 353).

Inserting Standard Tables with Doc-To-Help

You can use the **Standard Table** command to insert basic tables quickly and easily into your Word source documents. These tables can be aligned with your body text and formatted by selecting options contained in the Standard Doc-To-Help Table dialog. To access the Standard Table dialog, click the Standard Table button . The command behaves differently depending on whether you have selected text.

If no text is selected, you are presented with a dialog that allows you to specify the number of rows and columns in your table. The table also allows you to determine if your table has borders, whether it is aligned with your body text and whether the heading has a fill.



If text is selected, Doc-To-Help uses the convert text to table features contained within Word to format the table. If the text contains tabs, each tabbed section is placed into an individual table cell. If the selected text consists of more than

one paragraph, each paragraph is placed into a new row in the table. When you click the **Standard Table** button with text selected you are presented with the following dialog.



As you can see, the number of rows and columns is automatically selected for you and you are left with the standard “formatting” options.

By selecting the **Create Borders** checkbox, your table is formatted with a double line border as show below.

Table with **Create Borders** selected.

By selecting the **Indent table to align with Body text** checkbox, your table is aligned with the body text in your document as shown below.

Table with **Indent table to align with Body text** selected.

By selecting the **Fill Heading** checkbox, your table is formatted with the uppermost set of table cells containing a gray fill as show below.

Table with **Fill Heading** selected.

Alternatively, you can perform the same steps with the **Table | Standard Doc-To-Help Table** menu command.

Linking to Internet Sites

You can use Microsoft Word’s hyperlink feature to enhance your online Help targets with links to Web pages and downloadable files. Word supports Internet links via HYPERLINK field codes, which Doc-To-Help recognizes during source document compilation.

When Doc-To-Help builds an online Help system, it automatically converts HYPERLINK field codes into instructions that are appropriate for the active Help target. For WinHelp, it generates `ExecFile` macros; for NetHelp and HTML Help, it generates `<A HREF>` tags.

When compiling WinHelp, Doc-To-Help renders hyperlinks as ordinary topic jumps. On most systems, WinHelp topic jumps appear as single-underlined green text. Clicking a hyperlink always opens the default Internet browser in a separate application window; you cannot display a Web page within the confines of the main WinHelp window or a secondary window.

In HTML and HTML Help, hyperlinks are rendered according to your reader's browser settings. Unlike WinHelp, you can control whether hyperlinked URL's display within the same window or a separate application window. By default, they are displayed within the same window (or frame).

Entering Hyperlink Text

The simplest way to insert a hyperlink is to type a URL and let Word convert it to a HYPERLINK field for you. The following examples are all valid hyperlinks:

<http://www.componentone.com/>

www.componentone.com

<ftp://ftp.componentone.com/>

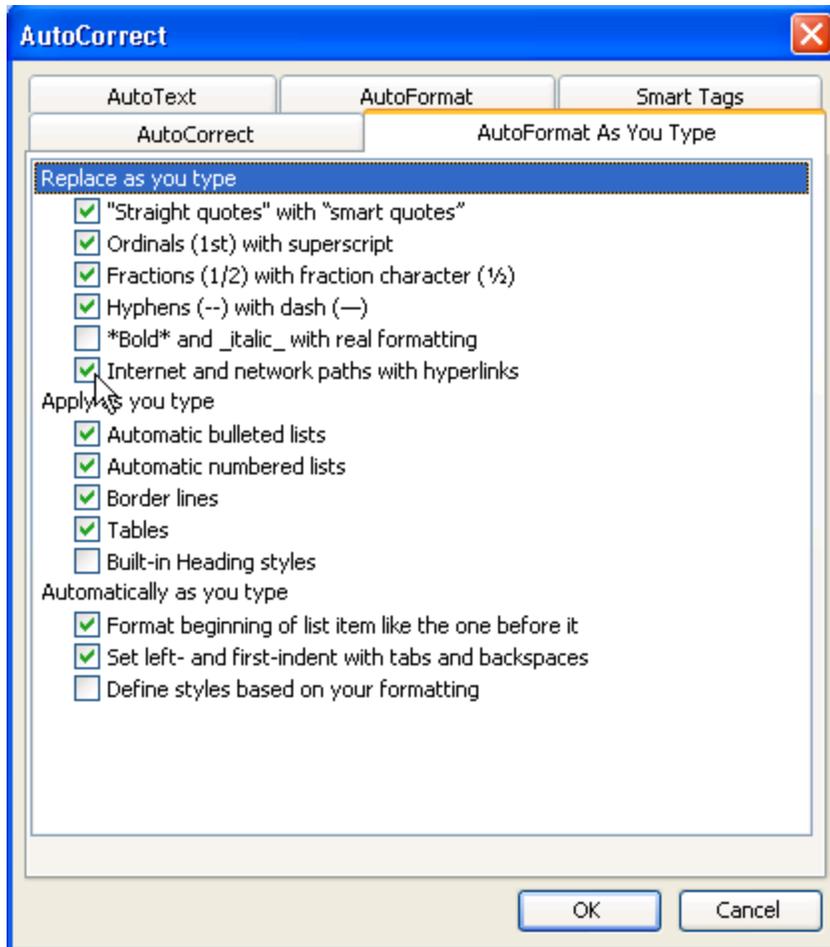
To enter hyperlink text:

1. Type the URL as you would normal text. You can generally omit the `http://` and `ftp://` protocol prefixes.
2. Terminate the URL with a space, tab, punctuation mark, or the ENTER key. Word automatically applies the *Hyperlink* style to the text you typed and converts it into a field code having the following format:

```
{ HYPERLI NK your -URL }
```

If Word does not convert the text as described here, the option has been turned off. You can turn it back on as follows:

1. On the **Tools** menu in Microsoft Word, click **AutoCorrect Options**.
The **AutoCorrect** dialog box opens.
2. On the **AutoFormat As You Type** tab, check the box labeled **Internet and network paths with hyperlinks**, then click **OK**.



Inserting Hyperlink Fields

To create a hyperlink that displays in the current window:

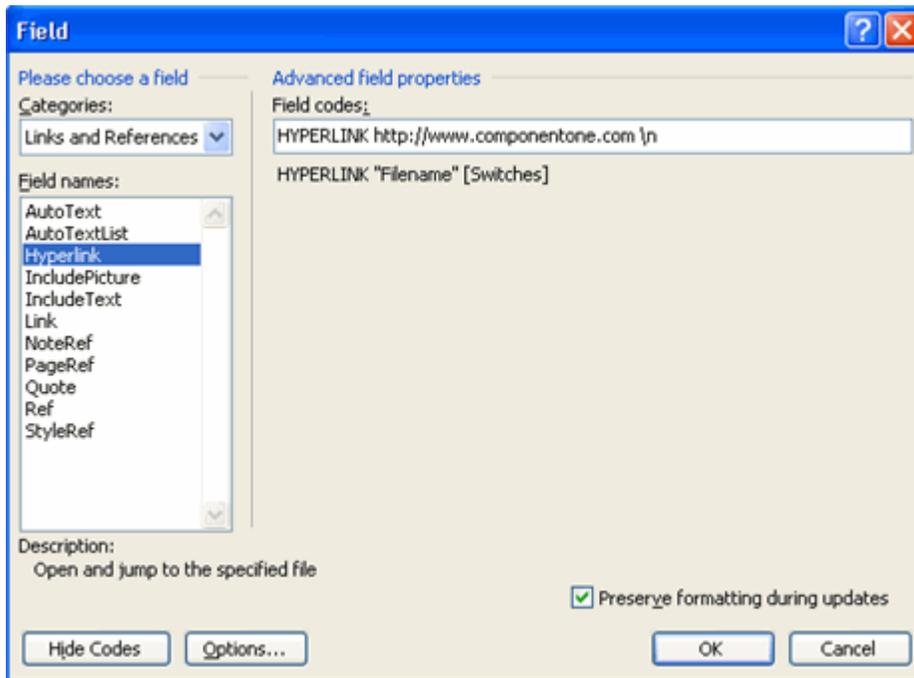
1. On the **Insert** menu in Microsoft Word, click **Hyperlink**.
2. The **Insert Hyperlink** dialog box opens.
3. Enter the text to display for your hyperlink in the **Text to display** box.
4. Enter or locate the URL in the **Type the file or Web page name** box.
5. You can enter the address of a Web page (HTTP protocol) or a downloadable file (FTP protocol). You can locate a specific file or Web site by clicking the File or Web Page buttons on the right. In addition you can add any valid protocol that Word recognizes (mailto, nntp, help, etc).
6. When you are finished, click **OK**.

To create a hyperlink that opens a new application window:

1. On the **Insert** menu in Microsoft Word, click **Field**.
The **Field** dialog box opens.
2. In the **Categories** list, select **Links and References**.
3. In the **Field names** list, select **Hyperlink**.
4. Click the **Field Codes** button and in the **Field codes** box, enter the URL and the \n switch as shown in the following figure.

5. Click **OK**.

Make sure that the URL is separated from the field code and the switch with a space and that you include the HTTP protocol identifier at the beginning of the link.



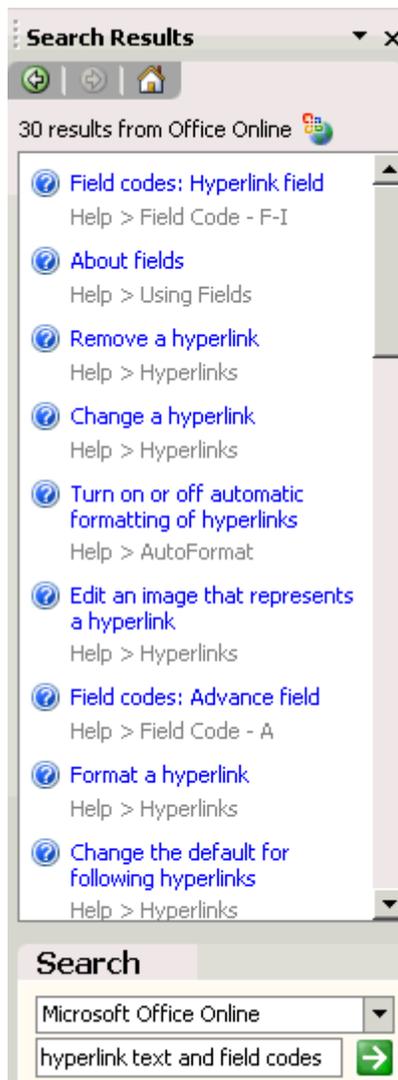
Note: The \n switch has no effect in WinHelp builds.

Editing Hyperlink Fields

To change the URL or optional argument for a hyperlink field:

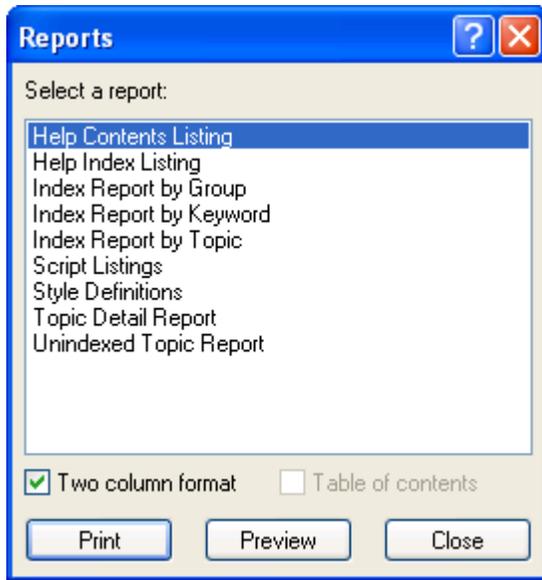
1. Right-click the hyperlink text, point to **Hyperlink**, and then click **Edit Hyperlink on the submenu**. The **Edit Hyperlink** dialog box opens.
2. Specify new display text in the **Text to display** box, if desired.
3. Specify a new URL in the **Type the file or Web page name** box, if desired.
4. When you are finished, click **OK**.

5. See Word's Help for more information on hyperlink text and field codes.



Generating Project Reports

Doc-To-Help offers a variety of project reports and printable views to help you organize your work and keep track of topics and index elements. You can either produce hard copy or preview the output on screen.



One way to get information on your current project is to print out the current view in the Doc-To-Help visual environment. This type of report prints the items in the panes of the current view.

Printing the Current View

1. On the **File** menu, click **Print**. Doc-To-Help opens a standard print dialog box.
2. Change the print job settings, if desired.

The output of the **Print** command reflects the either the right pane or the properties pane, depending on which has focus.

To change the paper size, source, or orientation, click **Print Setup** on the **File** menu.

Previewing Pages Before Printing

1. On the **File** menu, click **Print Preview**. Doc-To-Help replaces its main window with a preview window that displays individual pages as they will be printed.
2. Use the row of named buttons at the top of the preview window to navigate, zoom, or print:

Print document	Open the print dialog box to start a print job.
Zoom In	Enlarge the preview image.
Zoom Out	Shrink the preview image.
Actual Size	Sets the preview image to 100%.
Fit in window	Shrinks the preview image to fit in the window so you do not have to scroll.
Fit width	Shrinks the preview image to fit in the window so you do not have to scroll.
Go to first page	Display the first printed page.
Go to previous page	Display the previous printed page.
Go to next page	Display the next printed page.
Go to last page	Display the last printed page.

Whenever you open the preview window, Doc-To-Help rescales the preview image so that an entire page fits within it. If the resulting text is too small, you can resize or maximize the main window or click the **Zoom In** button.

Selecting a Report for Printing

1. On the **View** menu, click **Reports**.
2. Select the report you want to see. For a description of each, see [Available Reports](#) (page 375).



3. Depending upon which report is selected, it may be available in **Two column format** with an optional **Table of contents** at the end of the print job. If available, select or clear the corresponding check boxes as desired.
4. Click **Print** to send the report to the default printer, or click **Preview** to view the report within the Doc-To-Help application window. Click **Close** to dismiss the dialog box without printing or previewing.

Available Reports

This version of Doc-To-Help includes the following reports:

Help Contents Listing

This report depicts an online Help contents with all books expanded and all topics visible. It supports the two-column option.

Help Index Listing

This report depicts an online Help index with all secondary keywords visible. It supports the two-column option.

Index Report by Group

For each group name in the project, this report shows a list of associated topics, if any. Groups are listed in alphabetical order. This report supports both the two columns and table of contents options.

Index Report by Keyword

For each index keyword in the project, this report shows a list of associated topics, if any. Keywords are listed by full pathname in alphabetical order. This report supports both the two columns and table of contents options.

Index Report by Topic

For each topic in the project, this report shows a list of associated keywords and groups, if any. Topics are listed in alphabetical order by title. This report supports both the two columns and table of contents options.

Script Listings

This report lists the code for each script defined in the project. Scripts are listed in alphabetical order. This report supports the table of contents option.

Style Definitions

This report lists the properties of each style defined in the project. Styles are listed in alphabetical order. This report supports the table of contents option.

Topic Detail Report

For each topic in the project, this report lists its properties. Topics are listed in alphabetical order by title, and each topic starts on a new page. This report supports the table of contents option.

Unindexed Topic Report

This report lists each topic that is not associated with an index element (keyword or group). Topics are listed alphabetically by title, along with the document and style from which the topic was derived.

Managing Project Files

Doc-To-Help provides an **Organizer** dialog box similar to the one in Microsoft Word, which you can use to transfer styles, scripts, window definitions, and index elements from one project to another. You can also use the **Organizer** dialog box to delete these items from a project.

Copying Data Between Projects

1. On the **Project** menu, click **Organizer**. The **Organizer** dialog box opens.

Initially, all elements in the active project are displayed in the list on the left side of the dialog box. Use the **Show elements** list to view a specific type:



Paragraph and character style behaviors



Group names



Index keywords



Script code modules



HTML Help window definitions



WinHelp window definitions



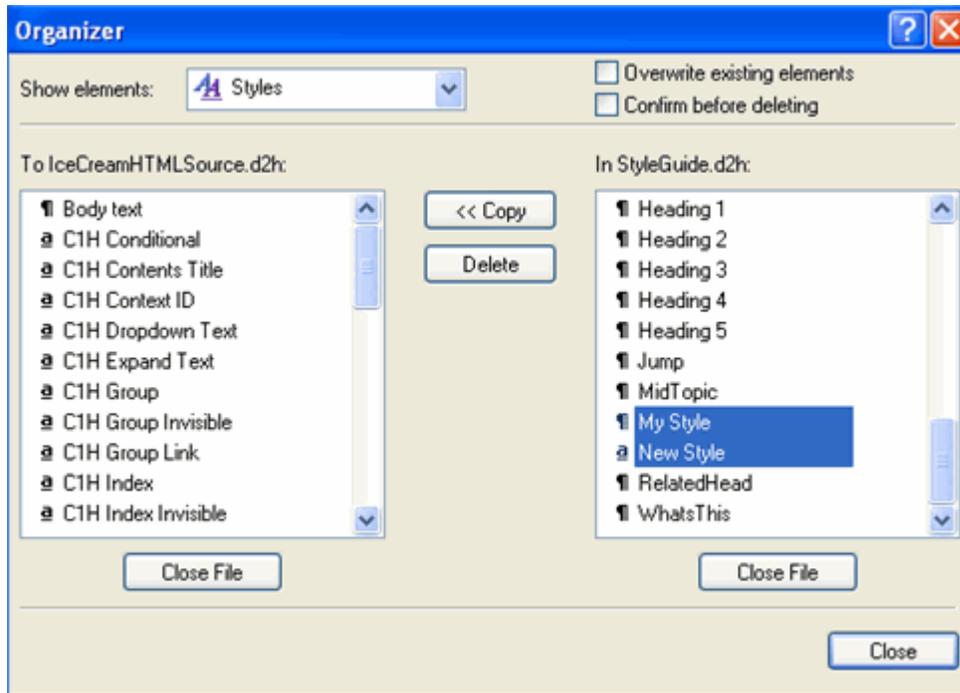
Help target properties



Topic type definitions

2. Click **Open File** to display a file selection dialog box. Specify a folder and file name for an existing Doc-To-Help project, then click **Open**. The list on the right side of the **Organizer** dialog box now contains elements from the specified project. Initially, none of the elements are selected, and the **Copy** button is unavailable.
3. By default, like-named elements are not copied from one project to another. If desired, check the **Overwrite existing elements** box to force the **Organizer** dialog box to redefine existing elements in the destination project.
4. Select one or more elements in the list on the right, then click **Copy**. When you are finished, click the **Close** button at the lower right corner of the dialog box.

The following figure shows the **Organizer** dialog box ready to copy two styles from one project (StyleGuide.d2h) to a new project file (CondText.d2h).



Deleting Individual Project Elements

To delete an individual style, script, topic type or window outside of the Organizer:

1. Click the **Projects** icon.
2. In the left pane of the project editor, choose the appropriate item.
3. In the right pane, right-click the element to be deleted, then click **Delete**.

To delete an individual index keyword or group:

1. Click the **Index** icon.
2. Ensure that the appropriate category is displayed by choosing **Groups** or **Keywords** from the **Index** toolbar.
3. Right-click the element to be deleted, then click **Delete**.

If you delete a script or window that is assigned to one or more styles, Doc-To-Help also detaches the deleted element from those styles.

You can also use the **Organizer** dialog box to delete individual project elements:

1. On the **Project** menu, click **Organizer**.
The **Organizer** dialog box opens.
2. Initially, all elements in the active project are displayed in the list on the left side of the dialog box. Use the **Show elements** list to view a specific type (groups, keywords, scripts, styles, or windows).
3. Select the **Confirm before deleting** check box if you want the **Organizer** dialog box to prompt you before deleting individual elements.
4. Select one or more elements in the list on the left, then click **Delete**. When you are finished, click the **Close** button in the lower right corner of the dialog box.

Deleting Help Topics

Doc-To-Help derives topics from text formatted with heading styles in your source documents. After removing any of the headings from your source documents, Doc-To-Help deletes the existing topic when the file is compiled. Additionally, if you modify a heading in your source documents, Doc-To-Help deletes the old topic from the database and adds the modified heading. In both cases, after compiling the Help project, the modified topic is marked as deleted in the topics pane of the project editor as shown below.

Title	Document	Order
(Filter)	(All)	(All)
Include only one issue in each designated paragraph	WordDocuments\StyleGuide.	19
Introduction	WordDocuments\StyleGuide.	1
Jargon	WordDocuments\Glossary.doc	2
Organize to meet your readers' needs	WordDocuments\StyleGuide.	-1
Place words carefully	WordDocuments\StyleGuide.	-1
Pose questions to help readers find information	WordDocuments\Techniques	4
Provide a well-organized table of contents	WordDocuments\Techniques	3
Style Guide	WordDocuments\StyleGuide.	0
Take care to eliminate unclear phrasing	WordDocuments\Techniques	14
Techniques for Better Writing	WordDocuments\Techniques	0

Removing Deleted Topics

To remove a topic that has been marked as “deleted” from the project editor:

In the right pane of the **Topics** view, select and right-click the deleted topic (marked with an X) to be removed and select **Delete** from the shortcut menu.

Alternatively, if you want to remove all deleted topics, you can right-click any deleted topic and select **Purge** from the shortcut menu.

Using HTML Help Object Tags in Word

By clicking the **Insert HTML Help ActiveX Control** button  on the **Doc-To-Help** toolbar, Doc-To-Help gives you access to the HTML Help ActiveX control contained within the HTML Help Workshop. Key features of the HTML Help ActiveX control include: an expanding table of contents, keyword search, shortcuts, and pop-up help topics.

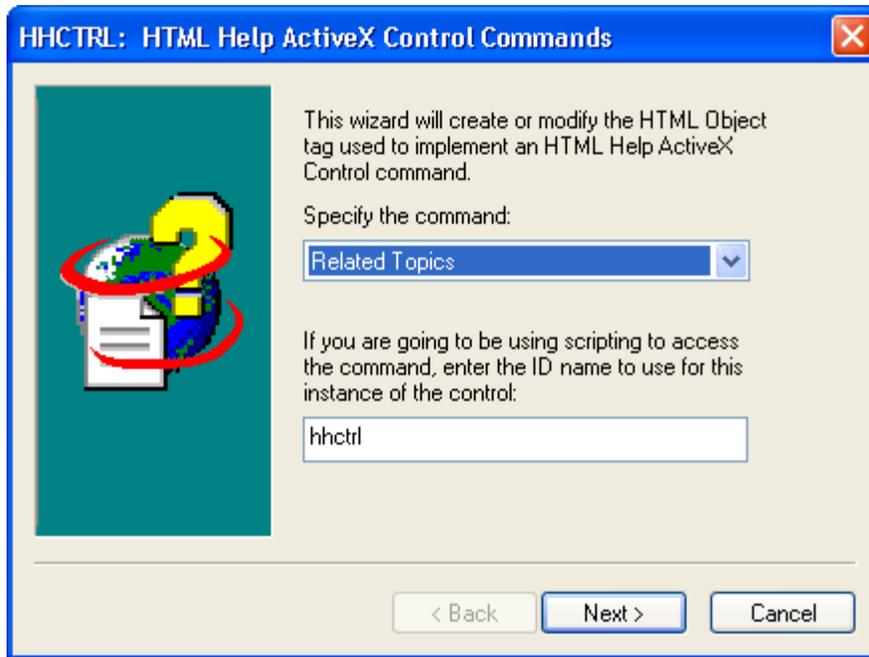
For HTML Help ActiveX Control documentation, visit the Microsoft HTML Help Workshop Web site.

Inserting an HTML Help Object Tag

Inserting the HTML Help Object Tag into a document involves adding a tag at the location you want the functionality to appear. The HTML Help ActiveX Control Wizard enables you to insert the object tag for the control.

1. On the icon bar, select **Project**.
2. In the left pane, click **Documents**.
3. In the right pane, select then right-click the document you wish to edit and choose **Open** from the shortcut menu.
4. Position the cursor at the location where you want the ActiveX control functionality to appear.
5. Click **Insert HTML Help Object Tag** button.

The HTML Help ActiveX Control Wizard opens.



6. Select the ActiveX object from the drop-down menu and follow the instructions provided by the wizard. Once you are finished, Doc-To-Help marks the tag code as [HTML PassThrough](#).

Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | HTML Help ActiveX Control** menu command.

Using WinHelp Macros in Word

By using the [WinHelpMacro](#) property, Doc-To-Help allows you easy access to any of the more than fifty macros which you can use to enhance the capabilities of your Help system. Macro uses include:

- Modifying the WinHelp button bar and menus.
- Adding keyboard shortcuts to WinHelp commands.
- Jumping to other Help files.
- Launching Windows applications.

Note: Help macros are only supported by WinHelp; not by HTML-based platforms.

Running WinHelp Macros

There are four ways you can run a Help macro.

Startup Macros

A Startup macro is run whenever the user loads the help file. Examples of Startup macros include adding new buttons and menus, registering a function or functions in a Dynamic Link Library (DLL) and playing sound. The settings for Startup macros are stored in the [Config] section of the Help Project File (HPJ File).

Help Topic Macros

You can create Windows Help macros that start whenever the user enters a certain topic. This is useful for customizing topics with buttons, menu items, and markers. For example, you may want certain buttons and menu items to be available only when a certain topic is displayed. Or perhaps you want to play a particular sound when displaying a topic.

Hotspot Macros

Hotspot macros look like jump text, but when you click on them they execute Help macros. Examples of Hotspot macros are text that when clicked runs another Windows application or prints the help topic.

Window Macros

Window macros are associated with the windows defined in the Project Editor. They execute when a window is activated. You might, for example, want to play a warning sound when an error message window is displayed. Window macros are added in the Macros tab of the **Window Settings** dialog box.

Sample WinHelp Macro Uses

The following sections give a brief overview of some of the enhancements that can be incorporated into your WinHelp files. In many cases the same macros can be run in startup, topic, hotspot or windows modes.

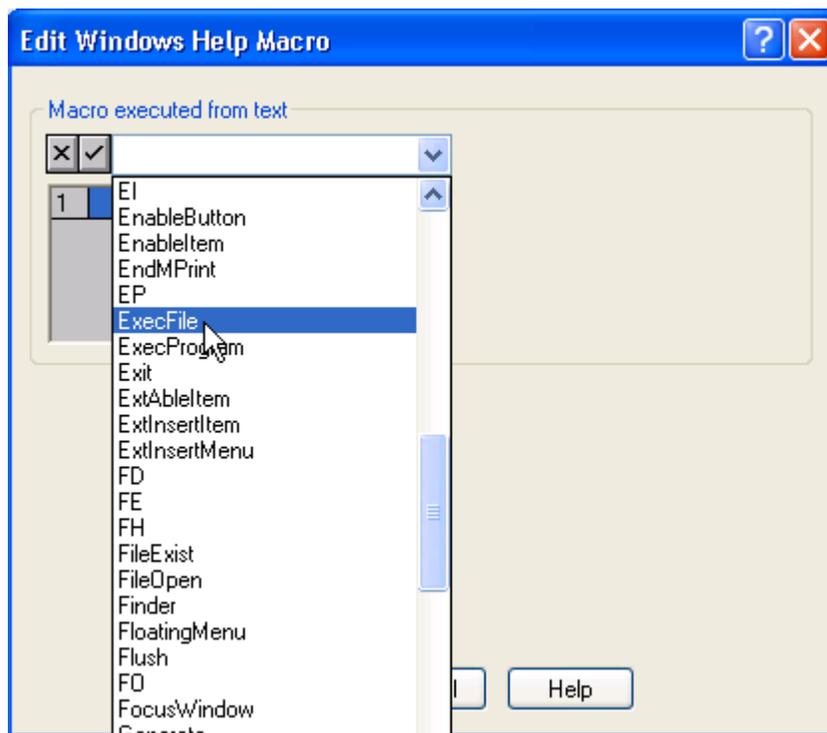
Launching an Application with Hotspot Macros

This procedure turns any text you choose into a hotspot that will open any application located in your Windows registry. In the following example, when the hotspot is clicked, Windows Notepad opens.

1. Open the Word source document that will contain the hotspot.
2. Select the text for the hotspot and click the **Insert Help Macro** button .

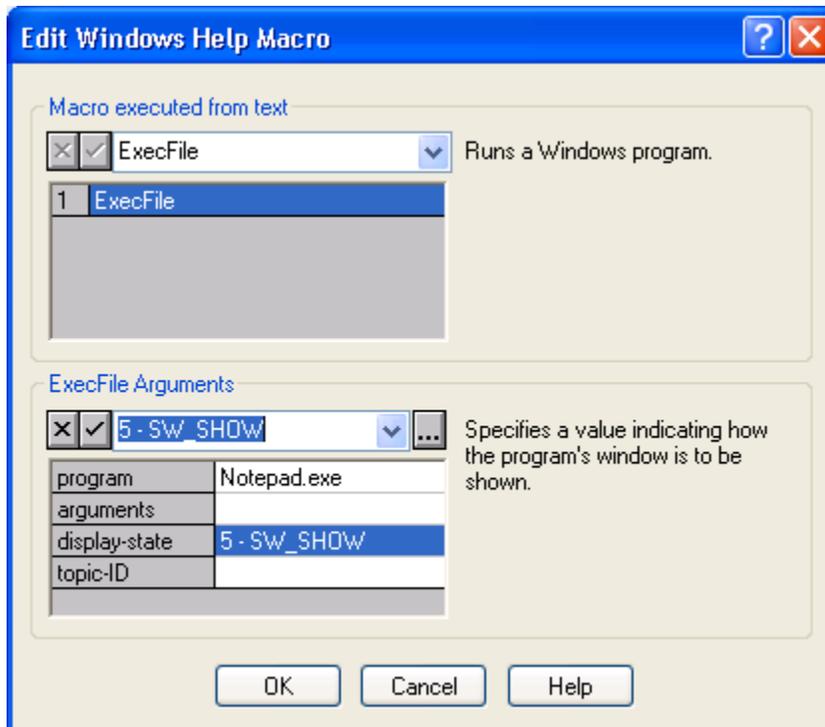
The **Edit Windows Help Macro** dialog opens.

3. From the drop-down list box in the **Macro executed from text** area, select **ExecFile**.



4. Type **Notepad.exe** in the textbox at the top of the **ExecFile Arguments** area and click the checkmark button.
5. Select the display-state box, and from the drop-down at the top of the ExecFile Arguments area, select the value, **5 – SW_SHOW**. Display-state determines how the application appears when it is started. In this case you want the Notepad.exe to open as it normally would when the hotspot is clicked.

Your dialog should look like the example below.



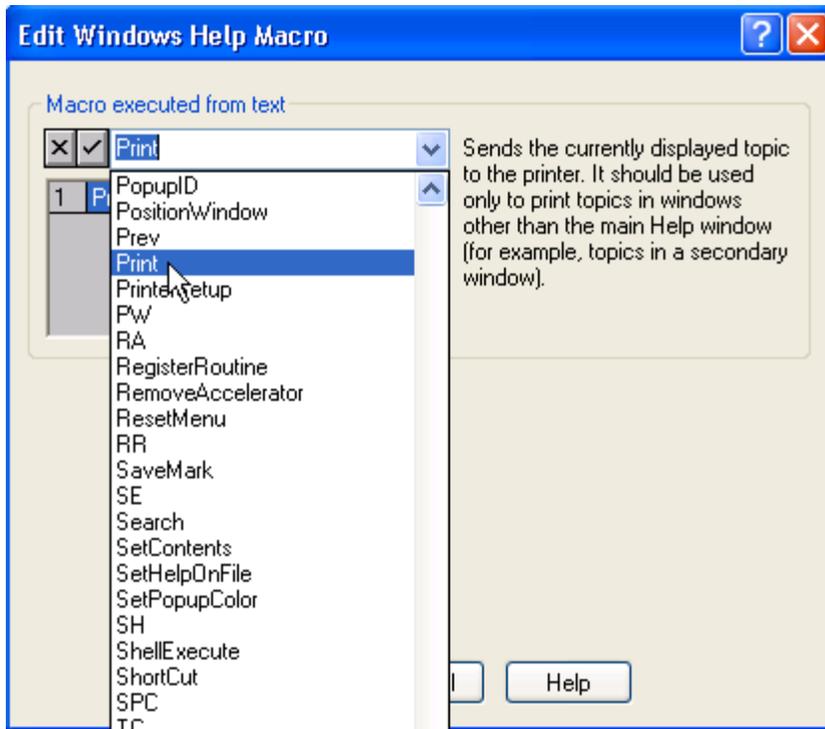
6. Click **OK** to insert the Macro Comment into your source document.
7. **Close** the source document and build your WinHelp file.
8. Open your WinHelp file and click the hotspot to test the macro.

Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Help Macro** menu command.

Printing with Hotspot Macros

This procedure turns any text you choose into a hotspot that will print your current Help topic.

1. Open the source document that will contain the hotspot.
2. Select the text for the hotspot and click the **Insert Help Macro** button  .
The **Edit Windows Help Macro** dialog opens.
3. From the drop-down list box in the **Macro executed from text** area, select **Print**.



4. Click **OK** to insert the Macro Comment into your source document.
5. **Close** the source document and build your WinHelp file.
6. Open your WinHelp file and click the hotspot to test the macro.

Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Help Macro** menu command.

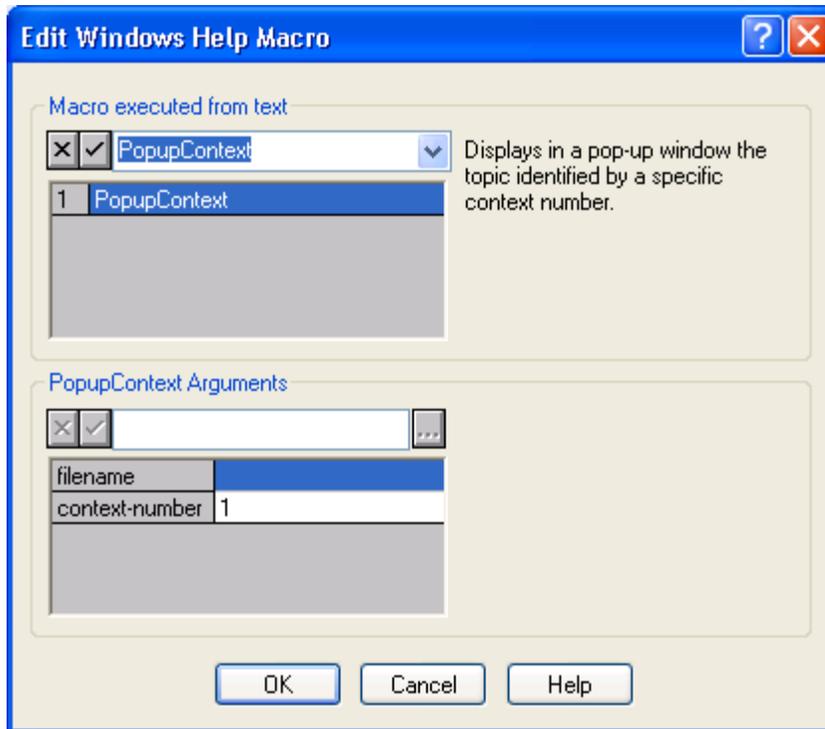
Displaying Help File Topic with Hotspot Macros

You can also use hotspot macros to perform certain Help-specific commands. This procedure allows you to display a Help topic which is located in a separate Help file.

1. Open the source document that will contain the hotspot.
2. Select the text for the hotspot and click the **Insert Help Macro** button .

The **Edit Windows Help Macro** dialog opens.

3. From the drop-down list box in the **Macro executed from text** area, select **PopupContext** (or its short form, **PC**).

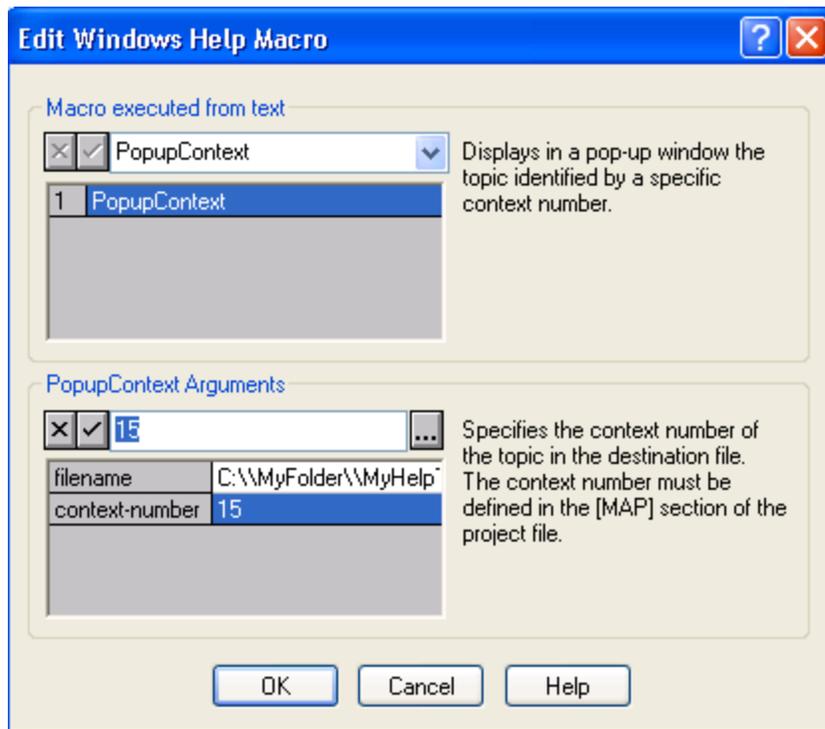


4. Type the name and path of your second Help file in the textbox at the top of the **PopupContexts Arguments** area and click the **checkmark** button.

Note: When defining the path to the Help file, use double back-slashes (\\) in place of the typical single back-slash. Example – C:\\MyFolder\\MyFile.hlp.

5. Select the **context-number** box, and type the Context ID number of the topic you wish to display. For more information on Context Ids, see [Context Sensitive Help](#) (page 297).

Your dialog should look similar to the example below.



6. Click **OK** to insert the Macro Comment into your source document.
7. **Close** the source document and build your WinHelp file.
8. Open your WinHelp file and click the hotspot to test the macro.

Alternatively, you can perform the same steps with the **Insert | ComponentOne Doc-To-Help | Help Macro** menu command.

Adding New Menus and Menu Items with Startup Macros

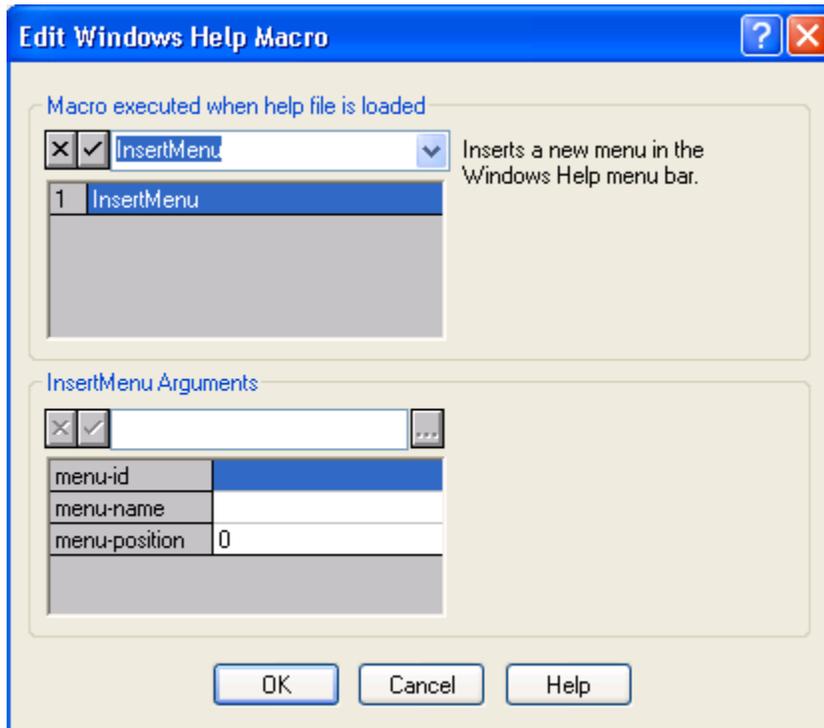
In the procedure below we will create several macros that add a new menu and new menu item when a Help file is loaded. In particular, we will add a new menu called Programs that has a menu item called Excel. Selecting Excel will load Microsoft Excel.

1. Open your Help project.
2. Select the **Project** icon, then **Help Targets** from the left pane.
3. Select your WinHelp target from the right pane.
4. From the property pane, click the [WinHelpMacro](#) property.

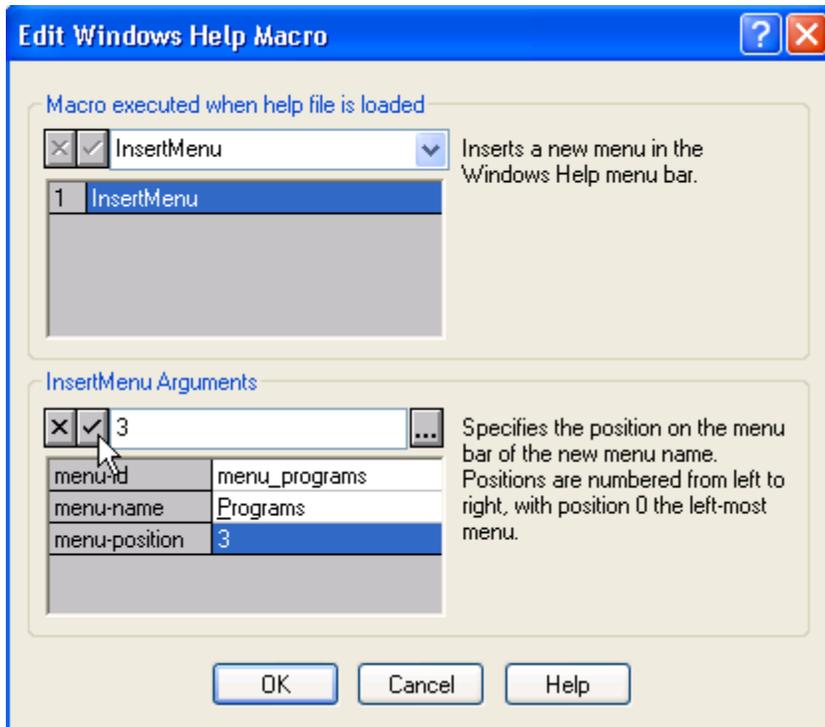
Property	Value
Name	StyleGuide WinHelp
SkipGlossary	False
StyleSheet	C1H_Help_full.css
Template	C1H_HELP.DOT
Title	
WinHelpMacro	
Natural Search	
AnswerCount	20
ConfidenceLevel	40

The **Edit Windows Help Macro** dialog opens.

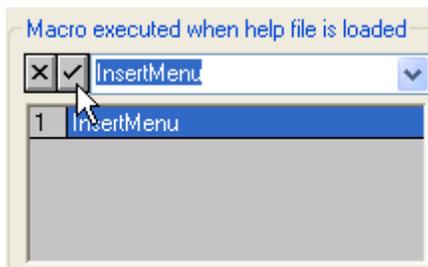
- From the drop-down list box in the **Macro executed when help file is loaded** area, select **InsertMenu**.



- Select the **menu-id** box and type **menu_programs** in the textbox at the top of the **InsertMenu Arguments** area. This is the name you use to identify the menu. It can be any unique identifier. You can also add items to Help's Standard menus.
- Select the **menu-name** box and type **&Programs** in the textbox at the top of the **InsertMenu Arguments** area. This is the text that you want to appear on the menu. The character following the ampersand is the hotkey.
- Select the **menu-position** box and type **3** in the textbox at the top of the **InsertMenu Arguments** area. This specifies the position on the menu bar of the new menu name. Positions are numbered from left to right, with the leftmost being position 0.
- Click the **InsertMenu Arguments** checkmark button.

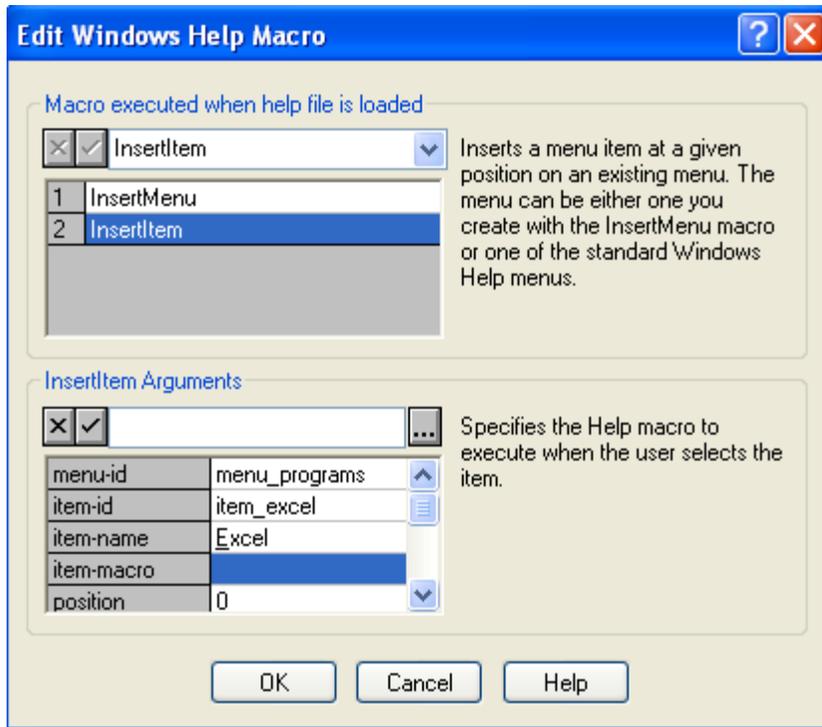


10. In the **Macro executed when help file is loaded** click the **checkmark** button. A new blank cell opens beneath the **InsertMenu** macro.



At this point we have added a new menu called **Programs** which will appear to the right of the **Bookmark** menu. In the next series of steps we will add the menu item **Excel** that when clicked will run Microsoft Excel.

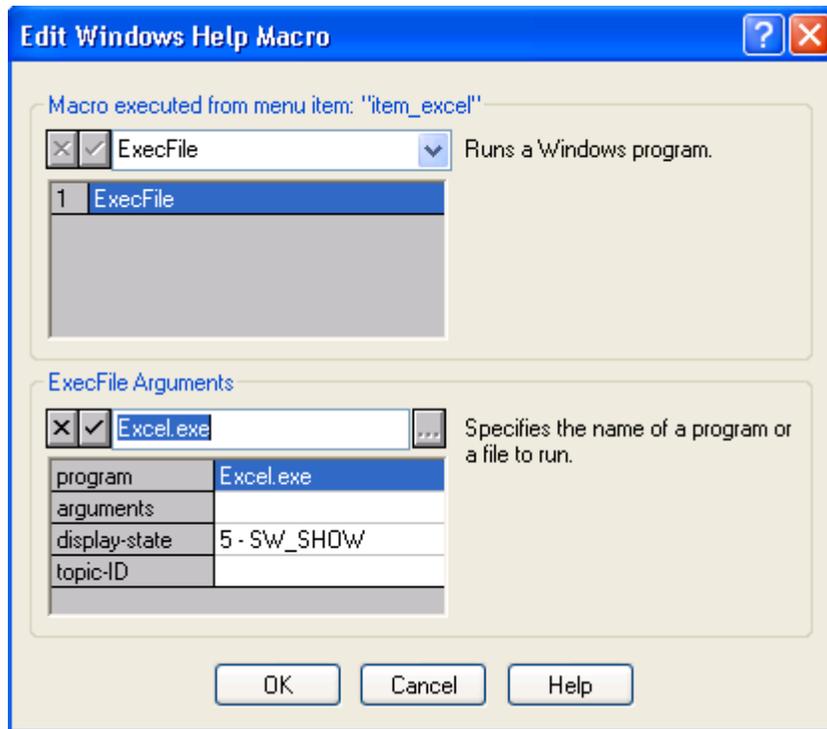
11. From the drop-down list box, select **InsertItem** for the second cell.
12. Select the **menu-id** box and type **menu_programs** in the textbox at the top of the **InsertItems Arguments** area. This must be a unique identifier to identify the menu or one of the names Help uses to identify its standard menus (if you're adding to a standard menu).
13. Select the **item-id** box and type **item_excel** in the textbox at the top of the **InsertItems Arguments** area. This is the label Help uses internally to identify the menu item. Use any label, but it must be different from any other item-id you have used.
14. Select the **item-name** box and type **&Excel** in the textbox at the top of the **InsertItems Arguments** area. This will place the word **Excel** on the **Programs** menu and make the letter "E" its keyboard shortcut.
15. Select **item-macro**, then click the **ellipsis** button.



A second Help Macro dialog box appears. You have to specify which macro you want executed when the user clicks on the menu item Excel. In this example, when the menu item is clicked, Excel loads. This second dialog box allows you to specify the macro and arguments you want executed when the user clicks the Excel menu item.

16. From the drop-down list box in the **Macro executed from menu item: item_excel** area, select **ExecFile**.
17. Type **Excel.exe** in the textbox at the top of the **ExecFile Arguments** area and click the **checkmark** button.
18. Select the display-state box, and from the drop-down at the top of the **ExecFile Arguments** area, select the value, **5 – SW_SHOW**.

Your dialog should look like the example below.



19. Click **OK**. The first **Help Macro** dialog box reappears and the **ExecProgram** macro appears in the item-macro box in the **InsertItem Arguments** section.
20. In the position box, type **0**. This is the number specifying where the new menu item will appear. The number must be an integer. The first item on a menu is represented by 0, the second item is 1, and so forth. So in this example the Exec will be the first item in the **Programs** menu.
21. Click the **checkmark** button in the **InsertItem Arguments** area. This completes the arguments.
22. Click the checkmark button in the **Macro executed when help file is loaded** area. This completes the macro definition.
23. Click **OK**. Doc-To-Help adds the appropriate macro to your Help Project File (HPJ File).
24. Build your WinHelp file.
25. Open your WinHelp file and test the menu.

Adding a Button with Startup Macros

You can use startup macros to create a button that appears every time a user opens the Help file. The following example creates a button labeled WinHelp Help that appears on the button bar every time the Help file is open. When the button is clicked, it takes the user to the WinHelp help file.

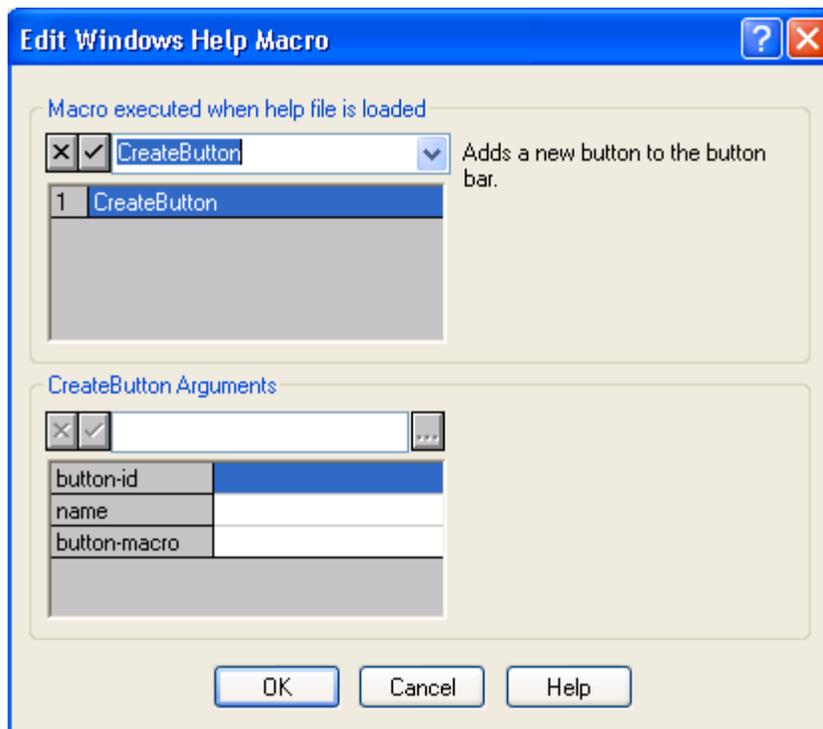
In this example we will use "nested" macros. You can have a single macro use up to two more additional macros. This nesting capability lets you combine the features of macros. In this example, two macros are used: CreateButton and ExecFile. ExecFile is the nested macro.

1. Open your Help project.
2. Select the **Project** icon, then **Help Targets** from the left pane.
3. Select your WinHelp target from the right pane.
4. From the property pane, click the [WinHelpMacro](#) property.

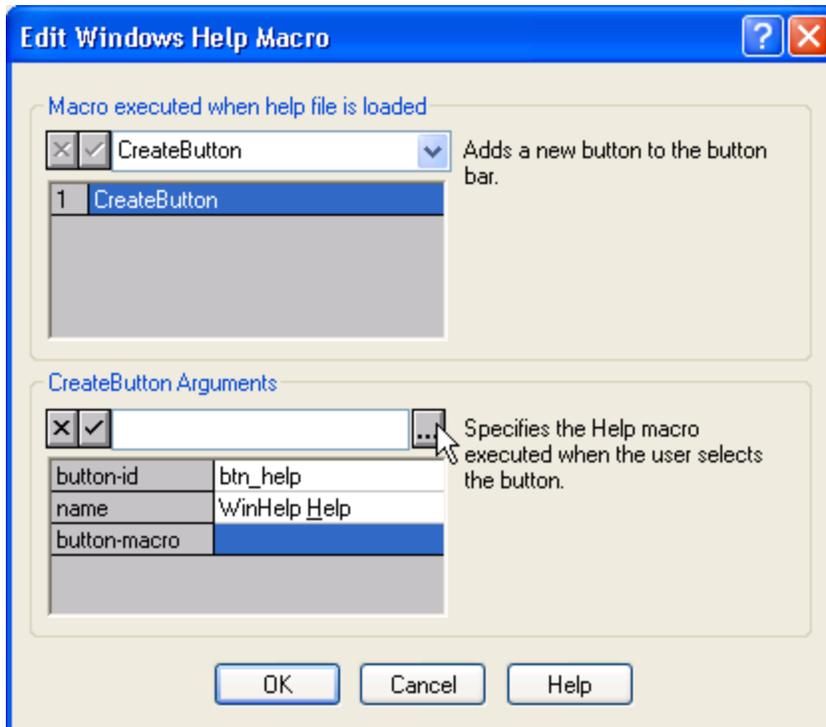
Property	Value
Name	StyleGuide WinHelp
SkipGlossary	False
StyleSheet	C1H_Help_full.css
Template	C1H_HELP.DOT
Title	
WinHelpMacro	
Natural Search	
AnswerCount	20
ConfidenceLevel	40

The **Edit Windows Help Macro** dialog opens.

- From the drop-down list box in the **Macro executed when help file is loaded** area, select **CreateButton**.

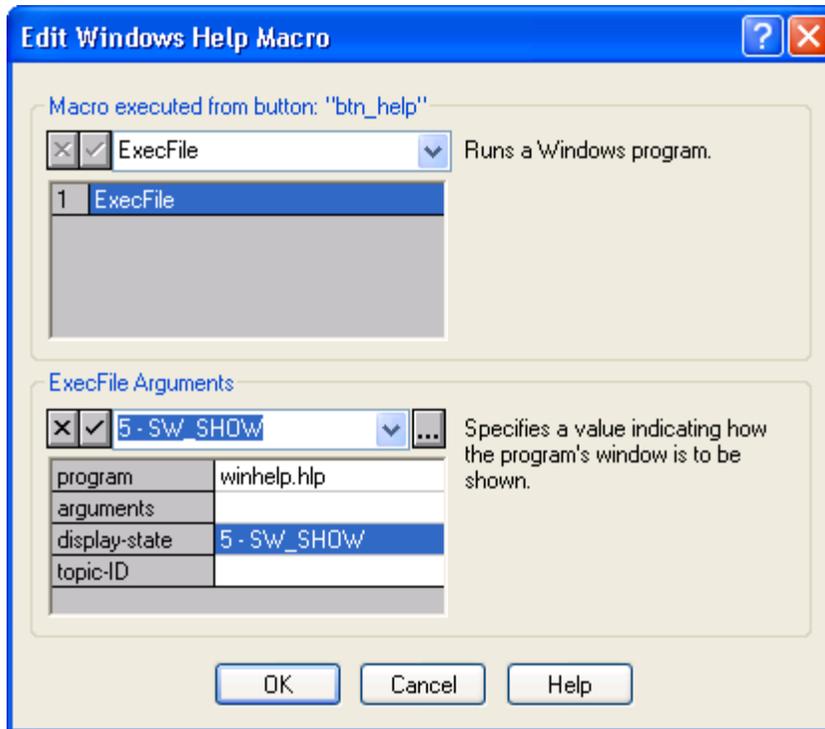


- Select the **button-id** box and type **btn_help** in the textbox at the top of the **CreateButton Arguments** area. This is the name you use to identify the button. It can be any unique identifier.
- Select the **name** box and type **WinHelp &Help** in the textbox at the top of the **CreateButton Arguments** area. This is the text that you want to appear on the button. The character following the ampersand is the hotkey.
- Select the **button-macro** cell and click the **ellipsis** button as shown below.



A second Help Macro dialog box appears. You specify here which macro you want executed when the user clicks on the WinHelp Help button. In this example, when the button is clicked, the WinHelp Help file opens.

9. From the drop-down list box, select **ExecFile**. The **ExecFile Arguments** section appears in the dialog box.
10. Select the program box and type **winhelp.hlp** in the textbox at the top of the **ExecFile Arguments** area. This is the file that will appear when the user clicks on the button.
11. Select the **display-state** box, and from the drop-down at the top of the **ExecFile Arguments** area, select the value, **5 – SW_SHOW**. Your dialog should look like the example below.

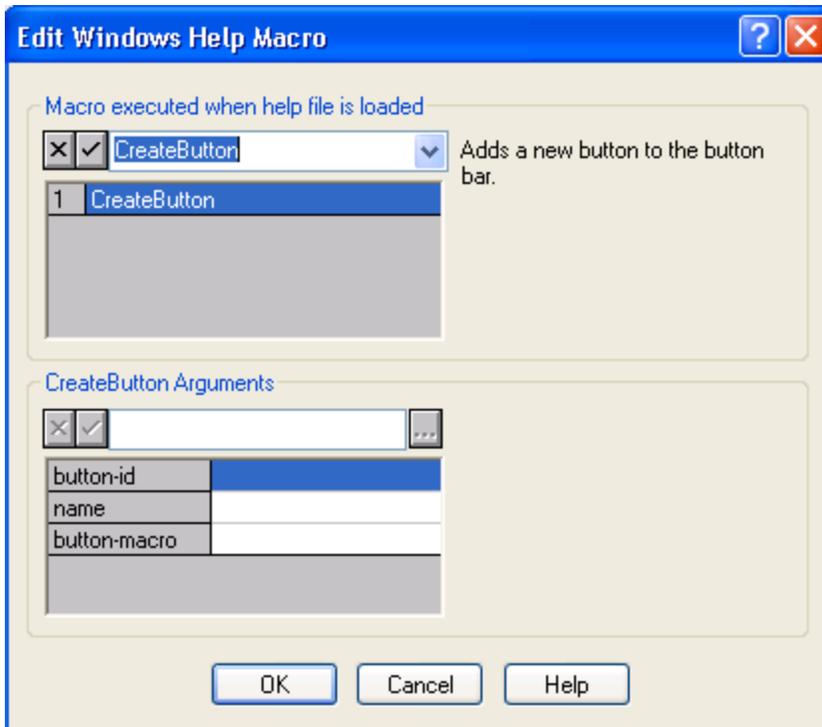


12. Click **OK** to return to the first dialog box.
13. Click **OK**.
14. Build your WinHelp file.
15. Open your WinHelp file and test the button.

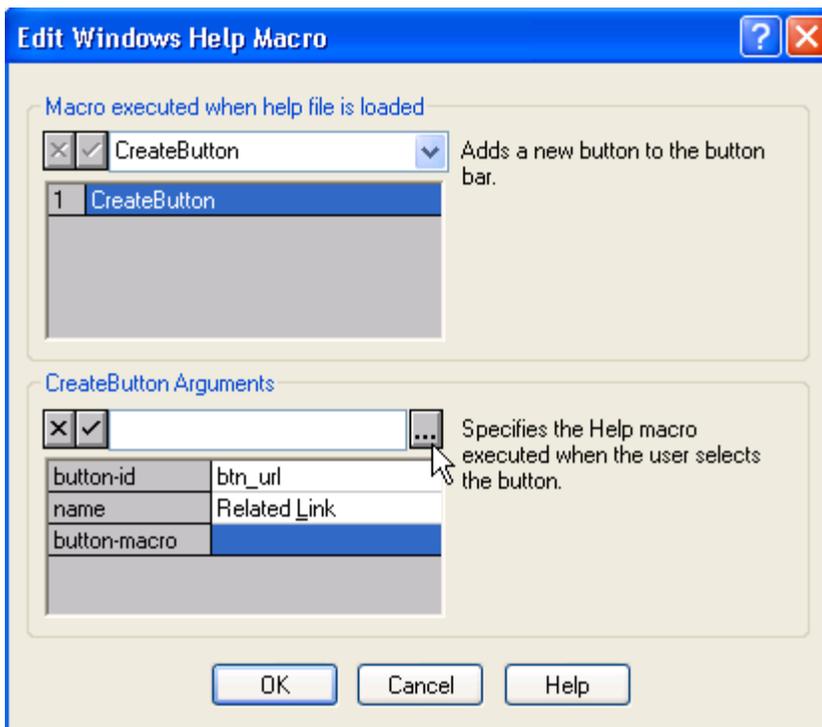
Adding a Button with Topic Macros

You can create Windows Help macros that start whenever the user enters a certain topic. This is useful for customizing topics with buttons, menu items, and markers. The procedure below uses "nested" macros. In this example, you will provide a button that takes the user to a URL you have defined.

1. Open your Help project.
2. Select the **Topics** icon.
3. Select the topic you want to use from the right pane.
4. From the property pane, click the [WinHelpMacro](#) property. The **Edit Windows Help Macro** dialog opens.
5. From the drop-down list box in the **Macro executed from topic** area, select **CreateButton**.

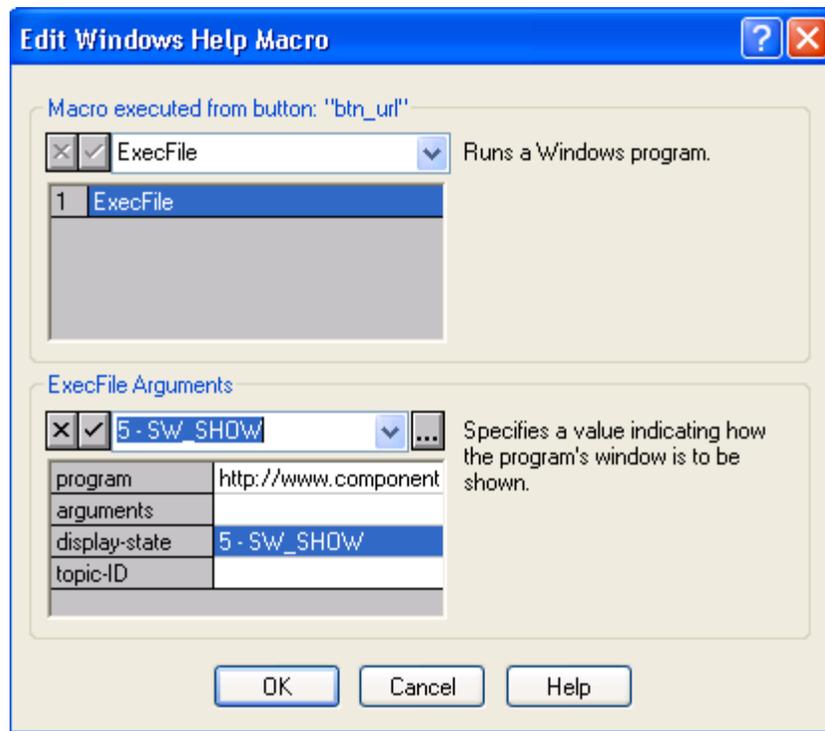


6. Select the **button-id** box and type **btn_url** in the textbox at the top of the **CreateButton Arguments** area. This is the name you use to identify the button. It can be any unique identifier.
7. Select the name box and type **Related &Link** in the textbox at the top of the CreateButton Arguments area. This is the text that you want to appear on the button. The character following the ampersand is the hotkey.
8. Select the **button-macro** cell and click the **ellipsis** button as shown below.



A second Help Macro dialog box appears. You specify here which macro you want executed when the user clicks on the Related Link button. In this example, when the button is clicked, a browser open with the ComponentOne Web site.

9. From the drop-down list box, select **ExecFile**. The **ExecFile Arguments** section appears in the dialog box.
10. Select the program box and type **http://www.componentone.com** in the textbox at the top of the **ExecFile Arguments** area.
11. Select the **display-state** box, and from the drop-down at the top, select the value, **5 – SW_SHOW**. Your dialog should look like the example below.



12. Click **OK** to return to the first dialog box.
13. Click **OK**.
14. Build your WinHelp file.
15. Open your WinHelp file, go to the topic you used for your button, and test the button.

Editing and Deleting Macros

Startup, Topic and Window Macros

Startup, topic and window macros can be edited or removed by selecting and modifying the related [WinHelpMacro](#) property as needed.

Hotspot Macros

Hotspot macros can be deleted by removing the corresponding comment from the source document. Modify hotspot macros by deleting the existing comment and reinserting the new macro.

Macro Definitions

Following is an alphabetical listing of all the Windows Help Macros. When you use the Doc-To-Help Macro Editor, you'll see an explanation for each macro when you select that macro from a drop-down list. More details on Help Macros can be found in the Help Author's Guide.

Macro Name	Description
About	Displays the About dialog box.
AddAccelerator (AA)	Assigns a Help macro to an accelerator key (or key combination) so that the macro is run when the user presses the accelerator key(s).
ALink	Searches for keywords specified by A footnotes.
Annotate	Displays the Annotation dialog box.
AppendItem	Appends a menu item to the end of a menu you create with the InsertMenu macro.
Back	Displays the previous topic in the Back list. The Back list includes the last 40 topics the user has displayed since starting WinHelp.
BackFlush	Removes the back history list from the current window. This macro does not affect the history list displayed in the History window.
BookmarkDefine	Displays the Define dialog from the Bookmark menu.
BookmarkMore	Displays the More dialog from the Bookmark menu. The More command appears on the Bookmark menu if the menu lists more than nine bookmarks.
BrowseButtons	Adds browse buttons to the button bar.
ChangeButtonBinding	Assigns a Help macro to a Help button.
ChangeEnable (CE)	Assigns a macro to a button bar button and enables that button.
ChangeItemBinding (CIB)	Assigns a Help macro to an item previously added to a Windows Help menu using the AppendItem macro.
CheckItem (CI)	Places a checkmark beside a menu item.
CloseSecondarys (CS)	Closes all but the current secondary window.
CloseWindow	Closes either a secondary window or the main Help window.
Compare	Displays a Help file in a second instance of WinHelp. The current Help file and the second Help file are displayed side-by-side. Most actions performed in one Help file will be automatically reflected in the other file.
Contents	Displays the Contents topic in the current Help file.
ControlPanel	Opens a control panel applet with a specific tab on top.
CopyDialog	Displays the Copy dialog from the Edit menu.
CopyTopic	Copies all the text in the currently displayed topic to the Clipboard.
CreateButton (CB)	Adds a new button to the button bar.
DeleteItem	Removes a menu item that was added by using the AppendItem macro.
DeleteMark	Removes a text marker added with the SaveMark macro.
DestroyButton (DB)	Removes a button added with the CreateButton macro.
DisableButton	Grays out a button added with the CreateButton macro. This button cannot be used in the topic until an EnableButton macro is executed.
DisableItem (DI)	Grays out a menu item added with the AppendItem macro. The menu item cannot be used in the topic until an EnableItem macro is executed.
EnableButton (EB)	Re-enables a button disabled with the DisableButton macro.
EnableItem (EI)	Re-enables a menu item disabled with the DisableItem macro.

Macro Name	Description
EndMPrint	Dismisses the printing message box and terminates the printing of multiple topics.
ExecFile (EF)	Runs a program or runs a program and opens a specified file.
ExecProgram (EP)	Executes a Windows application.
Exit	Exits the Windows Help application: the same effect as selecting Exit from the File menu.
ExtAbleItem	Enables or disables a menu item.
ExtInsertItem	Inserts a menu item at a given position on an existing menu.
ExtInsertMenu	Inserts a submenu in a previously defined menu.
FileExist	Checks to see whether the specified file or program exists.
FileOpen	Displays the Open dialog box from the File menu.
Find	Displays the Find tab in the Help Topics dialog box.
Finder	Displays the Help Topics dialog box.
FloatingMenu	Displays the context (floating) menu at the current mouse cursor position. This menu also appears when users click a topic using their right mouse button.
Flush (FH)	Causes WinHelp to process any pending messages, including previously called macros.
FocusWindow	Changes the focus to the specified window, either the main Help window or a secondary window.
Generate	Posts a message to the currently active Help window.
GotoMark	Jumps to a marker set with the SaveMark macro.
HelpOn	Displays the Help file for the Windows Help application. The macro carries out the same action as choosing the How to Use Help command on the Help menu.
HelpOnTop	Toggles the authored on-top state of a Help window.
History	Displays the history list, which shows the last 30 topics the user has viewed since opening a Help file in Windows Help. It has the same effect as choosing the History button.
IfThen	Executes a Help macro if a given marker exists, using the IsMark macro to make the test. The result of the test can be reversed by enclosing the IsMark macro within the Not macro.
IfThenElse	Executes one of two Help macros depending on whether or not a marker exists, as tested by the IsMark macro. The result of the test can be reversed by enclosing the IsMark Macro within the Not macro.
InitMPrint	Initializes WinHelp in preparation for printing multiple topics.
InsertItem	Inserts a menu item at a given position on an existing menu. The menu can be either one you create with the InsertMenu macro or one of the standard Windows Help menus.
InsertMenu	Inserts a new menu in the Windows Help menu bar.
IsBook	Determines whether WinHelp is running as a standalone system (a double-clicked book icon), or if it is being run from a program. This macro can be used as the first parameter of an IfThen or IfThenElse macro to take specific action depending on whether the current Help file is being run as a double-clicked book icon.

Macro Name	Description
IsMark	Used with IfThen and IfThenElse to determine if a text marker previously created with the SaveMark macro exists.
IsNotMark	Tests whether or not a marker that was set by the SaveMark macro exists.
JumpContents	Jumps to the Contents topic of a specified Help file.
JumpContext (JC)	Jumps to a topic identified by a context number. The context is identified by an entry in the [MAP] section of the HPJ file.
JumpHash	Jumps to a topic identified by a hash number.
JumpHelpOn	Jumps to the Contents topic of the How to Use Help file. The How To Use Help file is either the default WINHELP.HLP or WINHLP32.HLP.
JumpId (JI)	Jumps to the topic with the specified context string in the Help file.
JumpKeyword (JK)	Loads the indicated Help file, searches through the K keyword table, and displays the first topic containing the index keyword specified in the macro.
KLink	Searches for keywords specified by K-footnotes.
MPrintHash	Prints a topic identified by a hash number. This macro must be used in conjunction with the InitMPrint and EndMPrint macros.
MPrintId	Prints a topic. This macro must be used in conjunction with the InitMPrint and EndMPrint macros.
Next	Displays the next topic in the browse sequence for the Help file.
NoShow	Prevents a Help window from being displayed if it has not already been displayed.
Not	Returns zero (FALSE) if the marker text specified by the SaveMark macro exists or non-zero (TRUE) if the marker text does not exist. When used with a macro, the Not macro reverses the results of the macro. Used with IfThen and IfThenElse to determine if a text marker previously created with the SaveMark macro does not exist.
PopupContext (PC)	Displays in a pop-up window the topic identified by a specific context number.
PopupHash	Displays in a pop-up window the topic identified by a hash number.
PopupId (PI)	Displays a topic from a specified file in a pop-up window.
PositionWindow	Sets the size and position of a window.
Prev	Displays the previous topic in the browse sequence for the Help file. If the currently displayed topic is the first topic of a browse sequence, this macro does nothing.
Print	Sends the currently displayed topic to the printer. It should be used only to print topics in windows other than the main Help window (for example, topics in a secondary window).
PrinterSetup	Displays the Printer Setup dialog box from the File menu.
RegisterRoutine (RR)	Registers a function within a DLL as a Help macro.
RemoveAccelerator	Removes the assignment of a macro to an accelerator key or key combination.
ResetMenu	Deletes all added menus and menu items, restores and enables all standard menu items, and restores the item bindings of all standard menu items to their defaults.
SaveMark	Saves the location of the currently displayed topic and file and associates a text marker with that location. The GotoMark macro can then be used to jump to this location.

Macro Name	Description
Search	Displays the dialog for the Search button, which allows users to search for topics using keywords defined by the K footnote character.
SetContents	Designates a specific topic as the Contents topic in the specified Help file.
SetHelpOnFile	Designates the specific Help file that replaces WINHELP.HLP, the default Using Help file in the Windows environment. This macro is obsolete in Windows 95 and will be ignored.
SetPopupColor	Sets the background color for all subsequent pop-up windows.
ShellExecute	Opens or prints the specified file.
ShortCut	Runs the specified program if it is not already running. If the specified program is running, WinHelp activates it. If the wParam parameter is specified, a WM_COMMAND message with the specified wParam and lParam values are sent to the program.
TCard	Sends a message to the program that is invoking WinHelp as a training card.
Test	Runs an internal WinHelp test.
TestALink	Tests whether an ALink macro has an effective link to at least one topic.
TestKLink	Tests whether a KLink macro has an effective link to at least one topic.
UncheckItem (UI)	Removes a check mark besides a menu item.
UpdateWindow	Jumps to the topic with the specified topic ID in the specified window, and then returns the focus to the window that called the macro.

Doc-To-Help Shortcut Keys for Word

Doc-To-Help provides users with shortcut keys to automate some of the most common commands used while in Microsoft Word. The following key combinations can be used when formatting text using the **Doc-To-Help** toolbar in Word:

Key Combination	Result
Ctrl + Numeric Keypad Plus	Adds one point of space after the current paragraph.
Ctrl + Alt + Numeric Keypad Plus	Adds one point of space before the current paragraph.
Ctrl + Numeric Keypad Minus	Subtracts one point of space after the current paragraph.
Ctrl + Alt + Numeric Keypad Minus	Subtracts one point of space before the current paragraph.
Ctrl + Shift + 1	Applies the Heading 1 style.
Ctrl + Shift + 2	Applies the Heading 2 style.
Ctrl + Shift + 3	Applies the Heading 3 style.
Ctrl + Shift + 4	Applies the Heading 4 style.
Ctrl + Shift + B	Applies the Body text style.
Ctrl + Shift + G	Runs the Sort Glossary command.
Ctrl + Shift + H	Runs the Insert HTML Help ActiveX Control command.
Ctrl + Shift + L	Applies the List style.

Key Combination	Result
Ctrl + Shift + M	Runs the Insert Help Macro command.
Ctrl + Shift + S	Applies special bold formatting.
Ctrl + Shift + T	Runs the Standard Doc-To-Help Table command.
Ctrl + Shift + W	Runs the Apply Conditional Text command.
Ctrl + Shift + X	Runs the Add Dynamic Link command.
Ctrl + Shift + Y	Opens the Add Topic Link dialog box.
Ctrl + F1	Help on Doc-To-Help.
Ctrl + F5	Runs the View Target command.
F5	Runs the Make Target command.
Alt + 0	Applies the C1H Continue or C1H Continue 2 style.
Alt + 1	Applies the C1H Number style.
Alt + 2	Applies the C1H Number 2 style.
Alt + 3	Applies the C1H Bullet style.
Alt + 4	Applies the C1H Bullet 2 style.
Alt + 5	Applies the C1H Bullet 2A style.
Alt + Shift + C	Runs the View Conditional Text command.
Alt + Shift + G	Runs the Add Glossary Terms command.
Alt + M	Inserts a margin note.
Alt + Shift + M	Inserts a link to a margin note.
Alt + Shift + R	Completes a cross-reference.
Alt + F12	Switches to the Project Editor.

The following key combinations can be used when formatting text using the **D2HML Styles** toolbar in Word:

Key Combination	Result
Alt+Ctrl+J	Applies the C1H Jump style.
Alt+Ctrl+P	Applies the C1H Popup style.
Alt+Ctrl+K	Applies the C1H Keyword Link style.
Alt+Ctrl+G	Applies the C1H Group Link style.
Alt+Ctrl+E	Applies the C1H Inline Expand style.
Alt+Ctrl+D	Applies the C1H Inline Dropdown style.
Alt+Ctrl+U	Applies the C1H Inline Popup style.
Alt+Ctrl+W	Applies the C1H Index style.
Alt+Ctrl+Shift+W	Applies the C1H Index Invisible style.
Alt+Ctrl+R	Applies the C1H Group style.

Key Combination	Result
Alt+Ctrl+Shift+R	Applies the C1H Group Invisible style.
Alt+Ctrl+L	Applies the C1H Link Tag style.
Alt+Ctrl+Shift+L	Applies the C1H Link Tag Invisible style.
Alt+Ctrl+I	Applies the C1H Context ID style.
Alt+Ctrl+N	Applies the C1H Contents Title style.
Alt+Ctrl+T	Applies the C1H Topic Properties style.
Alt+Ctrl+O	Applies the C1H Online style.
Alt+Ctrl+M	Applies the C1H Manual style.
Alt+Ctrl+C	Applies the C1H Conditional style.
Alt+Ctrl+F	Erases any formatting applied to text.
Alt+Ctrl+H	Shows any invisible hot spots.
Alt+Ctrl+S	Formats any selection with any of the D2HML styles.

Conditional Text and Attributes

As you write documentation, you will no doubt write passages that you don't want included in the accompanying online Help file. Likewise, some content may be appropriate for the Help file but not the written documentation. You may even have content that you want targeted at specific audiences. Yet, if you are looking for the ease of single-source Help authoring, making separate files for each platform would defeat this entire concept.

There's a better solution: Doc-To-Help lets you mark text, graphics, whole topics, or entire documents to be included in or excluded from either the manual or any combination of online Help platforms using the conditional text and attribute features.

Specifying Conditional Text

There is more to single-source authoring than merely converting documents to the appropriate format. This is particularly true when designing a Help system for both online and print media. In many cases creating unique content for different platforms would better suit your audience. Here are some examples in which deviating from the single-source philosophy has its advantages:

- You use graphics in the printed manual, but omit them from the online Help to conserve screen real estate and minimize the time needed to load pages.
- You provide links to related topics in the online Help, but omit them from the printed manual to avoid disrupting the narrative flow.
- You need to describe a set of procedures, each of which shares a common set of preliminary steps. In the online Help, you include the preliminary steps at the beginning of each procedural topic to provide adequate context for the reader. In the printed manual, you describe the preliminary steps once, immediately before the procedural topics, to keep them from becoming tedious and repetitive.

Fortunately, Doc-To-Help's conditional text feature can help resolve design issues such as these. By allowing you to designate entire source documents or just portions of them for individual platforms, Help targets or attributes (or combinations thereof), Doc-To-Help gives you full control over the content of each one—without sacrificing the benefits of having a single-source project.

Doc-To-Help implements conditional text in Microsoft Word using Doc-To-Help Markup Language (D2HML) comments or D2HML styles. Conditional text may consist of more than one paragraph and require style formatting. Microsoft Word does not allow you to apply style formatting to text already formatted with a conditional text style, so Doc-To-Help formats conditional text as a comment, by default. Then you can apply styles within the conditional text. You also have the option of formatting conditional text using D2HML styles. This is recommended for smaller ranges of conditional text not spanning multiple paragraphs and not containing style formatting.

Note: You can also specify conditional text in Word documents using the classic, non-D2HML method, but this is not the recommended method. It has been kept for compatibility with older Doc-To-Help projects. See [Using Classic Doc-To-Help Conditional Text](#) (page 405) for more information.

In HTML documents, all conditional text can be specified using D2HML styles since HTML has no limitations on style formatting within conditional text. To specify conditional text:

- Using D2HML Comments in Word documents

Select a range of text and click the **Conditional Text** button  on the **D2HML Styles** toolbar. Then, in the **Condition** window, choose the platform, Help target or attribute in which you want the range to be visible. Doc-To-Help applies the conditional text comment to the selected text.

- Using D2HML Styles in Word or HTML documents

Select a range of text and click the **Conditional Text** button  on the **D2HML Styles** toolbar in Word, FrontPage or Dreamweaver. Then, in the **Condition** window, choose the platform, Help target or attribute in which you want the range to be visible.

Note: If using Word documents, you must uncheck the **Apply as Comment** checkbox so that Doc-To-Help formats the text as a style rather than a comment.

Doc-To-Help applies the **C1H Conditional** character style to the selected text. The [Behavior](#) property of this style is set to *Conditional Text*, telling Doc-To-Help to treat it as such. If you are using an HTML editor other than FrontPage or Dreamweaver, you can apply the **C1HConditional** style or use HTML code (usually done by advanced HTML users). See [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information.

Doc-To-Help also gives you the option of applying conditions to whole topics in both Word and HTML source documents by using the Topic.[Condition](#) property. To apply a condition to a topic using D2HML, click anywhere within the topic (or page, if using a single topic document) where you want to apply the condition (no specific text must be selected) and press the **Topic Properties** button on the **D2HML Styles** toolbar. You can then use the dialog box to apply the condition. For more information, see [Topic Properties Hot Spot](#) (page 270).

Alternatively, you can use the Doc-To-Help Project editor to apply conditions. Select the topic in the project editor, click the [Condition ellipsis](#) button and choose the platform, Help target or attribute in which you want the range to be visible. The condition is applied to the entire topic.

If you want to apply a condition to an entire Word or HTML source document, you can use the Document.[Condition](#) property. Simply select the document in the Doc-To-Help project editor, and use the **Condition** dialog box to specify the condition.

Conditional Text Target Options

When defining conditional text, Doc-To-Help provides three target options; **Platforms**, **Help Targets** and **Attributes**.

Platforms	This option allows you to incorporate conditional text into any Help file defined with a specific platform.
Help Targets	This option allows you to incorporate conditional text into a specific default or user defined Help target.
Attributes	This option allows you to incorporate conditional text into a specific version of a Help file.

Platforms

HTML (Any)	If chosen, conditional text is viewable in all HTML and HTML Help help targets.
HTML Help 1.x	If chosen, conditional text is viewable only in HTML Help help targets.
JavaHelp	If chosen, conditional text is viewable only in JavaHelp 1.1 help targets.
Microsoft Help 2.0	If chosen, conditional text is viewable only in Microsoft 2.0 help targets.
NetHelp	If chosen, conditional text is viewable only in the NetHelp targets.
Online Help	If chosen, conditional text is viewable only in all online help targets.
Printed Manual	If chosen, conditional text is viewable only in printed manual help targets.
WinHelp 4.0	If chosen, conditional text is viewable only in WinHelp help targets.

Help Targets

Microsoft Help 2.0	This target produces the standard Microsoft Help 2.0. If chosen, conditional text will only be viewable in this target type.
HTML Help	This target produces Microsoft HTML Help, which requires Internet Explorer to be installed on the client machine. Internet Explorer version 3.0 is acceptable, but 4.0 or greater is preferred. If chosen, conditional text will only be viewable in this target type.
Printed Manual	This target produces a Microsoft Word document that includes a table of contents, an index and the individual subdocuments generated from your source documents. If chosen, conditional text will only be viewable in this target type.
NetHelp	This target produces browser neutral HTML in accordance with version 4.0 of the HTML specification published by W3C. If chosen, conditional text will only be viewable in this target type.
WinHelp	This target produces the standard 32-bit Windows Help 4.0. If chosen, conditional text will only be viewable in this target type.

Note: The Help Target list contains the default Help targets. Any additional Help targets that the user defines will also appear in the list.

Attributes

The attribute Help target list is populated by attributes that you have created within your project. Attributes provide the Help author with the ability to create conditional text based upon specific Help files. For more information, see [Using Attributes](#) (page 422).

Working with Conditional Text in Word Documents

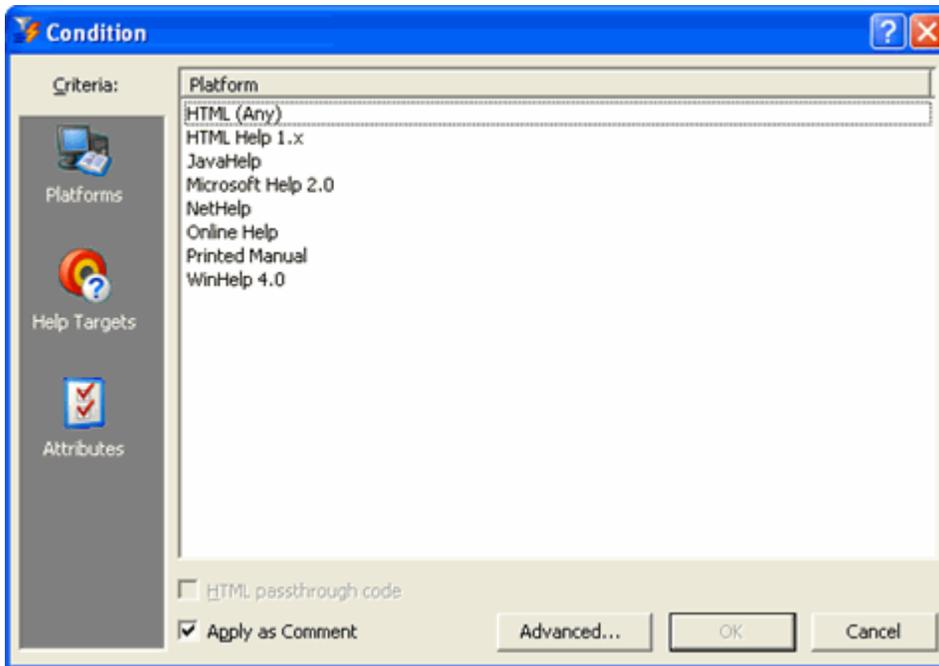
This section explains how to work with conditional text in Word source documents. There are two ways to specify conditional text in Word: using Doc-To-Help Markup Language (D2HML) and using classic, non-D2HML conditional text comments. D2HML is the preferred method, but the classic method has been kept to ensure compatibility with older Doc-To-Help projects. Both methods are described in the following sections, but it is strongly recommended that you use D2HML when specifying conditional text.

Using D2HML Conditional Text Comments

Using D2HML conditional text comments is the preferred method of specifying conditional text. You can use the **Conditional Text** button on the **D2HML Styles** toolbar, along with its **Condition** dialog box, to create the comments or you may choose to insert them manually.

Adding D2HML Conditional Text Comments

1. In Microsoft Word, select the text/image range to be converted into conditional text. Select the range as if you were going to copy it to the Clipboard.
2. On the **D2HML Styles** toolbar, click the **Conditional Text** button . Doc-To-Help displays the following dialog box.



3. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
4. From the list in the right pane, select the item in which you want to include the highlighted text or image.
Click **OK** to apply the D2HML conditional text comment to the selection. The selection now includes a comment after the text as in the example below.

[This sentence appears in online help targets only.]
[D2HML1]

Note: If you want the C1H Conditional D2HML style applied instead of a D2HML comment, uncheck the **Apply as Comment** checkbox.

You can make the text/image appear in multiple Help targets or attributes by applying multiple conditions. See [Applying Multiple Conditions to Text](#) (page 417) for more information.

Adding D2HML Conditional Text Comments Manually

1. In your Word source document, select the text/image range to be converted into conditional text. Select the range as if you were going to copy it to the Clipboard.
2. Select **Insert | Comment**.
3. Add the following text, which will become your comment text:

D2HML | <style name> | <secondary part>

- The **<style name>** is the name of a Doc-to-Help D2HML style such as C1H Conditional, C1H Manual, or C1H Online. For the C1H Manual and C1H Online styles, you only need to specify the style name because the secondary part is already set for these styles in the Doc-To-Help project.
- The **<secondary part>** is a D2HML hot spot secondary part.

For example, if you wanted to create conditional text that appears only in the NETHelp target, you would select your text/image range and insert the following comment text:

D2HML | C1H Conditional | platform=nethelp

Revision1, 1/24/2006 2:09:00 PM commented:
D2HML | C1H Conditional | platform=nethelp

will see :
is enom

The sentence appears in the NetHelp target. [Rev114]

You can make the text/image appear in multiple Help targets or attributes by applying multiple conditions. See [Manually Applying Multiple Conditions](#) (page 419) for more information.

Using Classic Doc-To-Help Conditional Text

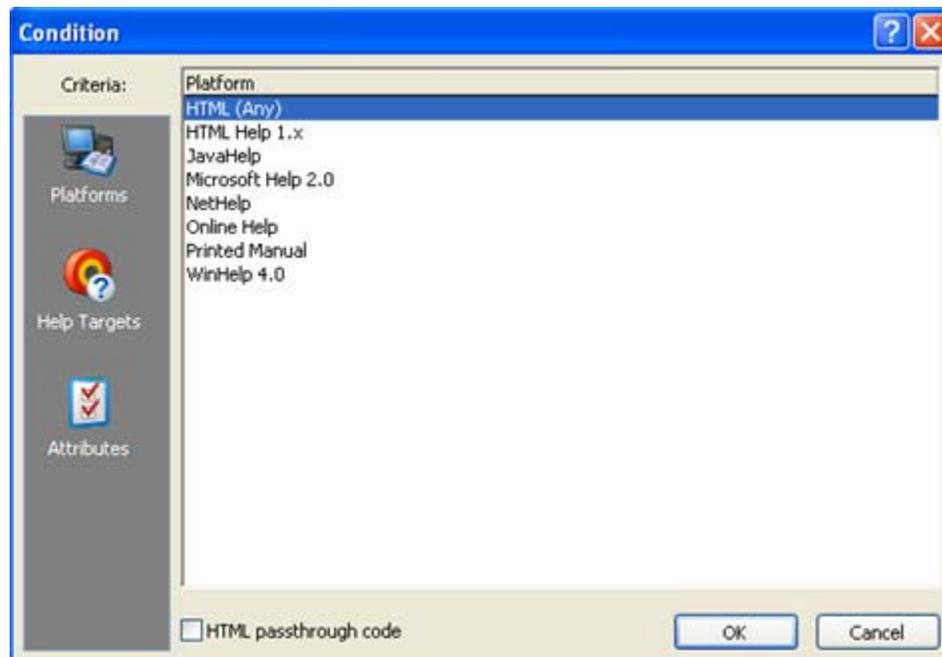
While Doc-To-Help Markup Language is the recommended method of specifying conditional text, you can also use the classic, non-D2HML conditional text comments in your Word source documents.

The classic method is different from the D2HML method in the following ways:

- With classic, non-D2HML comments, only one condition can be applied to your conditional text. For example, if you want your text to appear in both JavaHelp and the Printed Manual but not in any other Help targets, there is no way of specifying both conditions. D2HML allows you to apply multiple conditions to the text. See [Applying Multiple Conditions to Text](#) (page 417) for more information.
- The **View Conditional Text** feature can only be used to view conditional text that has been specified using the classic method. This feature has been retained to ensure compatibility with older Doc-To-Help projects.
- D2HML conditional text comments can be created manually, but Word conditional text comments can only be created using Doc-To-Help's **Condition** window.

Adding Word Conditional Text Comments

1. In Microsoft Word, select the text/image range to be converted into conditional text. Select the range as if you were going to copy it to the Clipboard.
2. On the **Doc-To-Help** toolbar, click the **Apply Conditional Text** button . Doc-To-Help displays the following dialog box.



3. In the left hand **Criteria** toolbar of the **Apply Conditional Text** dialog box, choose the **Platforms, Help Targets** or **Attributes** icon as desired.
4. From the list in the dialog box, select the item in which you want to include the highlighted text or image.
5. Click **OK** to apply conditional text in the source document.
6. Note that the selection now includes a comment after the chosen text as in the example below.

This sentence appears in online help targets only. [C:Text1]

You can make the text/image appear in multiple Help targets or attributes by simply performing the steps above for each choice you require.

Example

The following sentences are different for each Help target produced for Doc-To-Help.

[This text appears in the WinHelp Target.]
[This text appears in the HTMLHelp Target.]
[This text appears in the Printed Manual Target.]

In the source document, this was implemented with four separate paragraphs, each of which has a conditional text comment for one target. The following figure shows the appearance of the marked paragraphs in the source document.

[This text appears in the WinHelp Target. [C:Text8]]
[This text appears in the HTMLHelp Target. [C:Text9]]
[This text appears in the Printed Manual Target. [C:Text10]]

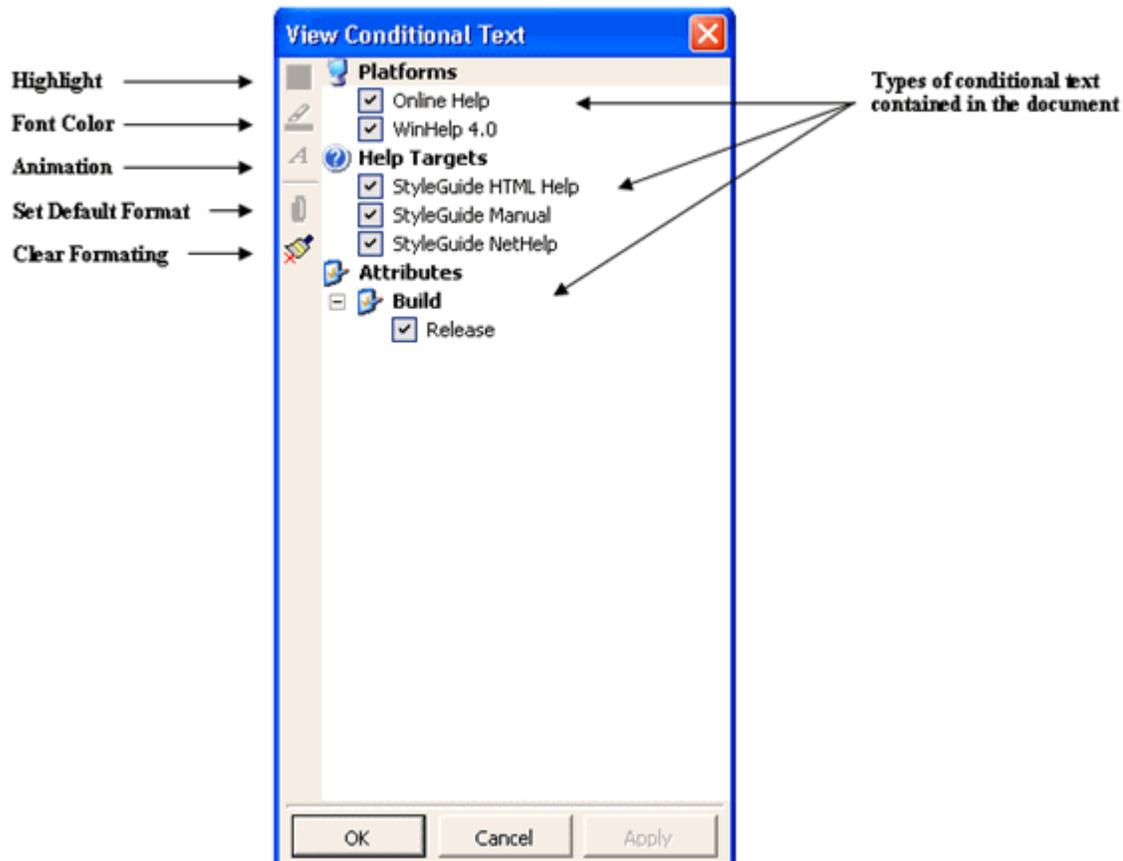
Alternatively, you can perform the same steps with the **Edit | Apply Conditional Text** menu command.

Viewing Classic Doc-To-Help Conditional Text

In some cases, you may wish to limit the types of conditional text you see in your source documents, or modify the text color of a specific type of conditional text. Doc-To-Help 2006 provides you with a feature which allows you to determine which types of conditional text will be visible in your source documents.

Note: This feature can only be used when applying classic Doc-To-Help conditional text. It has been retained to ensure compatibility with older Doc-To-Help projects.

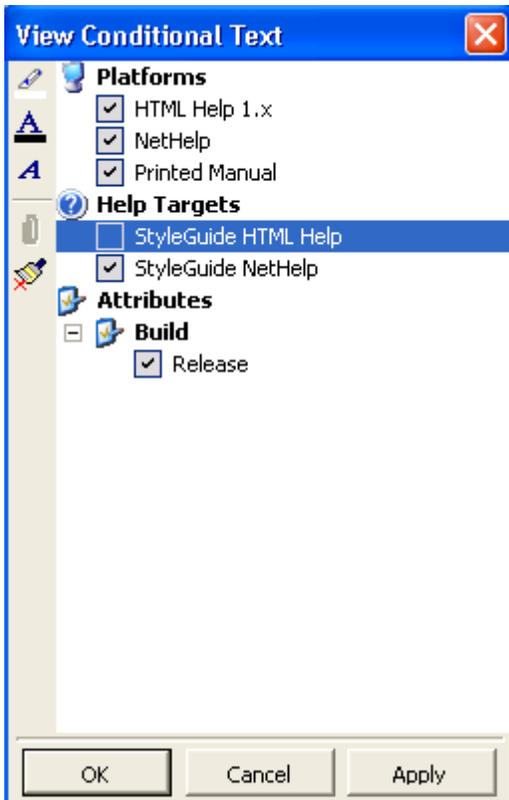
By simply clicking the **View Conditional Text** button  on the **Doc-To-Help** toolbar, a dialog provides you with the ability to select specific conditional text for viewing, modify conditional text colors, add animation and set the default conditional text features. To close the **View Conditional Text** dialog, simply click the **OK** or **Cancel** buttons.



Note: Only checkboxes for the types of conditional text found within the current source document will appear in the **View Conditional Text** dialog.

Selecting Conditional Text Views

The **View Conditional Text** dialog allows you to select which types of conditional text you view in your source documents. This can make complex source documents, with many types of conditional text, easier to read by filtering out those types that you do not need to see. By clearing any of the Conditional Text checkboxes, you can filter which types of conditional text are shown in the currently opened document.



To filter the conditional text in a document:

1. Open the document you want to filter.
2. Click the **View Conditional Text** button. The **View Conditional Text** dialog box opens. The dialog contains check boxes for each type of conditional text formatted within the document.
3. Clear (or check) the check boxes for the conditional text you wish to filter.
4. Click the **Apply** or **OK** button after you are done with your selections.

Doc-To-Help filters your conditional text so that only the types you selected are visible.

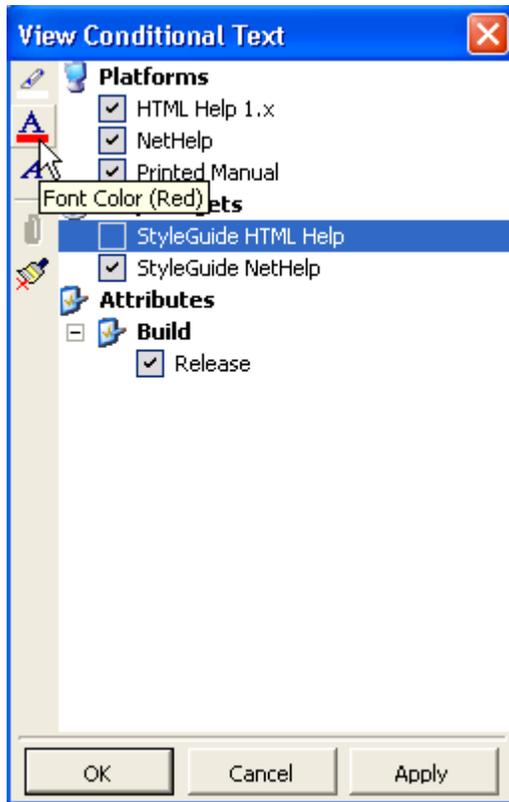
Note: To see the filtering, you must turn off your view hidden text option in Word.

Customizing the Conditional Text Font

The default font color for all conditional text in Doc-To-Help is black. In some cases, where you have a wide range of conditional text types, it may be more convenient to define the text with different colors or even use animation. The **View Conditional Text** dialog contained within Doc-To-Help uses features inherent in Microsoft Word to allow you to modify the text color and use animation to define various conditional text types.

To modify the text definition for conditional text:

1. Open the document you want to customize.
2. Click the **View Conditional Text** button . The **View Conditional Text** dialog opens.
3. Select the conditional text type you want to modify, then click the **Font Color** or **Animation** icon.



4. Select the color or animation type you wish to apply from the appropriate dialog.
5. Click the **Apply** or **OK** button after you are done with your selections.

In this way, you can modify the text color or animation of your conditional text as shown in the example below.

HTML Help ¶

This text is formatted to show up in HTML Help files only. ¶ [CText1] ¶

WinHelp ¶

This text is formatted to show up in WinHelp files only. ¶ [CText2] ¶

Manual ¶

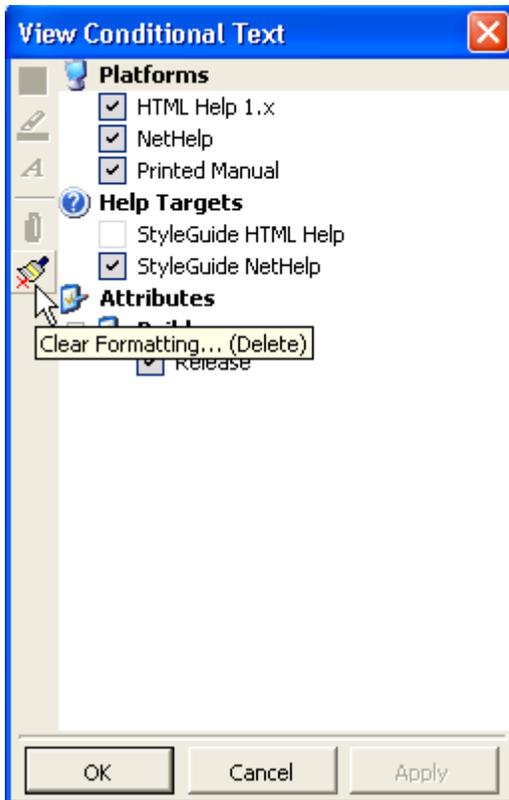
This text is formatted to show up in the Printed Manual only. ¶ [CText3] ¶

Note: Modifications to the conditional text affect the current file only. The font changes do not affect other source documents within the Help project.

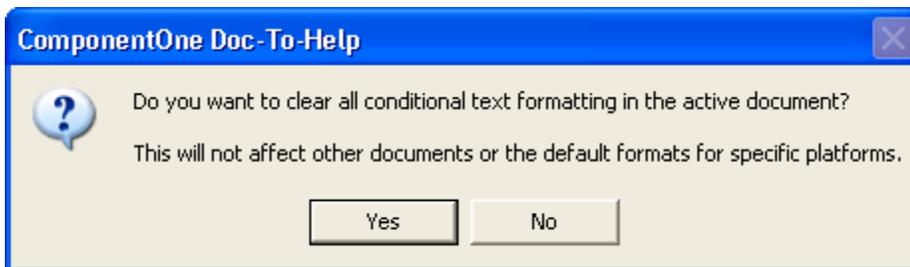
Clearing All Conditional Text Formats

In some cases you may wish to clear the formats you have made to the conditional text in your document. To do this, you simply need to click the **Clear All Formats** icon in the **View Conditional Text** dialog box.

1. Click the **View Conditional Text** button  on the **Doc-To-Help** toolbar. The **View Conditional Text** dialog box opens.
2. Click the **Clear Formatting** icon.



3. Click the **Yes** button to remove all conditional text formatting.

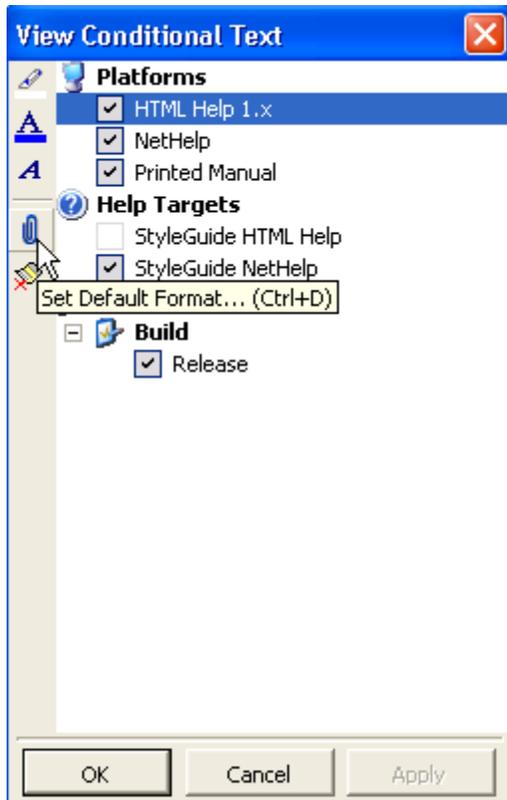


Note: Clearing modifications to the conditional text affect the current document only.

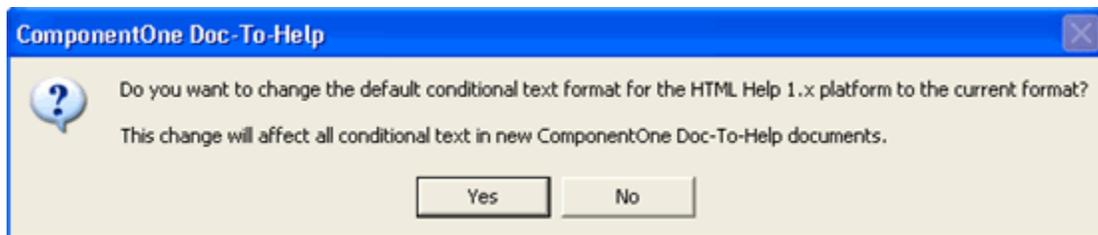
Modifying Conditional Text Default Format

The **View Conditional Text** dialog allows you to customize your conditional text platform styles then set them as the default for all future documents. After you have customized the conditional text for a specific platform, perform the following steps to set the customization as your default for that platform.

1. Click the **View Conditional Text** button  on the **Doc-To-Help** toolbar. The **View Conditional Text** dialog box opens.
2. Select the platform conditional text type you want to set as default, then click the **Set Default Format** icon.



3. Click the **Yes** button to change the default conditional text for that platform.



Note: Changes to the conditional text default settings are performed globally, affecting all future documents produced with Doc-To-Help. These changes will not affect the settings for existing documents including the document that is currently active.

Highlighting Conditional Text

If you are using Microsoft Windows XP or 2003 to edit your source documents, you will notice that the comment text (and therefore Doc-To-Help conditional text) no longer utilizes the yellow background color.

Note: This feature is not supported in Word 2000.

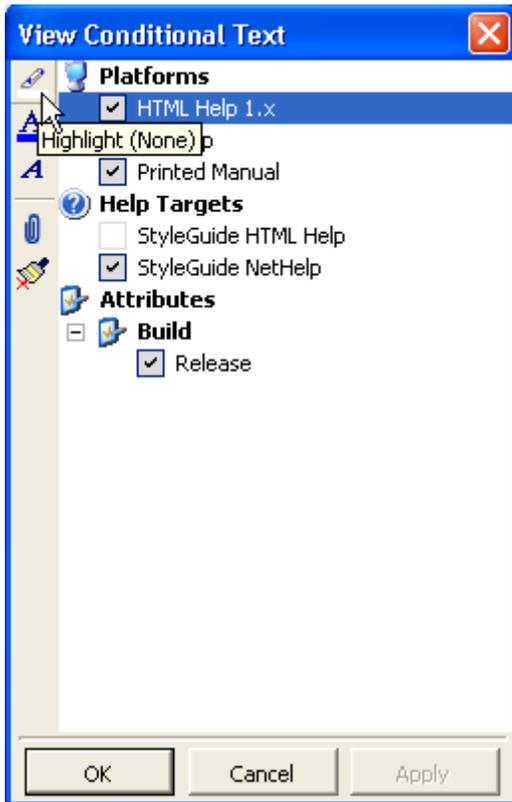
The **View Conditional Text** dialog box allows you to insert a background color behind your conditional text as desired.

To add highlight to your conditional text:

1. Open the document you want to customize.
2. Click the **View Conditional Text** button  on the **Doc-To-Help** toolbar.

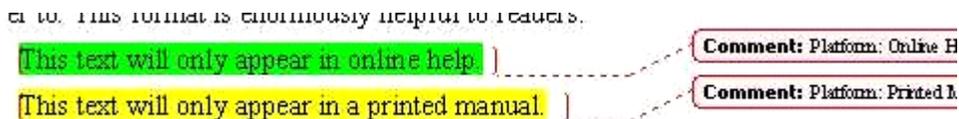
The **View Conditional Text** dialog opens.

3. Select the conditional text type you want to modify, then click the **Highlight** icon.



4. Select which color you want as your background.
5. Click the **Apply** or **OK** buttons after you are done with your selections.

Doc-To-Help inserts the background color into any conditional text of the type you selected.

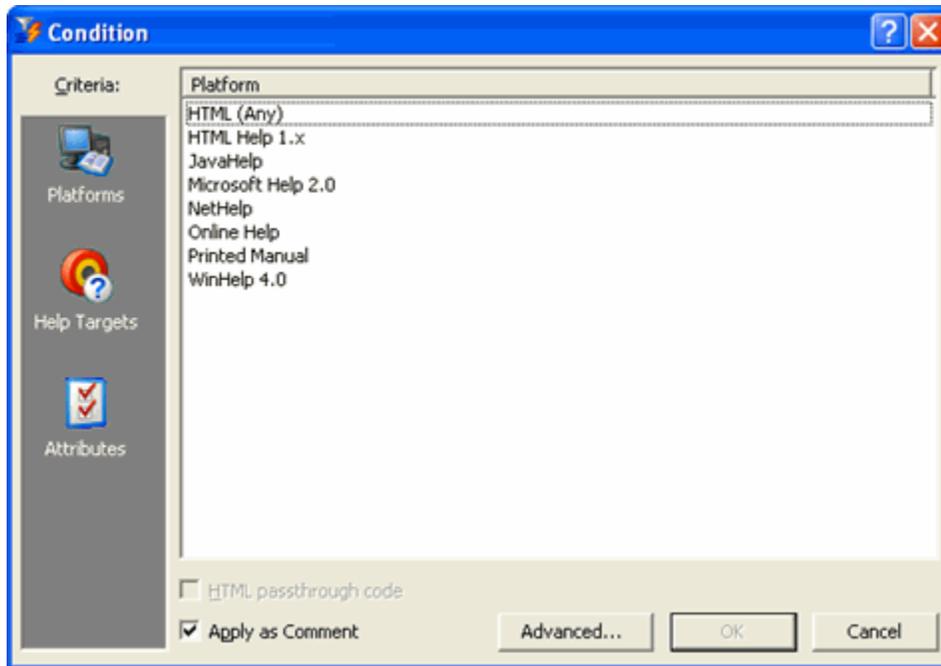


Note: This feature is only functional with Microsoft Word XP.

Using D2HML Conditional Text Styles

Doc-To-Help formats conditional text as a comment, by default, but you can use D2HML styles instead at any time.

1. In your Word source document, select the text/image range to be converted into conditional text. Select the range as if you were going to copy it to the Clipboard.
2. On the **D2HML Styles** toolbar, click the **Conditional Text** button . Doc-To-Help displays the following dialog box.



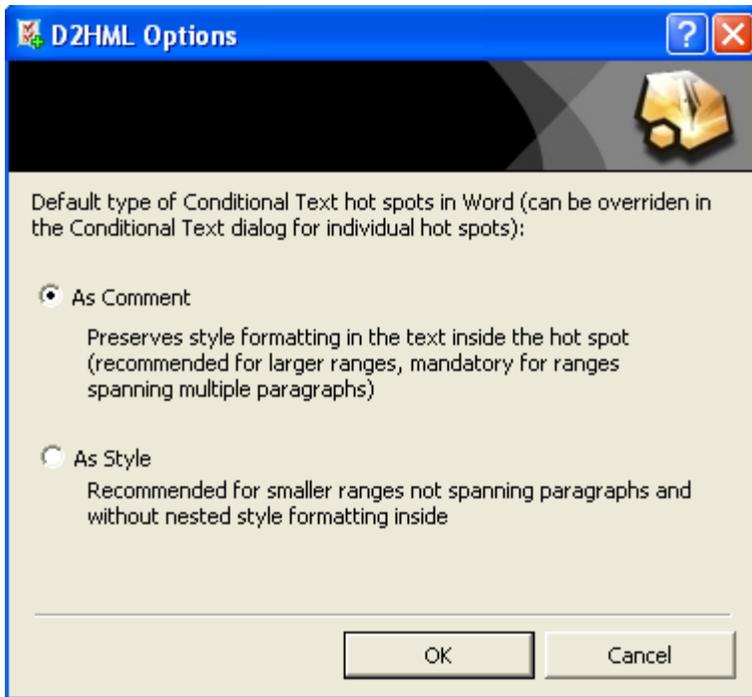
4. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
5. From the list in the right pane, select the item in which you want to include the highlighted text or image.
6. Uncheck the **Apply as Comment** checkbox.
7. Click **OK** to apply the C1H Conditional style to the selection.

You can make the text/image appear in multiple Help targets by applying multiple conditions. See [Applying Multiple Conditions to Text](#) (page 417) for more information.

Changing the Default Type of Conditional Text Hot Spots

If you want conditional text formatted with a D2HML style, by default, instead of a comment, you can change the default setting in the **D2HML Options** dialog box.

1. Click the **D2HML Options** button  on the **D2HML Styles** toolbar in Word. The **D2HML Options** dialog box appears.



2. Select **As Style** to change the default setting and click **OK**. Conditional text is now applied, by default, as the C1H Conditional D2HML style rather than a D2HML conditional text comment.

Note: If there are instances where you want a D2HML comment applied instead of the C1H Conditional D2HML style, check the **Apply as Comment** checkbox in the **Condition** dialog box for each instance.

Removing Conditional Text Comments

In Microsoft Word, right-click the comment and select **Delete Comment** from the shortcut menu or click the **Clear Formatting** button  on the **D2HML Styles** toolbar.

Editing Conditional Text Comments

To edit the condition, select your conditional text and click the **Conditional Text** button  on the **D2HML Styles** toolbar. You can make any changes in the **Condition** dialog box.

You can also edit the comment directly, if necessary. Right-click the conditional text and select **Edit Comment**. Edit the *platform*, *target* or *attribute* properties.

Note: If editing classic Doc-To-Help conditional text, remove the conditional text comment and specify a new condition as described in [Adding Word Conditional Text Comments](#) (page 405).

Using the HTML Passthrough Code Option

The Doc-To-Help HTML passthrough feature allows you to include HTML code directly in your document without Word treating the code as text.

Note: HTML Passthrough code is an advanced feature used in Word documents only. HTML Passthrough code should only be used on the rare occasion when it is necessary to insert HTML code into a Word document and an HTML source document cannot be used for this purpose.

To set HTML code as passthrough:

1. In Microsoft Word, select the HTML code to be marked as if you were going to copy it to the Clipboard.
2. On the **D2HML Styles** toolbar, click the **Conditional Text** button .
3. From the **Condition** dialog box, select the HTML based platform of your choice and check the **HTML passthrough code** checkbox.

HTML passthrough code

Note: Doc-To-Help does not validate HTML passthrough code, so you should ensure that it works properly before incorporating it into your project.

A note to users of previous versions of Doc-To-Help:

It is no longer necessary to define your code with the HTML character style; any style may be used for your HTML code.

Inserting Files with HTML Passthrough Code

By using the HTML passthrough feature and the [CopyFolder](#) property you can insert files (like animated gifs or Excel spreadsheets) directly into your Help target. The [CopyFolder](#) property allows you to define a folder to store linked files when using HTML passthrough code.

Note: HTML Passthrough code is an advanced feature used in Word documents only. HTML Passthrough code should only be used on the rare occasion when it is necessary to insert HTML code into a Word document, and an HTML source document cannot be used for this purpose.

To define a CopyFolder:

1. Open the HTMLHelp folder that Doc-To-Help created when you first built your HTML Help target.
2. Create a sub-folder within the HTMLHelp folder.

In this example, we will use **Media** as the name of the folder.

Note: In this example, we placed this [CopyFolder](#) within the HTMLHelp folder, but there is no limitation on the location of a [CopyFolder](#). It is a good practice though, to keep the [CopyFolder](#) within the Doc-To-Help project folder or Help target folder.

3. Save the files you want to insert using HTML passthrough to the media folder.
4. From the Doc-To-Help project editor, select **Project** from the Icon Bar.
5. Select the **Help Targets** item in the left hand pane.
6. Select the **HTML Help** target from the right pane.
7. From the Property Pane, select the [CopyFolder](#) property and type the following into the right cell:

```
. \HTMLHel p\Medi a
```

To set HTML code as passthrough:

1. Select the location for your HTML code and type it directly into your source document.

For Example:

```

<a href="MyExcel Spreadsheet. xl s">
```

Even though you have saved the file you want to insert to a sub-folder under your HTMLHelp folder, you do not need to include a path statement in your tag. This is because you set the [CopyFolder](#) property. The file will now be automatically be copied to the HTMLHelp Folder.

Note: Make sure that Word smart quotes are turned off so that straight quotes are included in the tags.

2. Select the HTML code and click the **Conditional Text** button  on the **D2HML Styles** toolbar.
3. From the **Condition** window, select the HTML based platform of your choice and check the **HTML passthrough code** checkbox.

HTML passthrough code

4. Rebuild your HTML Help target and check your help file for the inserted file.

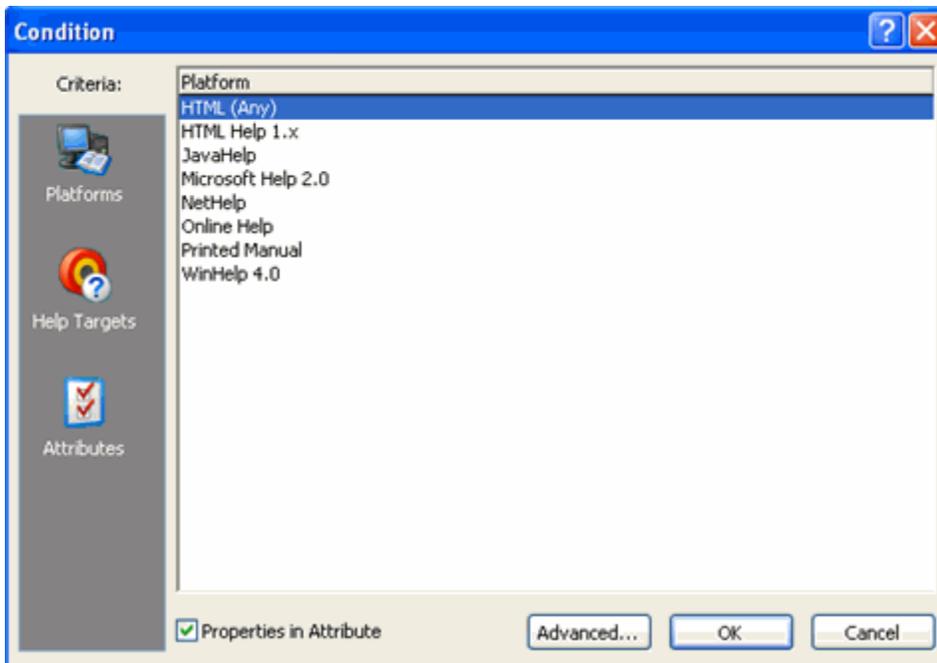
For more information, see [Using the HTML Passthrough Code Option](#) (page 414).

Working with Conditional Text in HTML Documents

This section explains how to work with conditional text in HTML source documents.

Applying Conditional Text in HTML Documents

1. In Microsoft FrontPage or Dreamweaver, select the text/image range to be converted into conditional text. Select the range as if you were going to copy it to the Clipboard.
2. On the **D2HML Styles** toolbar, click the **Conditional Text** button . Doc-To-Help displays the following dialog box.



3. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
4. From the list in the dialog box, select the item in which you want to include the highlighted text or image.

Note: The **Properties in Attribute** is selected by default; therefore, the D2HML properties are stored in the D2HML attribute and are not visible in the source document. If you uncheck **Properties in Attribute**, the D2HML properties appear in the source document. For more information, see [Using the D2HML Attribute in HTML Source Documents](#) (page 244).

- Click **OK** to apply the C1H Conditional style to the selection.

This sentence appears in online help targets only.

You can make the text/image appear in multiple Help targets by applying multiple conditions. See [Applying Multiple Conditions to Text](#) (page 417) for more information.

Note: If you are using an HTML editor other than FrontPage or Dreamweaver, specify a secondary part and apply the C1H Conditional style to it and the desired text or graphic. See [Doc-To-Help Markup Language \(D2HML\)](#) (page 233) for more information.

Removing Conditional Text

In Microsoft FrontPage or Dreamweaver, click the **Clear Condition** button  on the **D2HML Styles** toolbar. If you are using another HTML editor, set the style to None or remove the style's HTML tags.

Editing Conditional Text

To edit the condition, select your conditional text and click the **Conditional Text** button  on the **D2HML Styles** toolbar. You can make any changes in the **Condition** dialog box.

You can also edit the D2HML properties. If the D2HML properties are specified using the D2HML attribute, edit the *platform*, *target* or *attribute* properties in the attribute.

```
<span class=C1HConditional d2hml="platform=manual">recipe database</span>
```

If the D2HML properties are specified in the text, edit the *platform*, *target* or *attribute* properties which are visible in the source document.



Equipment

Equipment

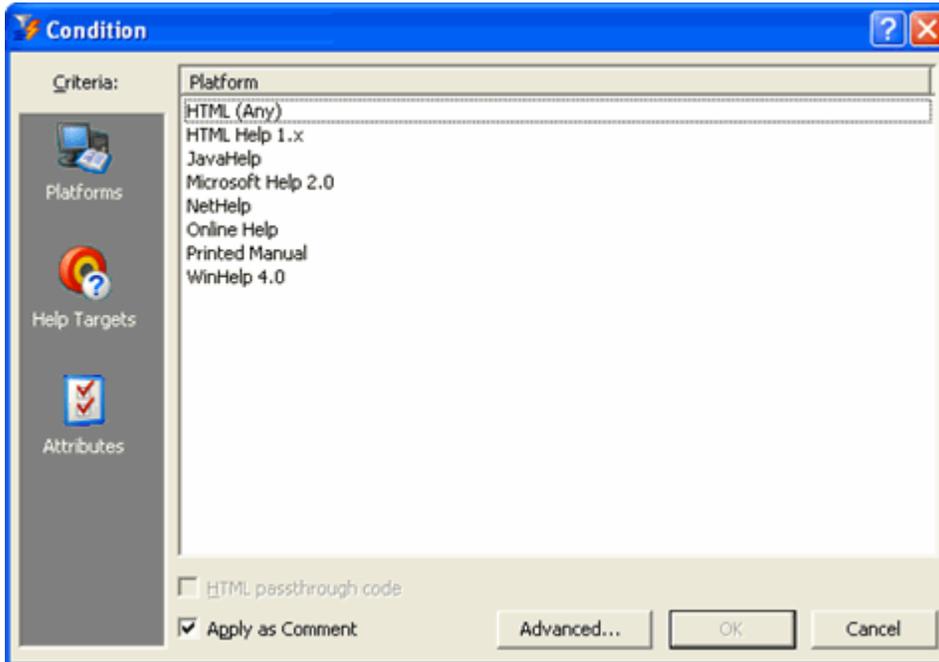
Scuba diving is a sport that requires a lot of equipment.
|platform=manual

For more information, see [Using the D2HML Attribute in HTML Source Documents](#) (page 244).

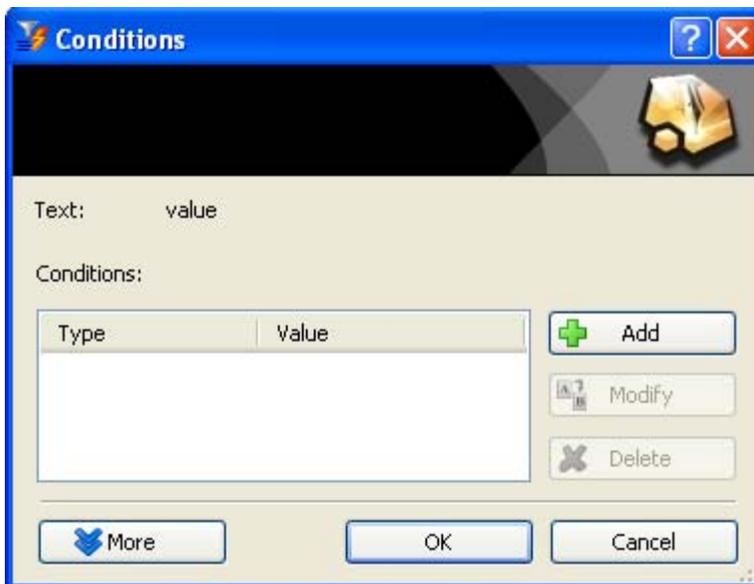
Applying Multiple Conditions to Text

Doc-To-Help has an advanced option when applying conditional text using the **D2HML Styles** toolbar. It allows you to apply multiple conditions in both Word and HTML source documents. To use this option:

1. Select the text/image range to be converted into conditional text and click the **Conditional Text** button  on the **D2HML Styles** toolbar. The **Condition** dialog box appears.



2. Click the **Advanced** button . The **Conditions** dialog box appears.



3. Click the **Add** button. The **Condition** dialog box reappears.
4. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
5. From the list in the dialog box, select the item in which you want to include the highlighted text or image.
6. Click **OK**.
7. Click the **Add** button again, and repeat steps 4-6 to add another condition.

- Click **OK** in the **Conditions** dialog box, and the conditions are applied to the selected text or image.

Manually Applying Multiple Conditions

You can also apply multiple conditions to text by manually editing the conditional text comments or styles.

Using D2HML Comments in Word Documents

If you applied the multiple conditions using a D2HML comment, right-click the comment and select **Edit**.

- Join multiple conditions of the same criterion with a plus (+) sign. For example:

Comment [D2HML3]	Doc-To-Help
D2HML\C1H Conditional attribute=Skill:Expert+Skill:Administrator	

Comment [D2HML4]	Doc-To-Help
D2HML\C1H Conditional platform=htmlhelp+manual	

- Join multiple conditions with different criteria with a semicolon (;). For example:

Comment [D2HML6]	Doc-To-Help
D2HML\C1H Conditional attribute=Skill:Expert;target=StyleGuide NetHelp	

Using D2HML Styles in Word or HTML Documents

If you applied multiple conditions using a D2HML style, edit the hot spot properties.

- Join multiple conditions of the same criterion with a plus (+) sign within the style-formatted text:

This text is assigned the Expert and Administrator attribute.
|attribute=Skill:Expert+Skill:Administrator

This text appears in the HTML Help and Printed Manual platforms.
|platform=htmlhelp+manual

- Join multiple conditions with different criteria with a semicolon (;). For example:

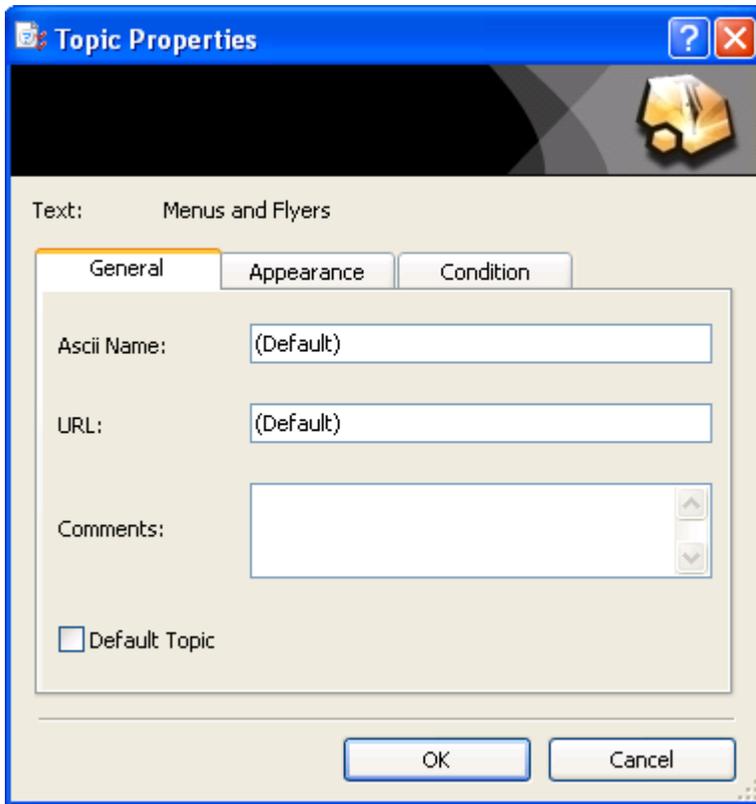
This text appears in the HTML Help platform and
StyleGuide Manual target |platform=htmlhelp;target=StyleGuide Manual

Applying Conditional Text to a Topic

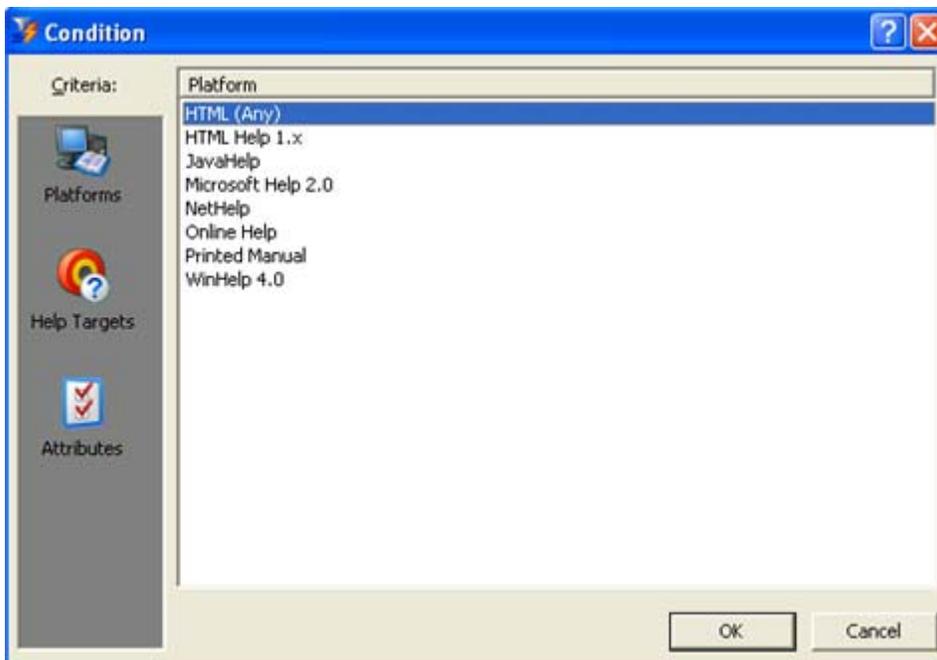
Doc-To-Help also gives you the option of applying conditions to entire topics in both Word and HTML source documents.

Using D2HML

- Click anywhere within the topic (or on the page, if using a single topic document) to be converted into conditional text and click the **Topic Properties** button  on the **D2HML Styles** toolbar.



2. Click the **Condition** tab and then click the **Add** button  **Add**. The **Condition** dialog box appears.



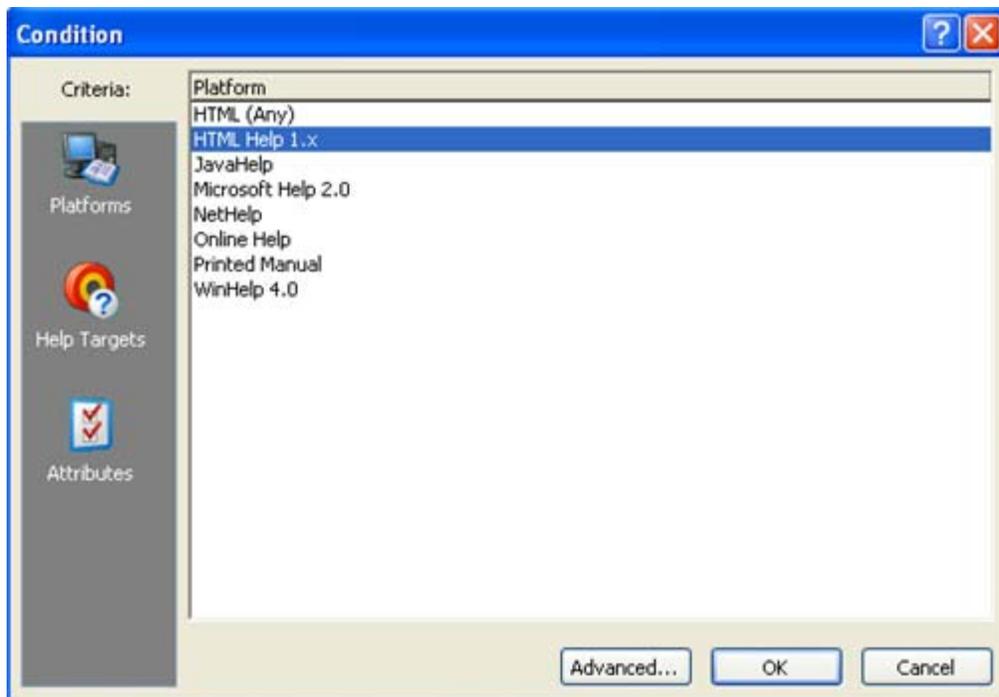
3. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
4. From the list in the dialog box, select the item in which you want to include the topic.
5. Click **OK** to close the **Condition** window and click **OK** again to close the **Topic Properties** dialog box.

Using the Project Editor

1. Open your Doc-To-Help project.
2. On the Icon bar, click **Topics**.
3. Select the **All Topics** item from the left pane.
4. In the right pane, choose the topic to which you would like to apply a condition.
5. Click the **ellipsis** button next to the [Condition](#) property in the Property Pane. The **Condition** dialog box appears.
6. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
7. From the list in the dialog box, select the item in which you want to include the highlighted text or image.
8. Click **OK** to apply conditional text to the topic.

Applying Conditional Text to a Document

1. Open your Doc-To-Help project.
2. On the Icon bar, click **Project**.
3. Select the **Documents** item from the left pane.
4. In the right pane, choose the document to which you would like to apply a condition.
5. Click the **ellipsis** button next to the [Condition](#) property in the Property Pane. The **Condition** dialog box appears.



6. In the left hand **Criteria** toolbar of the **Condition** dialog box, choose the **Platforms**, **Help Targets** or **Attributes** icon as desired.
7. From the list in the dialog box, select the item in which you want to include the document.
8. Click **OK** to apply conditional text to the document.

Using Attributes

Attributes provide the Help author with the ability to create conditional text based upon author specified criteria. In some instances, you may wish to include specific text in one group of Help files while having alternate text included in a separate group of Help files. By creating attributes and associating the attribute values with Help targets, you can determine which targets contain which text.

For example, you might create an attribute grouping named **Skills** that contains the values *Novice*, *Expert* and *Administrator*. In this way, you can modify the text contained in Help files for each audience by simply attaching the value to a given group of Help targets and formatting the conditional text with those attributes values.

There are three simple steps for using attributes:

1. In the Doc-To-Help project editor, [add a new attribute](#) (page 423) with values to your project.

Name	Value
(New)	
Build	Internal
Build	Release
Skill	Administrator
Skill	Expert
Skill	Novice

2. [Insert conditional text](#) (page 423) in your Word or HTML source document using attributes.

This sentence is not assigned an attribute.

[This sentence is assigned the Novice attribute.]D2HML2]

[This sentence is assigned the Expert attribute.]D2HML3]

[This sentence is assigned the Administrator attribute.]D2HML4]

3. [Associate the attribute value](#) (page 424) with Help targets in the Doc-To-Help project editor using the **View Attributes** button or using the **AttributeExpression** property. You can view a list of existing attributes at any time by selecting the **Attributes** item in the project editor tree list.

The screenshot shows the 'Attributes' window in the Doc-To-Help project editor. The window is divided into two main sections. On the left is a tree view showing the project structure, with 'Attributes' selected. On the right is a table of attributes and a detailed view for the selected 'Skill' attribute.

Name	Value
(New)	
Build	Internal
Build	Release
New Attribute	Internal
Skill	Administrator
Skill	Expert
Skill	Novice

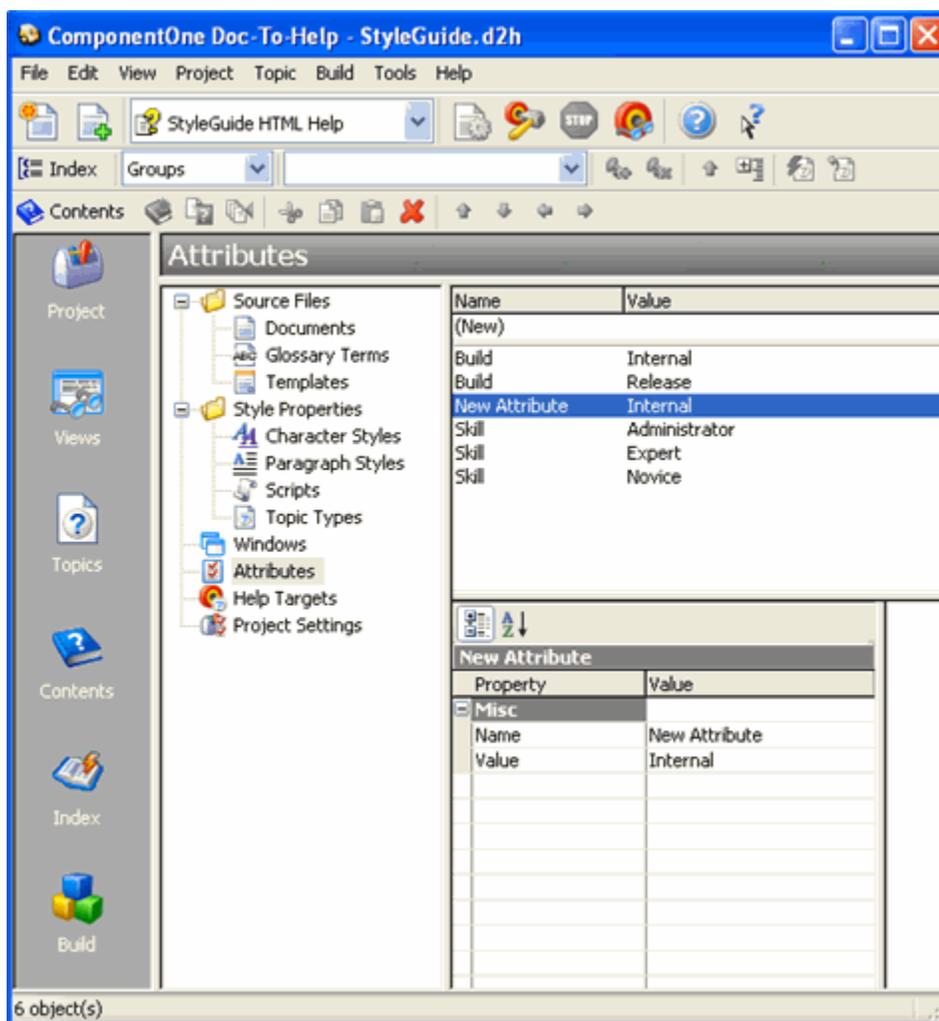
Property	Value
Skill	
Misc	
Name	Skill
Value	Expert

The list of attributes appears in the right pane. By choosing any one of the attributes, the property pane displays the property settings for that attribute.

Adding a New Attribute

To add a new attribute to the default list:

1. Click the **Project** icon and select the **Attributes** item on the topic icon project editor tree list.
2. Right-click anywhere in the attributes list pane on the right and choose **New** from the shortcut menu. Note that the focus of your cursor has moved to the top of the attribute list window.
3. Type the name of the new attribute and press **Enter**.
4. Enter a value for the attribute and press **Enter**. Note that the new attribute has been added to the list.
5. To add another value to the new attribute, simply perform steps 1 through 4 using the same attribute name but a new value. In this way you can have several attributes with the same name but different values.



Alternatively, you can add a new attribute by clicking the **Project** menu, choosing **New**, then clicking **Attribute**.

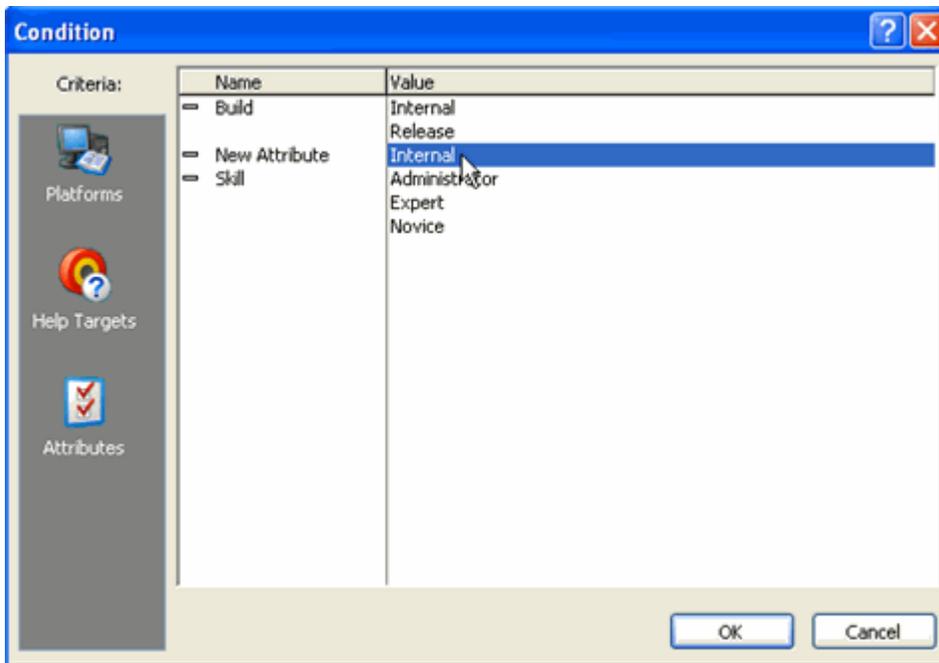
Inserting Conditional Text Using Attributes

In some cases you may have two Help files using the same Help target, but with somewhat different text. To do this, you can set the conditional text by attribute.

1. In your source document, highlight the text that you wish to limit by attribute and click the **Conditional Text** button on the **D2HML Styles** toolbar.

Note: You can also use the **Apply Conditional Text** button on the **Doc-To-Help** toolbar to format conditional text, but this method has been retained for compatibility with older projects and is not the recommended method. See [Using Classic Doc-To-Help Conditional Text](#) (page 405) for more information.

2. In the **Criteria** pane, choose the **Attributes** icon.
3. From the **Value** column, choose the attribute that you want to associate with the highlighted text.
4. Click **OK**.



5. Save and close the source document.

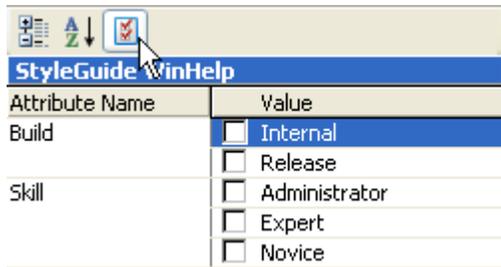
If you would like to associate text with more than one attribute, see [Applying Multiple Conditions to Text](#) (page 417) for more information.

Associating the Attribute Value with Help Targets

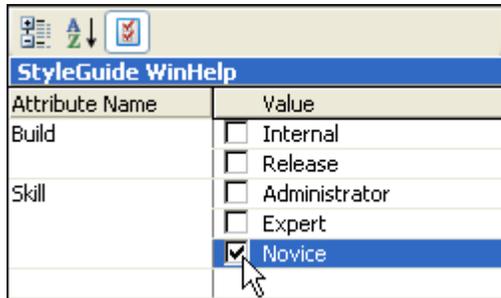
You can associate attribute values with Help targets using the **Show Attributes** button. You simply select the attribute and its value to associate with the Help target. However, in some cases you may need to use an advanced conditional build expression for a Help target. For example, suppose you have a **Skill** attribute with values *Novice*, *Expert* and *Administrator*. You may want specific text to appear in the *Expert* or *Administrator* Help files, but not in the *Novice* Help files. The [AttributeExpression](#) property allows you to use any combination of attributes with the logical operators OR, AND and NOT. In this case, you would enter **NOT Skill:Novice OR Skill:Expert OR Skill:Administrator** in the [AttributeExpression](#) property.

Using the Show Attributes Button

1. In the project editor, select your Help target from the **Help Target** drop-down list and click the **Show Attributes** button  in the property pane. The **Attributes** tab appears.

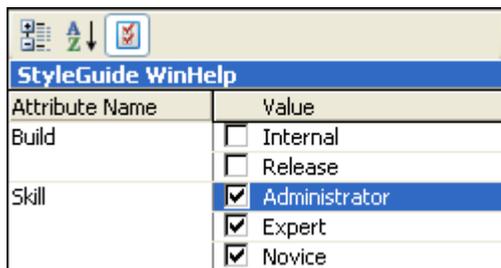


- Apply the desired content to the Help target by selecting an attribute value or multiple values.



In this example, any conditional text with the attribute value *Novice* applied appears in your compiled Help file.

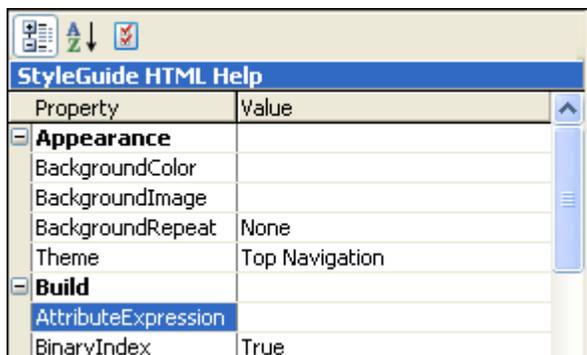
- By simply changing your choice of attribute values and recompiling, you can modify the contents of your Help file.



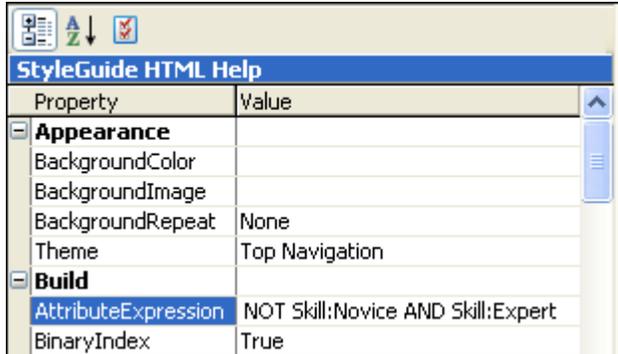
In this example, any conditional text with the attribute value *Administrator*, *Expert* or *Novice* applied appears in the compiled Help file.

Using the AttributeExpression Property

- In the project editor, select your Help target from the Help target drop-down list and select the [AttributeExpression](#) property in the property pane.



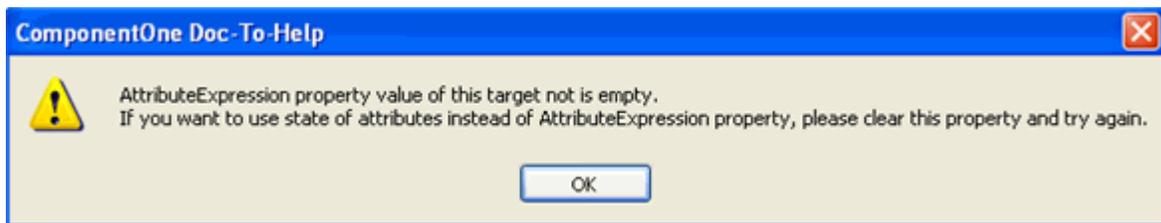
2. Enter any combination of attributes using the logical operators **OR**, **AND**, and **NOT**.



Property	Value
Appearance	
BackgroundColor	
BackgroundImage	
BackgroundRepeat	None
Theme	Top Navigation
Build	
AttributeExpression	NOT Skill:Novice AND Skill:Expert
BinaryIndex	True

In this example, only the conditional text with the attribute value *Expert* applied is included in the Help target. Conditional text with attribute value *Novice* applied does not appear in the target.

Note: If a condition is entered in the [AttributeExpression](#) property, clicking the **Show Attributes** button results in a warning asking you to clear the [AttributeExpression](#) property to use the **Attributes** tab. Click **OK** and clear the [AttributeExpression](#) property before clicking the **Show Attributes** button again.



3. By changing your condition in the [AttributeExpression](#) property, you can quickly modify the contents of the help file. For example, change the [AttributeExpression](#) property to **Skill:Administrator AND Skill:Expert OR Skill:Novice**. Only conditional text with both attribute values *Administrator* and *Expert* applied or with the attribute value *Novice* applied appears in the Help target. See [Applying Multiple Conditions to Text](#) (page 417) for more information on applying multiple attribute values.

Using Modular Help

Modular Help development breaks the creation of a large documentation project down into a set of smaller component pieces. Each component—containing, perhaps, one chapter of the overall project—becomes a separate document and Help file. The components are linked or merged together into a seamless single system, giving the end user appearance and functionality indistinguishable from a single large Help file.

In the context of ComponentOne Doc-To-Help 2006, each component project in a modular Help system is a separate Doc-To-Help project, with its own folder, document file(s), and so forth, creating its own Help file. A separate hub project links the components together.

When you decide to take a modular approach with a Help system, you don't want to lose any functionality. You want to be able to generate the same high quality WinHelp and HTML Help systems as you would with a single project. Following the instructions in this chapter will ensure this. Before using modular Help, be aware that modular Help is Help target specific and you cannot produce a printed manual from a modular Help project.

Note: Do not confuse modular development with a Doc-To-Help multifile project. In a multifile project, multiple .DOC Word document or .HTM HTML document files are compiled together into one Help file. In a modular project, there are different source documents, different Doc-To-Help projects, and different Help files, which are then made to appear to the user as one file.

Managing Your Modular Help System

Before beginning a modular Help project (or any project of a similar size and scope, for that matter), it's wise to take a step back and consider what the entire project will entail. With this perspective, you'll be better able to plan the project.

What's in a Modular Help System

Your modular Help system will consist of a number of modules or components, created from a number of Doc-To-Help projects. In addition to these component projects, the linking or merging of components in a modular Help project is done using what we shall call the hub project (generally one for each platform), which are primarily responsible for tying the modules together into an apparent whole.

Module Files:

- For WinHelp, each component is a separate WinHelp .HLP file and a WinHelp .CNT contents file. Each matching set of .HLP and .CNT files is the product of a separate Doc-To-Help project.
- For compiled HTML Help, each component is a separate HTML Help .CHM file. Just like with WinHelp, each .CHM file is the product of a separate Doc-To-Help project.

Hub Files

- For WinHelp, the hub is the WinHelp .HLP file and the WinHelp .CNT contents file.
- For compiled HTML Help, the hub is another compiled HTML Help .CHM file.

Note: When working with HTML Help hub files, there is a limitation to the naming convention that is unrelated to Doc-To-Help. Using spaces when naming the hub .CHM files will disable the next/previous functionality of the modular Help system. It is therefore advisable to eliminate spacing in your .CHM file names wherever possible. If your .D2H project file contains spaces (and, be default your .CHM file also contains spaces) simply remove the spaces from the [BaseName](#) property of the Help target.

In the examples in this documentation, we often describe the files as the hub project or hub Help file. You do not have to name your project hub, and in fact probably shouldn't. A better choice would be a descriptive name that reflects the content and purpose of your Help system.

Similarly, in the module examples, we describe files as module document or module Help file. Again, except for illustrative purposes in discussions of modular Help development, these are poor choices for file names—a descriptive and specific name is much better.

File Organization

There is no single right way to organize projects and files when you are developing modular Help systems. However, a good system of organization makes management easier and production less error-prone. The following suggestions are based on techniques and standards that have proven themselves again and again.

Create a folder for the development files for the entire system. Give each component project its own subfolder underneath the overall folder. The hub document should also have a separate subfolder.

If a team of several authors will be working on the system, you will probably want to put the project on a network drive. If you are working alone, you can put it on a local drive, but in either case, be sure it gets backed up regularly.

Your project will need a folder in which you can collect, collate, and test the total Help system. This folder needs only the files which will be distributed to end users, such as the .HLP, .CNT or .CHM files. It does not need the component project folders or their contents.

If end users on your network need to access your modular Help system, it's best to have another folder somewhere with the user-accessed copy of the Help. This allows you to develop and test the Help system, and your users to access the information they need, without interfering with one another.

Project Standards

In order for your modular project to mesh seamlessly, it's important that all the components adhere to a set of standards. Among the things you should consider standardizing are:

- Which templates or cascading style sheets your documents and help files are based on
- Which styles are used, for what, and how
- Style customizations
- Typographical conventions
- Help window designs and usage
- Indexing
- Frequency and type of hypertext jumps
- Use of graphics
- Tone and style of writing

You should also ensure that, if you are using a centralized glossary, authors do not attempt to add terms to their local glossary, but instead place their terms in the central glossary.

Creating a Modular Help Project for WinHelp

1. Start by creating your module Help projects as you normally would.
2. Create a separate project to use as the hub Help project.
3. Highlight the **Project Settings** item in the left pane of the project editor.
4. Set the [ModularHub](#) property to **True**.

ModularHelp.d2h	
Property	Value
Localization	
OnPage	on page
Misc	
DefaultTemplate	C1H_NORM.DOT
DocumentType	Mixed
Glossary	WordDocuments\Glossa
HTMLDefaultCSS	C1H_Source_full.css
HTMLSourceFolder	HTMLDocuments
MapNumberOffset	0
ModularHub	True
URLMode	True
WordSourceFolder	False

- Open the hub project document and create placeholder topics for the each of the module Help files you wish to associate with the hub.

The placeholder topic text is generally formatted as Heading 1 topics, but you can use other heading levels as desired.

- Click the **Make Target** button to build the hub WinHelp file.
- Click the **Topics** Icon.
- Highlight one of the placeholder topics and enter the corresponding module hlp Help file to the right of the [MergeFile](#) property.
- Enter the corresponding module .cnt Help file to the right of the [MergeContents](#) property.

Note that you can use the [MergeTitle](#) property to differentiate between like named Help topics.

Chapter 1	
Property	Value
URL	chapter1.htm
WinHelpMacro	
Modular	
MergeContents	Xarray.cnt
MergeFile	Xarray.hlp
MergeTitle	
Natural Search	
SearchEnabled	True

- Build the hub project again.
- Copy the module Help files (.hlp and .cnt) into the hub Help file output directory.
- Click the hub Help file **View Target** button.

Note that the contents pane includes all the hub Help files.

- Click the **Index** tab in the Help file.

Note that the index list has been merged to include index keywords from all the module Help files.

Note: A full rebuild (Rebuild Target command) deletes all existing files in the Help file output directory. If you do a full rebuild, be sure to copy the module Help files back into the hub output directory.

Modifying the Context IDs in your Modular Projects

To avoid numbering conflicts when using automatic context IDs, customize the context ID numbers in each module project with the [MapNumberOffset](#) property. For more information, see [Context Sensitive Help](#) (page 297) and [Customizing your Automatically Generated Context ID Numbers](#) (page 300).

WinHelp Module Glossaries

Doc-To-Help includes a glossary feature that automatically generates pop-up definitions throughout your Help, based on glossary definitions. If you would like to use this feature with modular Help, each component Help project (hub and module projects) will need to have a duplicate glossary.doc as part of the project. To eliminate the duplicate glossaries contained in the module Help projects, from the hub modular help table of contents, perform the following procedure on all **module** Help projects:

1. Open one of the module projects.
2. Select the **Project** icon.
3. In the left pane, select **Help Targets**.
4. From the right pane, select the Help Target you wish to modify.
5. From the property pane, set the [SkipGlossary](#) property to **True** and press **Enter**.
6. Click the **Make Target** button.
7. Copy the compiled module Help files (.hlp and .cnt) into the hub Help file output directory, overwriting the previous version.
8. Perform the same steps to remove the glossary contents from all of the module Help projects.

For more information, see [Creating a Glossary](#) (page 333).

Browse Footnotes

Browse footnotes (+) are not emitted for placeholder topics, and these topics do not participate in the browse sequence. However, placeholder topics are **not** excluded from automatic hierarchical jumps. Depending upon the structure of the hub file, to exclude the placeholder topics from automatic jumps, it may be necessary to set the [RelatedJumps](#) property to **None** for non-placeholder topics.

For example, if a hub project contains a main Heading 1 topic with placeholder topics at the Heading 2 level, then the Heading 1 topic will have hierarchical jumps to the placeholder topics. The reason for this is that the Heading 1 paragraph style's [TopicType](#) property is set to **Contents** and the **Contents** Topic Type has the [AutoNavigate](#) property set to **True**. In this case, you should set the [RelatedJumps](#) property to **None** for the Heading 1 topic to eliminate the automatic jumps.

Referencing Absent CNT Files

If your hub .CNT file contains references to module .CNT files that are not found on the user's computer (either in the current folder or in the WinHelp folder), don't worry; WinHelp won't display any error messages and everything will work fine. The headings and topics referenced by the missing .CNT files won't be displayed.

The same is true for files included in the index. If one of the Help files you specify is not found on the user's computer, its keywords are simply not included, without any errors displayed to the user.

This means you can include references to files that don't yet exist, but will exist when you send an updated version of your Help system or if someone purchases an additional module to your program. For example, let's say a user purchases a program from you. Your program's main .CNT file references ADDON.CNT which was not included with your program. Several months later, the user purchases the add-on module which comes with ADDON.CNT and ADDON.HLP. Assuming your add-on's setup program regenerates the main Help file, ADDON.CNT and ADDON.HLP will automatically become part of the Help system.

What's in a CNT File

A simple .CNT file that references other Help files and .CNT files is shown below.

```
main.cnt - Notepad
File Edit Search Help
:Base main.hlp
:Index Audit Results=Audit.hlp
:Index Methodology=method.hlp
1 Introduction
2 Overview=Overview1
2 Results=Results1
1 Results
2 East Coast Finding=EastCoastFinding
2 West Coast Findings=WestCoastFindings
1 Projections
2 Overview=Overview2
2 The Next Two Years=TheNextTwoYears
:include method.cnt
:include audit.cnt
```

The line

```
: Base mai n. hl p
```

indicates that MAIN.HLP is the main Help file. The lines

```
: I ndex Audi t Resul ts=audi t. hl p
: I ndex Methodol ogy=method. hl p
```

indicate that AUDIT.HLP and METHOD.HLP should be included in the keyword search, full-text search and in K-Links. The lines

```
1 I ntroducti on
2 Overvi ew=Overvi ew1
2 Resul ts=Resul ts1
1 Resul ts
. . .
2 The Next Two Years=TheNextTwoYears
```

determine the outline that appears in the Contents tab. The lines

```
: i ncl ude method. cnt
: i ncl ude audi t. cnt
```

indicate that METHOD.CNT and AUDIT.CNT should be appended to the current outline.

For a modular system, the component Contents files (generated automatically by Doc-To-Help during the Build Online Help process) contain the topic lines that determine the Contents tab outline. The hub Contents file contain :Include and :Index statements.

Creating a Modular Help Project for HTML Help

1. Start by creating your module Help projects as you normally would.
2. Set the [BinaryTOC](#) and the [BinaryIndex](#) properties of the module Help projects to **False** and build the HTML Help targets.

ModularHelp HTML Help	
Property	Value
Theme	(Default)
Build	
BinaryIndex	False
BinaryTOC	False
GenerateCSS	True

3. Create a separate project to use as the hub Help project.
4. Highlight the **Project Settings** item in the left hand pane of the project editor.
5. Set the [ModularHub](#) property to **True**.

ModularHelp.d2h	
Property	Value
Misc	
DefaultTemplate	C1H_NORM.DOT
DocumentType	Mixed
Glossary	WordDocuments\Glossa
HTMLDefaultCSS	C1H_Source_full.css
HTMLSourceFolder	HTMLDocuments
MapNumberOffset	0
ModularHub	True
URLMode	True
WordSourceFolder	False

6. Open the hub project document and create placeholder topics for the each of the module Help files you wish to associate with the hub.

The placeholder topic text is generally formatted as Heading 1 topics, but you can use other heading levels as desired.

7. Click the **Make Target** button to build the hub HTML Help file.
8. In the hub project, choose the **Help Target** item from the left pane and the **HTML Help** target from the right pane of the project editor.
9. Set the [BinaryTOC](#) property to **False** and the [BinaryIndex](#) property to **True**.
10. Click the project editor **Topics** Icon.
11. Highlight one of the placeholder topics and enter the corresponding module Help file chm to the right of the [MergeFile](#) property. Enter the corresponding module Help file hhc to the right of the [MergeContents](#) property. (The hhc file name is normally the same as the chm file name.)

Topic 1	
Property	Value
URL	topic1.htm
WinHelpMacro	
Modular	
MergeContents	Xarray.hhc
MergeFile	Xarray.chm
MergeTitle	

12. Build the hub project again.
13. Copy the module Help files (.chm) into the hub Help file output directory.
Note that you do not have to copy the .hhc files into the hub Help file output directory.

- Click the hub Help file **View Target** button. The contents pane includes all the hub Help files. Click the **Index** tab; notice that the index list has been merged to include index keywords from all the module Help files.

Note: A full rebuild (Rebuild Target command) overwrites all existing files in the Help file output directory. If you do a full rebuild, be sure to copy the module Help files back into the hub output directory.

Modifying the Context IDs in your Modular Projects

To avoid numbering conflicts when using automatic context IDs, customize the context ID numbers in each module project with the [MapNumberOffset](#) property. For more information, see [Context Sensitive Help](#) (page 297) and [Customizing your Automatically Generated Context ID Numbers](#) (page 300).

HTML Help Module Glossaries

Doc-To-Help includes a glossary feature that automatically generates pop-up definitions throughout your Help, based on glossary definitions. If you would like to use this feature with modular Help, each component Help project (hub and module projects) will need to have a duplicate glossary.doc as part of the project. To eliminate the duplicate glossaries contained in the module Help projects, from the hub modular help table of contents, perform the following procedure on all **module** Help projects:

- Open one of the module projects.
- Select the **Project** icon.
- In the left pane, select **Help Targets**.
- From the right pane, select the Help Target you wish to modify.
- From the property pane, set the [SkipGlossary](#) property to **True** and press **Enter**.
- Click the **Make Target** button.
- Copy the compiled module Help file (.chm) into the hub Help file output directory, overwriting the previous version.
- Perform the same steps to remove the glossary contents from all of the module Help projects.

For more information, see [Creating a Glossary](#) (page 333).

HTML Help Hub Project

The hub project in a modular HTML Help system is a Doc-To-Help project resulting in a compiled HTML Help (.CHM) file that ties the whole system together through special entries in the contents (.HHC) and project (.HHP) files.

We recommend that you keep an HTML Help hub project quite small and restrict it to information that does not change often, such as a welcome screen, directions for using the Help, or company contact information.

Referencing Module Files

A modular HTML Help system's hub file uses two mechanisms to reference the components: Merge tags in the Contents (.HHC) file to combine the components' tables of contents with that of the hub; and [Merge Files] entries in the project (.HHP) file to include the components in the hub HTML Help file's Index and Find features.

Merge Tags

A typical Merge tag looks like this:

```
<OBJECT type="text/si temap">
  <param name="Merge" value="/File.e.chm: /File.e.hhc">
</OBJECT>
```

HTML Help Contents files can be merged even if the HTML Help files are not compiled .CHM files.

[MERGE FILES]

Syntax of the [MERGE FILES] section is very simple; just list the .CHM files, as shown here:

```
[MERGE FILES]
Al pha. chm
Bravo. chm
Charl i e. chm
```

Support for the [MERGE FILES] section is limited to compiled HTML Help (.CHM) files.

Creating Cross-Module Links in HTML Help

In an HTML Help modular Help system, you can create cross-module links, or links from one module topic to a topic in a different module. First, create a group in both the source and destination module projects, and index the destination topic to that group in the destination module project. Then, when the hub project is built and the link in the source module is clicked, it jumps to the topic in the destination module.

For example, suppose we have an HTML Help modular system with a hub project and two module projects that were created following the steps in the [Creating a Modular Help Project for HTML Help](#) (page 431) topic. We can create a link from Module1 to Module2. We'll call Module1 the source module, and Module2 will be the destination module since it contains the topic to which we want to link. Use the following steps:

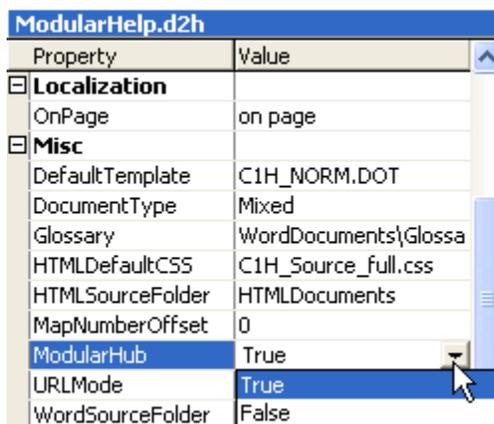
1. Create a group named *Modules* in your Module1 project.
2. In your Module1 source document, [create a group link](#) (page 311) to the *Modules* group.
3. Set the [BinaryTOC](#) and the [BinaryIndex](#) properties of the Module1 project to **False** and build the HTML Help target.
4. Create a group named *Modules* in your Module2 project.
5. Assign the topic you want to link to from Module1 to the *Modules* group.
6. Set the [BinaryTOC](#) and the [BinaryIndex](#) properties of the Module2 project to **False** and build the HTML Help target.
7. Open your hub project and create a Module1 and Module2 placeholder topic for the two modules. The placeholder topic text is generally formatted as Heading 1 topics, but you can use other heading levels as desired.
8. Click the **Make Target** button to build the hub HTML Help file.
9. In the hub project, choose the **Help Target** item from the left pane and the **HTML Help** target from the right pane of the project editor.
10. Set the [BinaryTOC](#) property to **False** and the [BinaryIndex](#) property to **True**. Note that the [ModularHub](#) property should be set to **True** in the **Project Settings**.
11. Click the project editor **Topics** icon.
12. Highlight one of the placeholder topics and enter the corresponding module Help file (.chm) name to the right of the [MergeFile](#) property.
13. Enter the corresponding module Help file (.hhc) name to the right of the [MergeContents](#) property. (The .hhc file name is normally the same as the .chm file name.)
14. Build the HTML Help target.
15. Copy the new module Help files (.chm) into the hub HTMLHelp output directory. Note that you do not have to copy the .hhc files into the hub Help file output directory.
16. Click the hub Help file **View Target** button.
17. Click the group link in your Module1 topic. It jumps to the topic assigned to the *Modules* group from your Module2 project.

Creating a Modular Help Project for NetHelp

Unlike with HTMLHelp, there are no limitations or requirements for module projects when creating a modular Help project for NetHelp. Any NetHelp target built in Doc-To-Help can be used as a module in a modular project. The only requirement is that it must be built with a Doc-To-Help version compatible with the current version. For example, you cannot merge NetHelp projects built with a Doc-To-Help version older than Doc-to-Help 2006.1; you must rebuild them in Doc-to-Help 2006.1.

To create a modular Help project:

1. Start by creating your module Help projects.
2. Create a separate project to use as the hub Help project.
3. In your hub project, highlight the **Project Settings** item in the left pane of the project editor.
4. Set the [ModularHub](#) property to **True**.

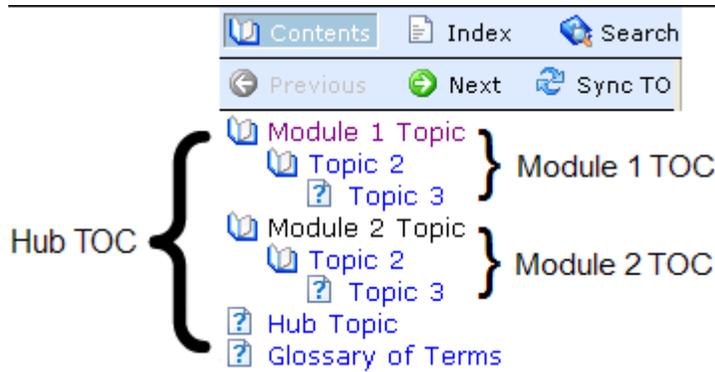


ModularHelp.d2h	
Property	Value
Localization	
OnPage	on page
Misc	
DefaultTemplate	C1H_NORM.DOT
DocumentType	Mixed
Glossary	WordDocuments\Glossa
HTMLDefaultCSS	C1H_Source_full.css
HTMLSourceFolder	HTMLDocuments
MapNumberOffset	0
ModularHub	True
URLMode	True
WordSourceFolder	False

5. Open the hub project document and create placeholder topics for each of the module Help files you wish to associate with the hub. The placeholder topic text is generally formatted as a Heading 1, but you can use other heading levels as desired.
6. Click the **Make Target** button to compile the hub NetHelp files.
7. Click the **Topics** Icon.
8. Highlight one of the placeholder topics and click the ellipsis button next to the [MergeFile](#) property.
9. Locate the **project.npj** file in the module's **NetHelp/ProjectInfo** folder.
10. Select **project.npj** and click **Open**. The file path appears next to the [MergeFile](#) property.
11. Repeat steps 8-10 for all placeholder topics.
12. Rebuild the hub project and click **View Target** button. Doc-to-Help merges the NetHelp modules, including their table of contents, index, search and glossary, with the NetHelp target of the hub project.

NetHelp Hub Table of Contents

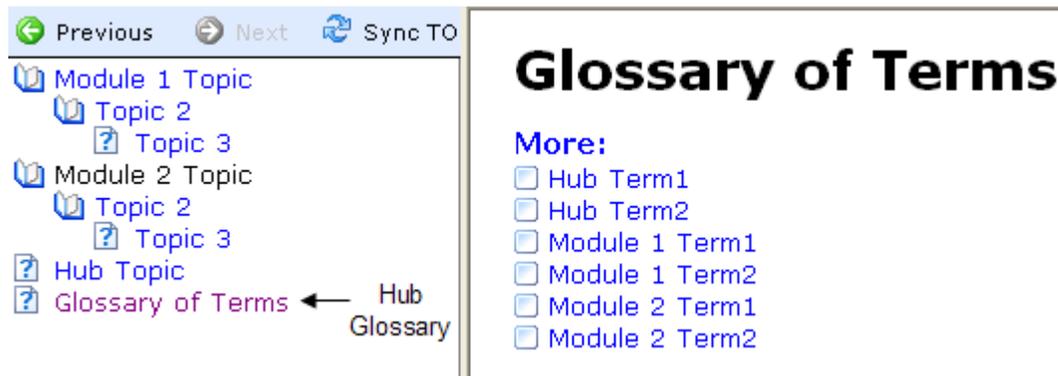
Once the hub project has been built and the modules and hub NetHelp targets have been combined, the placeholder topics in the TOC are replaced with the entire TOC of the corresponding module.



If you want to customize the TOC and add topics that are not placeholders, [edit the TOC](#) (page 291) in the hub project.

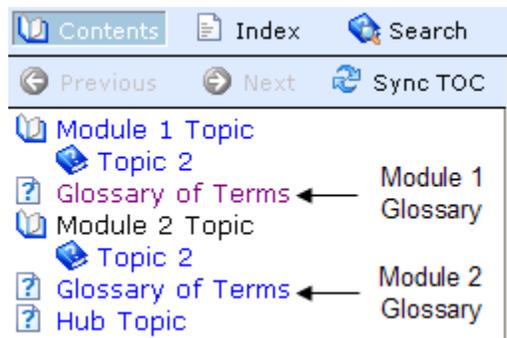
NetHelp Module Glossaries

A merged glossary containing the module and hub glossaries is created in the hub NetHelp target only if the hub project has a glossary document, that is, its [Glossary](#) property is not empty. The module glossaries do not appear separately in the TOC; they are combined with the hub glossary and appear in the hub TOC.



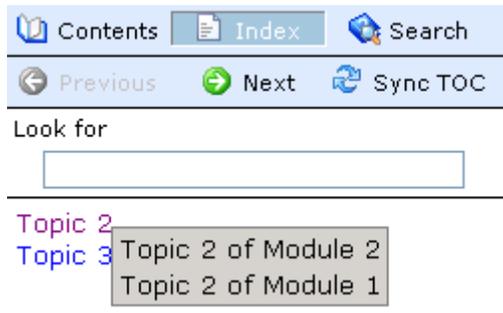
Note: Unlike modular HTML Help and WinHelp, it is not necessary to use the [SkipGlossary](#) property in modular NetHelp. In the NetHelp target, if the glossaries are merged, duplicate glossary entries are automatically eliminated.

If the hub project does not have a glossary document, the modules retain their own separate glossaries. The glossary for each appears in the TOC of its corresponding module.



Keyword and Group Links in NetHelp

Keyword and group links work globally across modules in the NetHelp target. When a keyword or group link is clicked, all topics that are indexed by this keyword or group in all modules appear in the multiple topic links menu.



If there is only one topic indexed by a keyword or group, the multiple topic links menu does not appear; the link goes directly to the topic. The link can jump to a topic within the module it appears, or it can be a cross-module link, linking to a topic from a different module.

Creating Cross-Module Links in NetHelp

In a NetHelp modular Help system, you can create cross-module links, or links from one module topic to a topic in a different module. First, create a group in both the source and destination module projects, and index the destination topic to that group in the destination module project. Then, when the hub project is built and the link in the source module is clicked, it jumps to the topic in the destination module.

For example, suppose we have two module projects and we want to create a link from Module1 to Module2. Module1 is the source module, and Module2 is the destination module since it contains the topic to which we want to link. Use the following steps:

1. Create a group named *Modules* in your Module1 project.
2. In your Module1 source document, [create a group link](#) (page 311) to the *Modules* group.
3. Build the NetHelp target.
4. Create a group named *Modules* in your Module2 project.
5. Assign the topic you want to link to from Module1 to the *Modules* group.
6. Build the NetHelp target.
7. Open your hub project and build the NetHelp target.
8. Click the group link in your Module1 topic. It jumps to the topic assigned to the *Modules* group from your Module2 project.

Scripting Techniques

This chapter describes common operations necessary to write code modules using VBScript that automate indexing and hyperlinks in ComponentOne Doc-To-Help 2006 projects.

For complete VBScript documentation, visit the Microsoft Scripting Technologies site at <http://msdn.microsoft.com/scripting>.

What are Scripts?

Doc-To-Help scripts are code modules written in the VBScript. They can be used to modify the behavior of paragraph and character styles during compilation. If you are familiar with event-driven languages such as Visual Basic, you can think of a script as an "event handler" for a style. Just as the Click event is fired when the user clicks a button in a Visual Basic application, a script is executed whenever Doc-To-Help encounters a topic or a hot spot defined by a scripted style.

You can perform the following operations in script code:

- Create top-level and secondary index keywords.
- Create named topic groups.
- Assign index keywords and groups to topics.
- Assign context-sensitive Help IDs to topics.
- Modify the link tags of topics and hot spots.

You can perform all of these operations in Doc-To-Help's authoring environment using techniques outlined elsewhere. If the notion of writing code seems daunting, then you are not required to use scripts at all. However, a simple script can save you many hours of tedious labor. This is especially true in cases where the title of a topic or the text of a hot spot can be split apart or rearranged to yield a textual match or a different focal point for an index entry. For example, given a set of topic titles of the form *Adjective Noun*, you could write a script to parse the title and associate three index keywords with each topic:

```
Adj ecti ve noun
nouns
nouns, Adj ecti ve
```

Note that you can control both the capitalization and the number (singular or plural) to conform to standard indexing conventions. You can also employ secondary keywords to improve the readability of the index:

```
Adj ecti ve noun
nouns
  Adj ecti ve
```

Suppose further that the topics are organized into several documents, with each document describing the features of a particular *Widget*. With just a few more lines of code, you could extend the index profile for each topic as follows:

```
Adj ecti ve noun
Adj ecti ve noun, of Wi dget
nouns
nouns, Adj ecti ve
nouns, of Wi dget
Wi dget
Wi dget, Adj ecti ve noun
Wi dget, nouns of
```

If secondary keywords are used, the reader sees:

```
Adj ecti ve noun
  of Wi dget
nouns
  Adj ecti ve
```

```
of Widget
Widget
Adjective noun
nouns of
```

If you have to document hundreds of nouns for dozens of widgets, then maybe writing a script isn't such a crazy idea after all!

Scripts are Subroutines

When you create a script named `xyz`, Doc-To-Help automatically creates a subroutine declaration for it, resulting in the following VBScript code:

```
Sub xyz()
' Body of xyz script
End Sub
```

This has the following implications:

- Script names must be legal procedure names in VBScript. That is, they must start with a letter and contain only letters, numbers, and underscores.
- You cannot define other subroutines or functions within the body of a Doc-To-Help non-global script, as VBScript does not support nested procedures.

```
Function pi () ' Not a valid Doc-To-Help script
pi = 3.14159
End Function
```

- You can use the `Exit Sub` statement to exit a script.

```
If expression Then
    MsgBox "Exiting script"
    Exit Sub
End If
```

- You can use the `Call` statement to execute another script.

```
Call xyz
xyz ' Call keyword is optional
```

Unlike other Doc-To-Help objects such as styles, scripts have no properties that you can set in the authoring environment (other than the code itself)

Note: There is one exception to the rule that script names must be legal VBScript procedure names. If a script name starts with a character that is not a letter, Doc-To-Help treats it as a global module. This means that it can contain subroutines and functions that can be called from other scripts. For more information, see [Creating a Global Script Module](#) (page 442).

Scripts Access the Doc-To-Help Object Model

Since scripts are subroutines without arguments, your code must communicate with the Doc-To-Help objects. Whenever a script runs, there is always one default global object named **Project**. The term *global* means that the object is accessible to all scripts. The term *default* means that you do not have to refer to it explicitly by name. That is, the following statements are equivalent:

```
Project.Print "The current Help target is " & Project.Platform
Print "The current Help target is " & Target
```

The **Project** object has three collections that represent the individual items in a project. The most important of these are the following:

Topics	All topics in the project.
Keywords	All top-level index keywords in the project.
Groups	All named topic groups in the project.

In addition to its collections, the **Project** object also has two fundamental properties, each of which returns an object that is central to the compilation process:

ActiveTopic	Returns a Topic object representing paragraph style formatting. This property can be used only in scripts assigned to a paragraph style.
ActiveLink	Returns a Link object representing character style formatting. This property can be used only in scripts assigned to a character style.

Scripts are Extensions of Styles

In order for a script to be executed during compilation, you must assign it to a style. Scripts assigned to paragraph styles handle topics; scripts assigned to character styles handle topic references or hot spots.

If you assign an empty script to a style, the behavior of the style during compilation is the same. The behavior of your style only changes if the code or properties of the Help system is changed.

For both paragraph and character styles, you can use the following properties to override or cancel the default behavior as specified in the **Styles** tab:

Tag	Sets or returns the default link tag.
Key	Sets or returns the default index keyword.
Cancel	Cancels both the default link tag and the default index keyword when set to True .

These three properties apply to the **Topic** object returned by the **ActiveTopic** property as well as the **Link** object returned by the **ActiveLink** property.

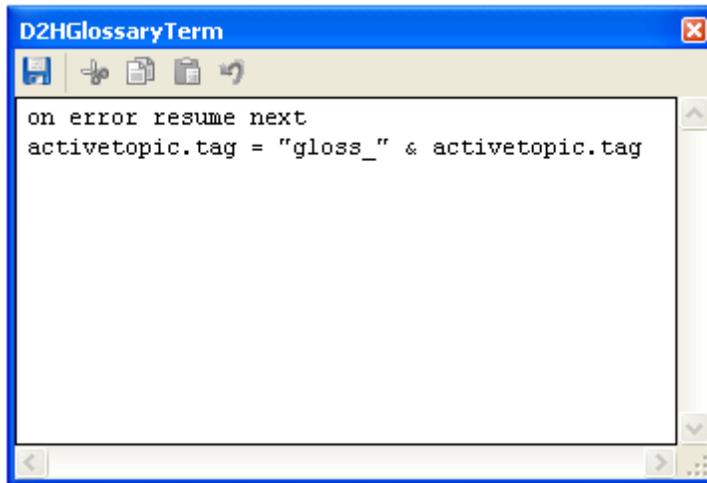
Working with Scripts

This section describes how to create, edit, run, and debug scripts.

Creating a New Script

To create a new script:

1. Highlight the **Script** item on the project editor tree list.
2. Right-click anywhere in the script list pane on the right and choose **New** from the shortcut menu.
3. Note that the focus of your cursor has moved to the top of the script list window.
4. Type the name of the new script and press **Enter**.
5. Note that the new script name has been added to the list.
6. To activate the editable code textbox, highlight the new script name in the **Upper Right Pane**, then click the box located to the right of the **Code** property box in the **Properties Pane**. In this textbox you can enter a new script code or edit an existing one.



Alternatively, you can add a new script by clicking the **Project** menu, choosing **New**, then clicking **Script**.

Creating a Global Script Module

If you have multiple scripts that perform similar tasks, you can simplify your code by moving the common statements to a separate procedure, then calling the procedure directly from the original scripts. Normally, scripts cannot contain subroutines or functions, but if a script is designated as a global script it can contain them.

You can designate a script as a global module by choosing a name that does not begin with a letter, such as "(General)". For more specific instructions, see [Creating a New Script](#) (page 441).

Since a script specified as a global script is treated as a repository for multiple `Sub` and `Function` declarations, Doc-To-Help will not wrap the body of the script with `Sub` and `End Sub` statements.

You cannot assign a global script module to a style.

Editing Scripts

1. Highlight the **Script** item on the project editor tree list.
2. Select the name of the script to be edited from the right pane.
3. To activate the editable code textbox, click the box located to the right of the [Code](#) property box. Edit the code within the text box as desired.
4. Note that your changes affect the selected script instantly and need not be saved explicitly. However, the full effect of your changes is not realized until you build the Help target.

For details on how to write scripts, see [Common Script Operations](#) (page 444).

Assigning a Script to a Style

In order for a script to be executed during compilation, you must assign it to a style as follows:

1. Highlight the **Paragraph Styles** or **Character Styles** item on the project editor tree list.
2. Select the name of the style to be edited from the right pane.
3. In the properties pane for the style, click the box to the right of the [Script](#) property and choose the appropriate script from the drop-down.

Note: You can assign one script to multiple styles. A script must be created in the project before it can be assigned to a style.

Running a Standalone Script

1. In the left hand pane of the project editor, highlight the **Scripts** item.
2. In the right hand pane of the project editor, right-click on the script you wish to run and select **Run** from the shortcut menu.

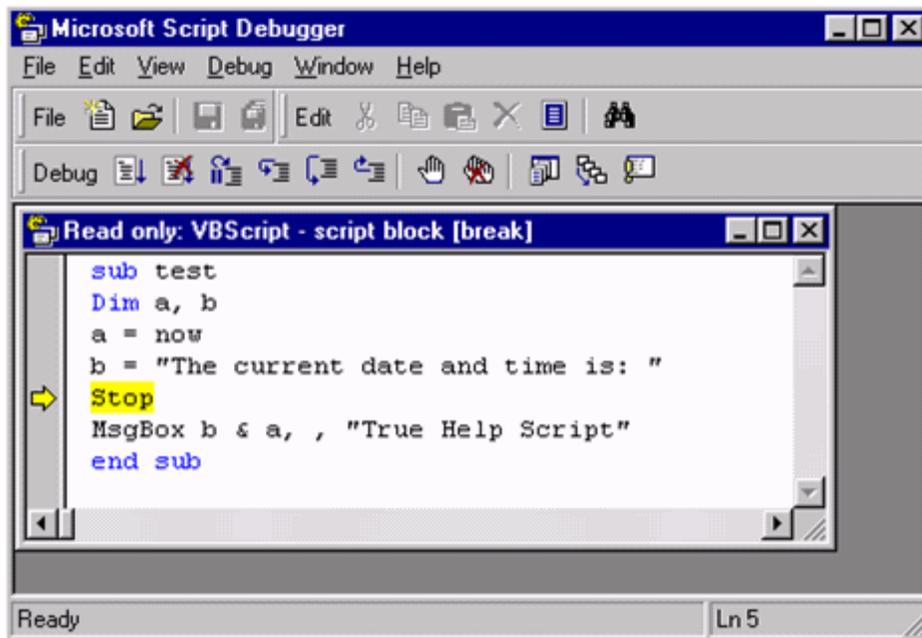
Debugging a Script

The following programming constructs are always available for displaying intermediate results or interacting with a running script:

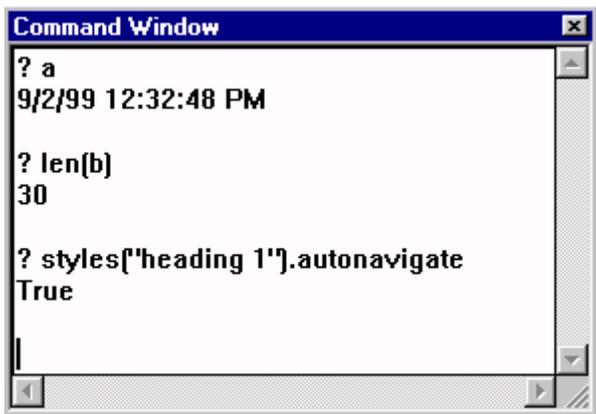
Print	This Doc-To-Help method writes the value of a variable or constant to the build output window.
MsgBox	This VBScript function displays a message in a modal dialog box and optionally prompts for a response using command buttons.
InputBox	This VBScript function displays a message in a modal dialog box and prompts for a character string.

If you have the Microsoft Script Debugger installed, you can also use the `Stop` statement to halt execution and open a separate window for interactive debugging.

Doc-To-Help setup does not install the Microsoft Script Debugger, but you can download a free copy from the Microsoft Web site at <http://msdn.microsoft.com/scripting>.



You can use the debugger's command window to display the variable values, execute intrinsic VBScript functions, or examine the object model of Doc-To-Help.



Common Script Operations

This section demonstrates some typical uses of scripts, particularly for indexing, and provides tips for handling errors.

Changing automatic link tags in a script

In order for a topic to be a destination for a jump or pop-up link, it must have a unique link tag that identifies it. Typically, you do this by enabling automatic link tags for an active paragraph style. For specific instructions, see [Adding a Link Tag to a Topic](#) (page 289).

Automatic link tags are based on the topic title, but are normalized so that they do not contain any illegal characters (as restricted by the Windows Help compiler):

- Spaces, hyphens, and periods are converted to underscores.
- Letters, numbers, and underscores are unchanged.
- All other characters are removed.

When a script assigned to a paragraph style executes at compile time, the following expression returns the normalized link tag generated by Doc-To-Help:

```
ActiveTopic.Tag
```

However, you can change the default behavior by setting the **Tag** property to a different value. For instance, if you have several documents with like-named topics, you can generate a unique link tag for each topic by appending the document name:

```
Dim name, pos
name = ActiveTopic.Document.Name
pos = Instr(1, name, ".")
ActiveTopic.Tag = ActiveTopic.Tag + "_" + Left(name, pos - 1)
```

Note the use of the intrinsic VBScript functions `Instr` and `Left` to extract the root filename, minus the extension. The variable `pos` receives the character position of the period in the filename, and the `Left` function returns a string containing all characters to the left of the period.

To prevent a link tag from being generated, set the **Tag** property to an empty string:

```
ActiveTopic.Tag = ""
```

For more information, see [Linking Related Topics](#) (page 27).

Changing automatic index keywords in a script

For reference material, the topic title often serves as a convenient index. Typically, you index reference topics by enabling automatic index keywords for an active paragraph style. For specific instructions, see [Creating Index Keywords Automatically Using Styles](#) (page 343).

When a script assigned to a paragraph style executes at compile time, the following expression returns the name of the automatic index keyword generated by Doc-To-Help:

```
ActiveTopic.Key
```

However, you can change the default behavior by setting the **Key** property to a different value. For instance, you can convert the keyword to lower case with the following statement, which uses the intrinsic VBScript function `LCase`:

```
ActiveTopic.Key = LCase(ActiveTopic.Key)
```

To prevent an index keyword from being generated, set the **Key** property to an empty string:

```
ActiveTopic.Key = ""
```

Creating index keywords in a script

To create top-level index keywords, use the **Add** method of the **Keywords** collection:

```
Keywords.Add "animal"  
Keywords.Add "vegetable"  
Keywords.Add "mireal"
```

Keywords created in this manner are not associated with any topics. This is analogous to creating top-level keywords in the **Index** toolbar.

However, if you use the **Keywords** collection of the active topic, and not of the project, then the keywords are created and associated with the active topic:

```
ActiveTopic.Keywords.Add "animal"  
ActiveTopic.Keywords.Add "vegetable"  
ActiveTopic.Keywords.Add "mireal"
```

This is analogous to entering keywords in the **Index** tab of the topic properties pane. Note that you can simplify the script and make it more efficient by assigning the **Keywords** collection of the **ActiveTopic** object to a variable:

```
Dim keys  
Set keys = ActiveTopic.Keywords  
keys.Add "animal"  
keys.Add "vegetable"  
keys.Add "mireal"
```

Creating secondary index keywords in a script

To create secondary index keywords, use the **Add** method of the **Keywords** collection and specify the parent keyword as the second argument. The following example creates a top-level keyword and then creates a secondary keyword as its child:

```
Dim key  
Set key = Keywords.Add("mammals")  
Keywords.Add "Homo sapiens", key
```

Note the use of the variable `key` as a placeholder for the top-level **Keyword** object. Keywords created in this manner are not associated with any topics. This is analogous to creating secondary keywords in the **Index** toolbar.

However, if you use the **Keywords** collection of the active topic, and not of the project, then both top-level and secondary keywords are created and associated with the active topic:

```
Dim key  
Set key = ActiveTopic.Keywords.Add("mammals")  
ActiveTopic.Keywords.Add "Homo sapiens", key
```

This is analogous to using the **Index** command of the **Topic** menu twice, once for each keyword. Note that you can simplify the script and make it more efficient by assigning the **Keywords** collection of the **ActiveTopic** object to a variable:

```
Dim key, keys  
Set keys = ActiveTopic.Keywords  
Set key = keys.add("mammals")  
keys.add "Homo sapiens", key
```

Creating topic groups in a script

To create named groups, use the **Add** method of the **Groups** collection:

```
Groups.Add "novice"
Groups.Add "intermediate"
Groups.Add "expert"
```

Groups created in this manner are not associated with any topics. This is analogous to creating groups in the **Index** toolbar.

However, if you use the **Groups** collection of the active topic, and not of the project, then the groups are created and associated with the active topic:

```
ActiveTopic.Groups.Add "novice"
ActiveTopic.Groups.Add "intermediate"
ActiveTopic.Groups.Add "expert"
```

This is analogous to entering groups in the **Index** tab of the topic properties pane. Note that you can simplify the script and make it more efficient by assigning the **Groups** collection of the **ActiveTopic** object to a variable:

```
Dim grps
Set grps = ActiveTopic.Groups
grps.Add "novice"
grps.Add "intermediate"
grps.Add "expert"
```

Handling error conditions in a script

When writing scripts, you need to be aware that some operations may fail. Fortunately, you can use the built-in error handling features of VBScript to take the appropriate action.

When working with collections, a common error is attempting to access an element that does not exist. Consider the following standalone script, which tries to assign a bogus keyword to a variable:

```
Dim k
Set k = Keywords("@#!%$")
Print TypeName(k)
```

Running this script produces the following error message:

```
Script error on line 2, column 1:
Keyword not found: '@#!%$'
```

Since all script errors are fatal, the `Print` statement never executes. However, if an `On Error` statement is inserted before line 2, the script executes in its entirety:

```
Dim k
On Error Resume Next
Set k = Keywords("@#!%$")
Print TypeName(k)
```

This time, the script prints the word `Empty` to the output window, since the variable `k` is still uninitialized. Technically, the `Set` statement still results in an error, but the `On Error` directive causes execution to continue with the statement immediately following the one that caused the error.

Another common error is attempting to add an element to a collection that already exists. Consider the following standalone script, which adds a new group to the project:

```
Groups.Add "qwerty"
```

The first time this script is run, it executes silently. However, running it again produces the following error message:

```
Script error on line 1, column 1:
Group already exists: 'qwerty'
```

Again, the solution is to use the `On Error` statement:

```
On Error Resume Next
Groups.Add "qwerty"
```

VBScript also provides an **Err** object that you can use to determine whether the last statement was successful. The following example demonstrates how to use the **Err** object to write code that always operates on a specified member of a collection, whether or not it already exists:

```
Dim k
On Error Resume Next
Set k = Keywords("primary")
If Err <> 0 Then ' keyword not found
    Set k = Keywords.Add("primary")
    Err.Clear
End If
' Add a secondary keyword under "primary"
Keywords.Add "secondary", k
```

The **Set** statement in the third line attempts to access a specific keyword. If it exists, the default property of the **Err** object returns zero, and execution continues with the last line of the script. If the keyword does not exist, the script creates it using the **Add** method. In either case, when the last line of the script executes, the variable **k** contains the appropriate object.

Finally, note that the scope of the **On Error** statement is limited to the procedure in which it appears. Consider the following function, which resides in a global script module:

```
Function GetKeyword(Name)
    On Error Resume Next
    Set GetKeyword = Keywords(Name)
    If Err <> 0 Then ' keyword not found
        Set GetKeyword = Keywords.Add(Name)
        Err.Clear
    End If
End Function
```

The **GetKeyword** function generalizes the technique employed in the previous example. It accepts a string and unconditionally returns a top-level keyword having the same name. If necessary, the function creates a new keyword; otherwise, it returns an existing one. The following example demonstrates how to call **GetKeyword** from another script:

```
Dim k
Set k = GetKeyword("primary")
Keywords.Add "secondary", k
```

If the specified secondary keyword already exists, the third line results in an unhandled error, even though the **GetKeyword** function uses the **On Error** statement. In order to trap all errors, an additional **On Error** statement is required:

```
Dim k
Set k = GetKeyword("primary")
On Error Resume Next
Keywords.Add "secondary", k
```


DocuMenter for .NET

DocuMenter for .NET creates and maintains documentation for one or more .NET assemblies. All you have to do is select the assemblies you want to document and **DocuMenter** will automatically generate MSDN-style reference documentation, complete with formatting and references. Additionally, DocuMenter for .NET generates a Doc-To-Help project with the reference document. This project can be used to build any Help target supported by Doc-To-Help.

Note: DocuMenter creates Word source documents for your reference section, although you may use Word and/or HTML documents in the project.

DocuMenter uses reflection to collect information about the classes in your assemblies. Because the information is generated directly from the assemblies, it is always accurate and up-to-date. *Even without source code* comments, DocuMenter builds a complete reference document with all types and members present.

DocuMenter can also use XML documentation files created by the C# compiler. The XML documentation is created automatically by the compiler using comments embedded in the source code. Because this type of documentation is written along with the code, it is easy to keep up-to-date. Also, the compiler can check any cross-references automatically and spot documentation errors at compile time. For details on XML documentation, please refer to MSDN (see the XML Documentation Tutorial).

DocuMenter is fully integrated with **Doc-To-Help**. Because of this, you can leverage all the power of **Doc-To-Help** to enhance the automatically generated documentation. You can, add custom narrative sections, cross-references, indices, Help targets, styles, and more.

Note: DocuMenter for .NET supports Microsoft Visual Studio 2005 .NET assemblies in addition to Visual Studio .NET 2002 and 2003 assemblies.

Generation Speed and Supported Word Versions

Generation speed in DocuMenter for .NET is dramatically optimized - up to 20 times faster - with Word 2003. The previous Word versions, Word 2000 and Word XP, are also fully supported with exactly the same document generated. However, the generated document is slower with old Word versions. This dramatic speed enhancement in Word 2003 is due to using the new XML features (internally in DocuMenter and therefore transparent to the user) that are only available starting with Word 2003.

Please note that only Generate New Document is optimized with using Word 2003 XML features. Regenerate Existing Document has the same speed with Word 2003 as with earlier Word versions. However, Regenerate Existing Document is usually very fast as it is, without additional optimization.

DocuMenter for .NET Guided Tour

This section walks you through the process of documenting an existing assembly. We will use an assembly from a Barcode application for this tutorial. The components for this tutorial are located in *C:\Program Files\ComponentOne\DocToHelp\Tutorial\BarCode*.

\BarCode Assembly\Original	Contains the .DLL and .XML files used for the original build of the tutorial project.
\BarCode Assembly\Updated	Contains the .DLL and .XML files used for the update build of the tutorial project.

\Barcode Assembly\Document	Contains the CustomTopics.doc used to add an additional document during the tutorial.
\Barcode Project	Contains the completed Documenter project example.

Note: The Documenter for .NET Guided Tour assumes familiarity with .NET programming.

Selecting the Assemblies to Document

Start by selecting the assembly (or assemblies) you want to document.

Note: Documenter for .NET supports Microsoft Visual Studio 2005 .NET assemblies in addition to Visual Studio .NET 2002 and 2003 assemblies.

Even without source code comments, Documenter produces a complete reference document with all types and members present. Brief descriptions are included if they are supplied in the **Description Attribute** in the assembly.

XML comments are optional, but they are a great feature when used with **Documenter**. They are easy to add and maintain, and the compiler helps ensure the documentation is complete and accurate. XML comments encourage you to write and update the documentation in a consistent manner, while you are writing or modifying the code.

If the assembly contains XML documentation comments, make sure the corresponding XML documentation file is up-to-date by rebuilding the assembly.

For this tutorial, we will use the **C1.Win.C1Barcode.Dll** assembly, a simple .NET assembly that contains a single WinForms control. The assembly is located in *C:\Program Files\ComponentOne\DocToHelp\Tutorial\Barcode\Barcode Assembly*.

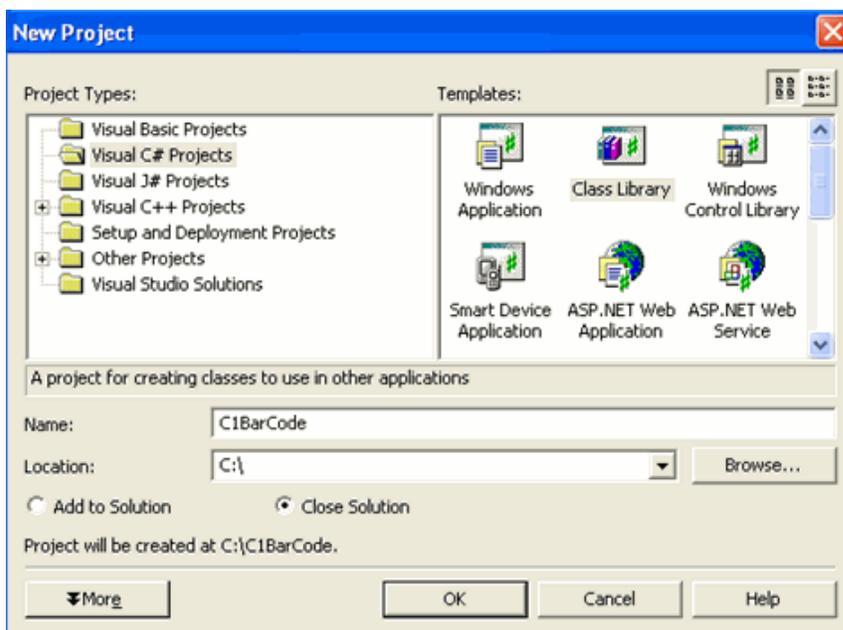
Creating an Assembly

The following steps are provided as a guide to creating an assembly with XML comments, which can be used to create an XML documentation file. The assembly XML documentation file for this tutorial has already been created and is available in the *C:\Program Files\ComponentOne\DocToHelp\Tutorial\Barcode\Barcode Assembly\Original* folder.

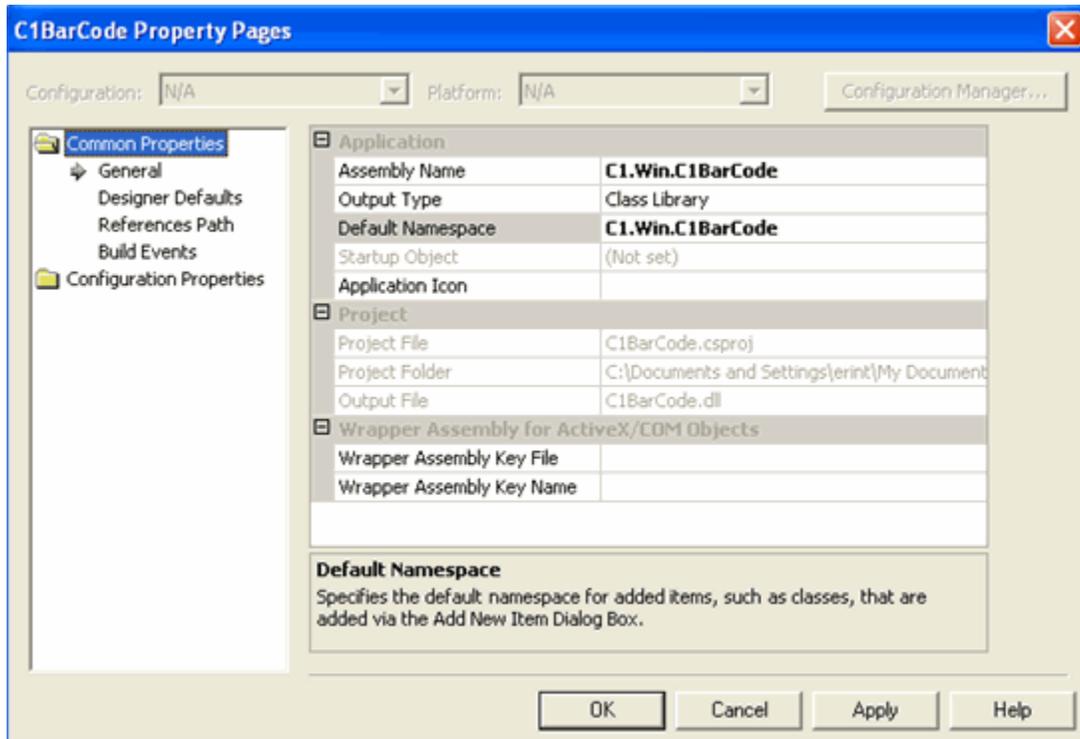
Steps are provided for both Visual Studio 2002/2003 and Visual Studio 2005.

Using Visual Studio 2002/2003

1. Open Visual Studio and select **File | New | Blank Solution**. The **New Project** dialog box appears.



2. Under Project Types, select Visual C# Projects.
3. Under Templates, select ClassLibrary.
4. Enter **C1Barcode** in the **Name** textbox.
5. Choose a location and click **OK**.
6. Select Project | C1Barcode Properties. The Property Pages dialog box appears.



7. Under the **Common Properties** folder in the left pane, select **General**.
8. In the right pane, enter **C1.Win.C1Barcode** in the **Assembly Name** textbox. Notice the **Output Type** property is already set to **Class Library**.
9. Enter **C1.Win.C1Barcode** in the **Default Namespace** property and click **OK** to close the property pages.
10. Add the following code, which was taken from the **C1Barcode** assembly, to the **Class1.cs**.

```

namespace C1.Win.C1Barcode
{
    /// <summary>
    /// Specifies the type of barcode to be generated by the C1Barcode control.
    /// </summary>
    /// <remarks>
    /// The CodeTypeEnum enumeration represents the different types of barcode that can
    /// be generated by the C1Barcode control. Some types can be used to represent
    /// alphanumeric values, others can only represent numeric values.
    /// </remarks>
    public enum CodeTypeEnum
    {
        /// <summary>
        /// Code 39 is an alpha-numeric encoding also known as 3 of 9 and LOGMARS.
        /// This was the first alphanumeric symbology developed, and is one of the
        /// most widely used encodings.
        /// </summary>
        Code39,

        /// <summary>
        /// Code 93 is an alpha-numeric encoding that is slightly denser than code 39.
        /// </summary>
    }
}

```

```

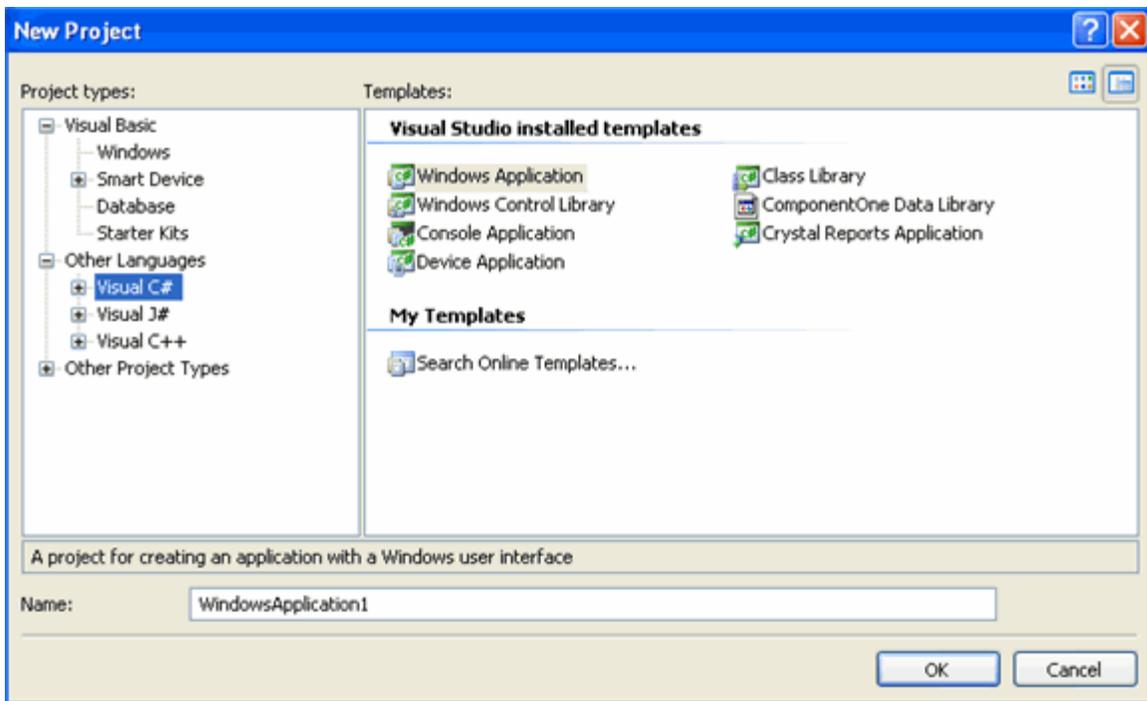
Code93,
/// <summary>
/// Code 128 is a very high density alpha-numeric bar code. It will use the
/// least amount of space of any current 1-D symbology for symbols with 6
/// characters or more.
/// </summary>
Code128,
}
}

```

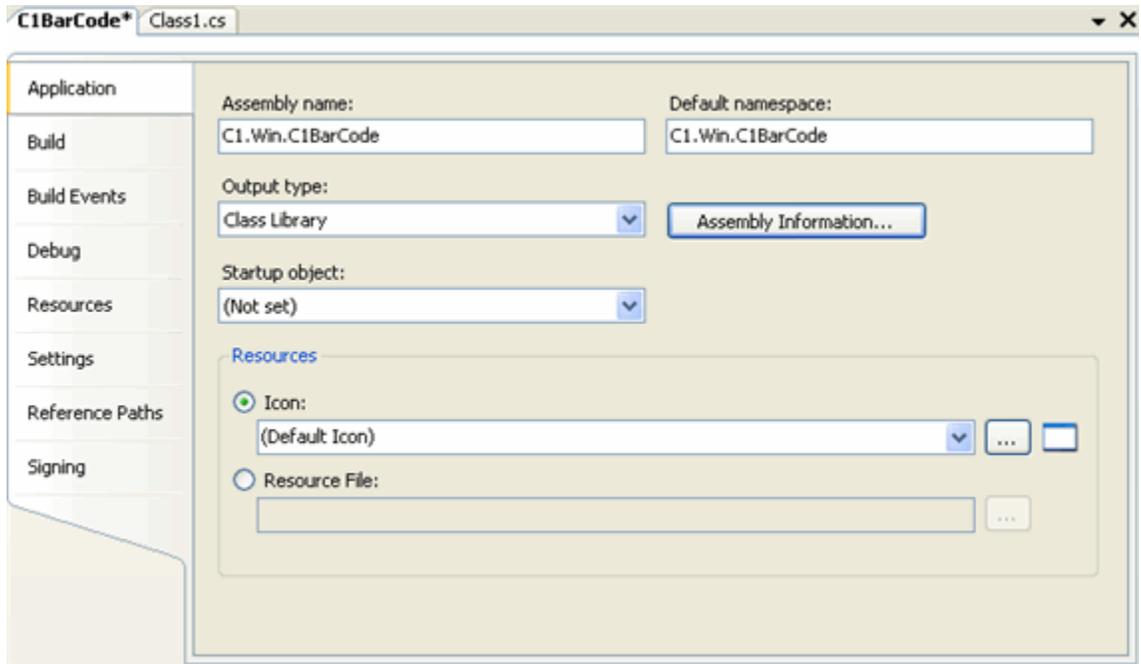
11. Run and save your project. The assembly is created and can be used to generate an XML documentation file.

Using Visual Studio 2005

1. Open Visual Studio and select **File | New Project**. The **New Project** dialog box appears.



2. Under Project types | Other Languages, select Visual C#.
3. Under Templates, select ClassLibrary.
4. Enter **C1Barcode** in the **Name** text box and click **OK**.
5. Select **Project | C1Barcode Properties**. The **Property Pages** dialog box appears, and the **Application** tab is selected.



6. On the **Application** tab, enter **C1.Win.C1BarCode** in the **Assembly Name** textbox. Notice the **Output Type** property is already set to **Class Library**.
7. Enter C1.Win.C1BarCode in the Default Namespace.
8. Add the following code to the **Class1.cs**, which was taken from the **C1BarCode** assembly.

```
namespace C1.Win.C1BarCode
{
    /// <summary>
    /// Specifies the type of barcode to be generated by the C1BarCode control.
    /// </summary>
    /// <remarks>
    /// The CodeTypeEnum enumeration represents the different types of barcode that can
    /// be generated by the C1BarCode control. Some types can be used to represent
    /// alphanumeric values, others can only represent numeric values.
    /// </remarks>
    public enum CodeTypeEnum
    {
        /// <summary>
        /// Code 39 is an alpha-numeric encoding also known as 3 of 9 and LOGMARS.
        /// This was the first alphanumeric symbology developed, and is one of the
        /// most widely used encodings.
        /// </summary>
        Code39,

        /// <summary>
        /// Code 93 is an alpha-numeric encoding that is slightly denser than code 39.
        /// </summary>
        Code93,

        /// <summary>
        /// Code 128 is a very high density alpha-numeric bar code. It will use the
        /// least amount of space of any current 1-D symbology for symbols with 6
        /// characters or more.
        /// </summary>
        Code128,
    }
}
```

- Run and save your project. The assembly is created and can be used to generate an XML documentation file.

Generating the XML Documentation File

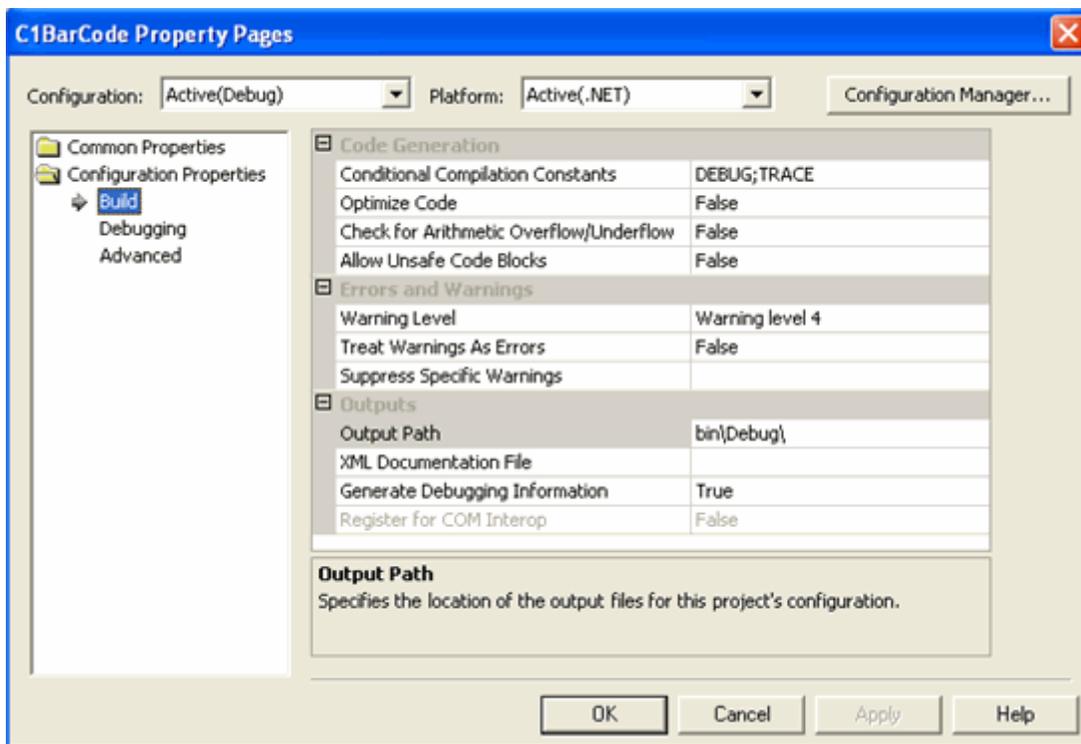
To generate the XML documentation file, configure your project in Microsoft Visual Studio. The C# compiler will parse the XML comments and generate the XML documentation file when you build the project. The compiler will even issue warnings about undocumented members and broken cross-references, so you can check that the documentation is complete.

Note: This tutorial assumes you have a C#.NET project named C1Barcode containing the **C1.Win.C1Barcode.Dll** assembly. The following steps have been provided only as an example of how to create an XML documentation file. The XML documentation file for this tutorial has already been created and is available in the *C:\Program Files\ComponentOne\DocToHelp\Tutorial\Barcode\Barcode Assembly\Original* folder.

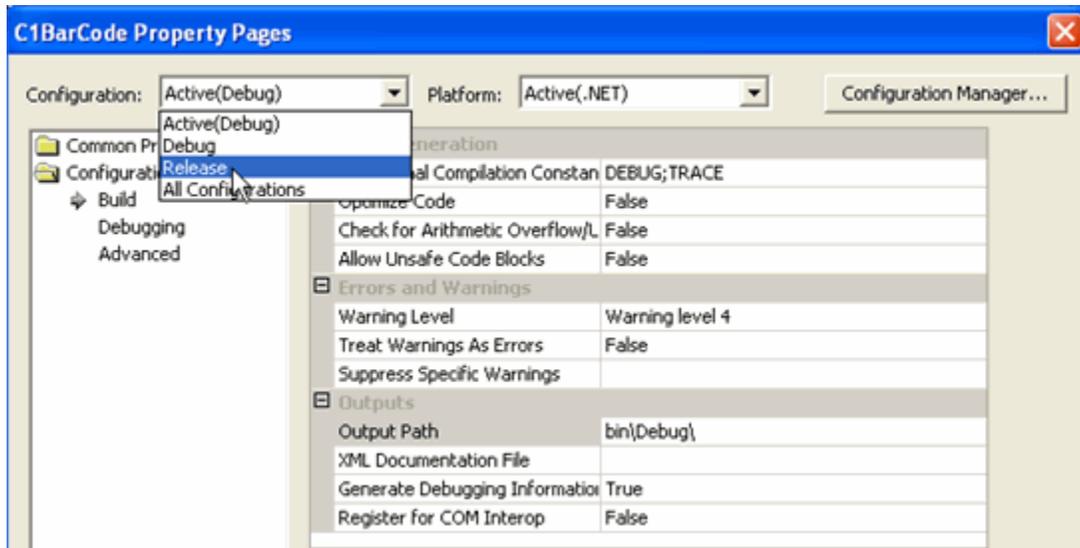
To generate the XML Documentation file:

Using Visual Studio 2002/2003

- Open the **C1Barcode** assembly project created in [Creating an Assembly](#) (page 450) if it is not already open.
- Select **C1Barcode Properties** from the **Project** menu. The **Property Pages** dialog box appears.

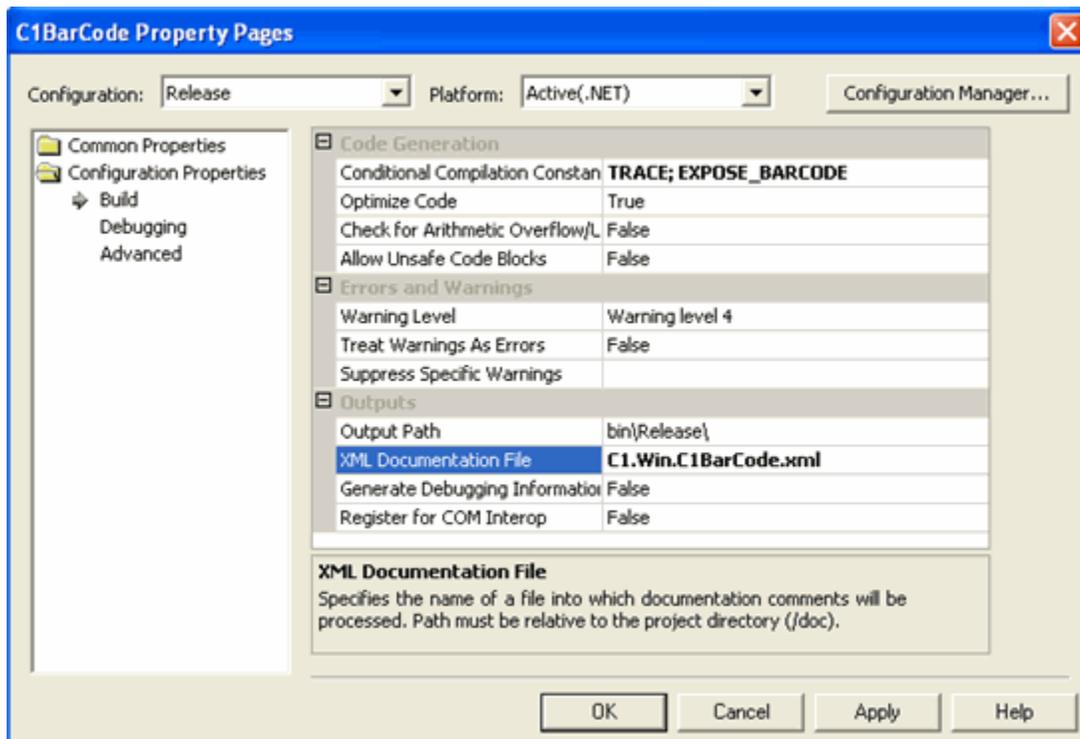


- In the **Configuration** drop-down list at the top of the **Property Pages**, select **Release**.



Note: The XML documentation option was added to the **Release** configuration only. This is because most projects are built in **Debug** mode many times, and in **Release** mode only a few times. Adding documentation only in **Release** builds saves some time. Also, the **Debug** configuration may have conditional compilation blocks which result in object models slightly different from the **Release** configuration. It is important to document exactly what will be released, rather than debug builds.

4. Under the **Configuration Properties** folder in the left pane, select **Build**.
5. In the **Conditional Compilation Constants** textbox, enter TRACE;EXPOSE_BARCODE.
6. In the **XML Documentation File** textbox, enter **C1.Win.C1Barcode.xml**. The **Property Pages** should now look like the following example.



7. Click **OK** to close the **Property Pages** dialog box.

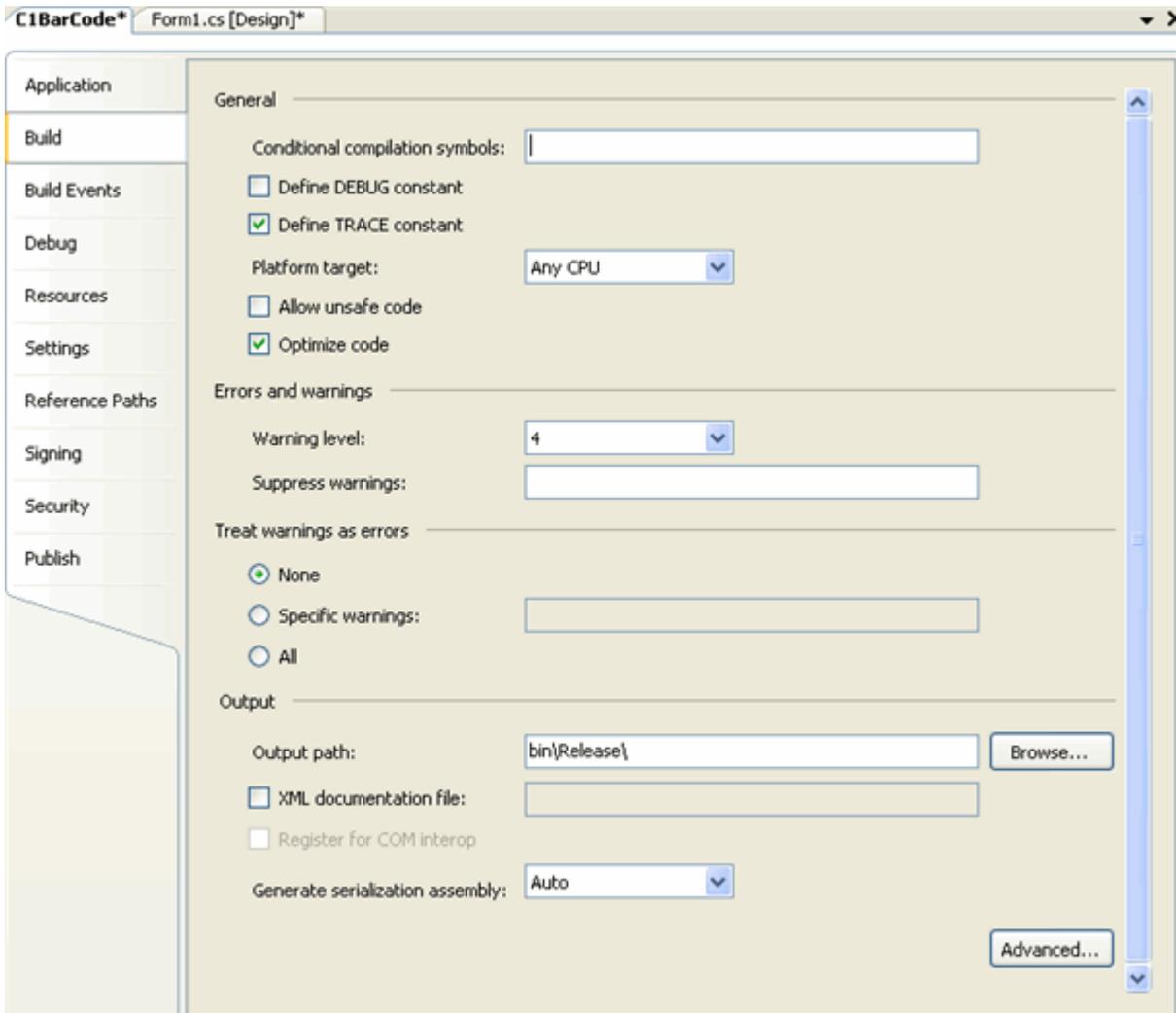
8. Select **Release** from the **Solution Configurations** drop-down list on the **Standard** toolbar.



9. Click **Build | Build Solution** to build the project. The **C1.Win.C1Barcode.xml** is created within the **bin\Release** folder where your project is located.

Using Visual Studio 2005

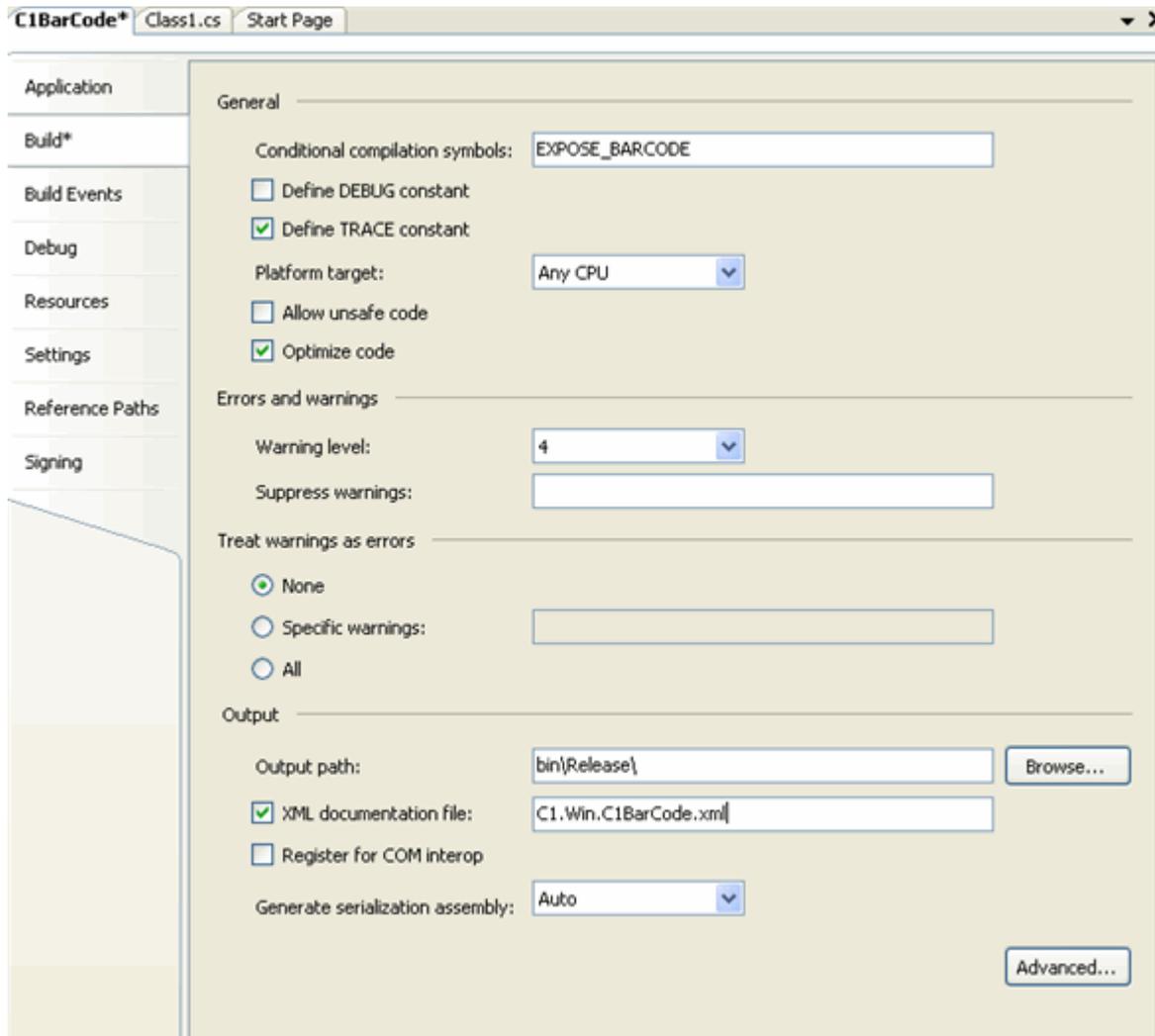
1. Open the **C1Barcode** assembly project created in [Creating an Assembly](#) (page 450) if it is not already open.
2. Select **C1Barcode Properties** from the **Project** menu. The **Properties** page appears.



3. Select the **Build** tab.
4. In the **Conditional compilation symbols** textbox, enter **EXPOSE_BARCODE**.
5. Enter **bin\Release** in the **Output path** textbox.

Note: The XML documentation option was added to the **Release** configuration only. This is because most projects are built in **Debug** mode many times, and in **Release** mode only a few times. Adding documentation only in **Release** builds saves some time. Also, the **Debug** configuration may have conditional compilation blocks which result in object models slightly different from the **Release** configuration. It is important to document exactly what will be released, rather than debug builds.

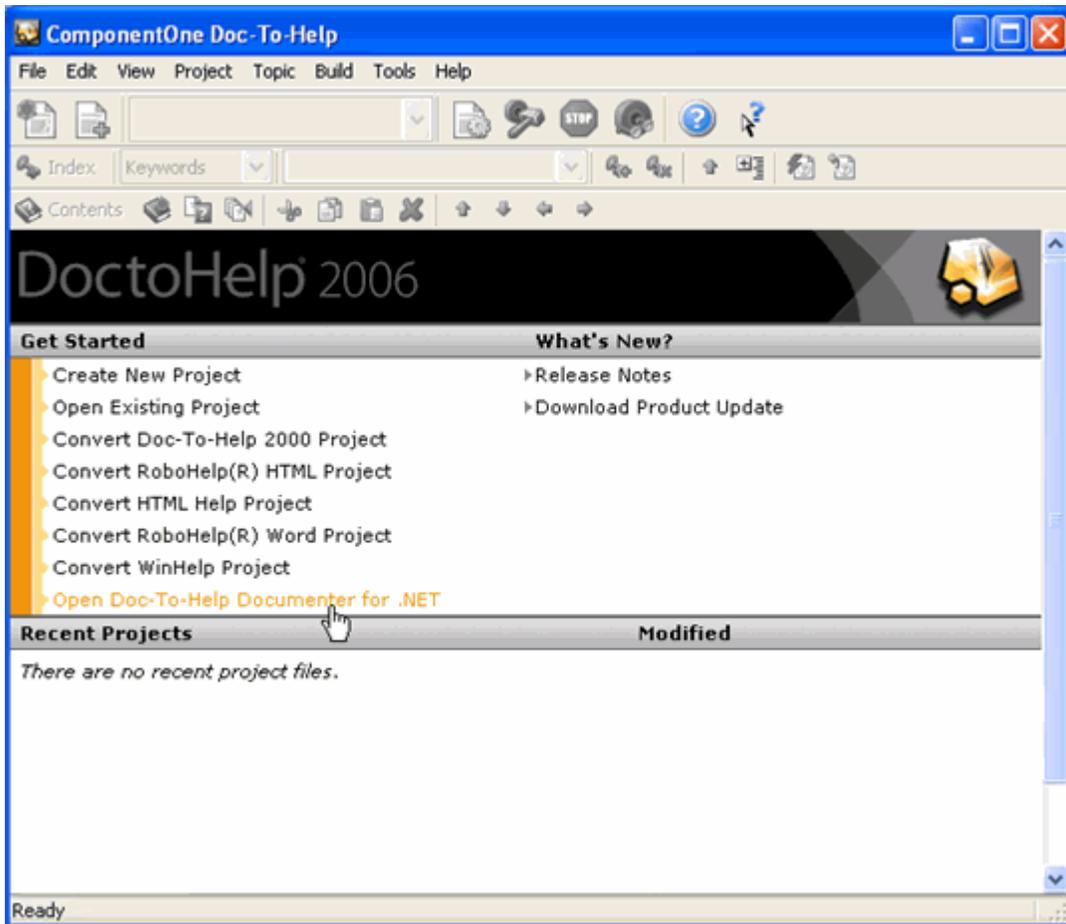
6. Check the **XML documentation file** checkbox and enter **C1.Win.C1Barcode.xml** for the file name. The **Properties** page should now look like the following example.



7. Click **Build | Build C1Barcode** to build the project. The **C1.Win.C1Barcode.xml** is created within the **bin\Release** folder where your project is located.

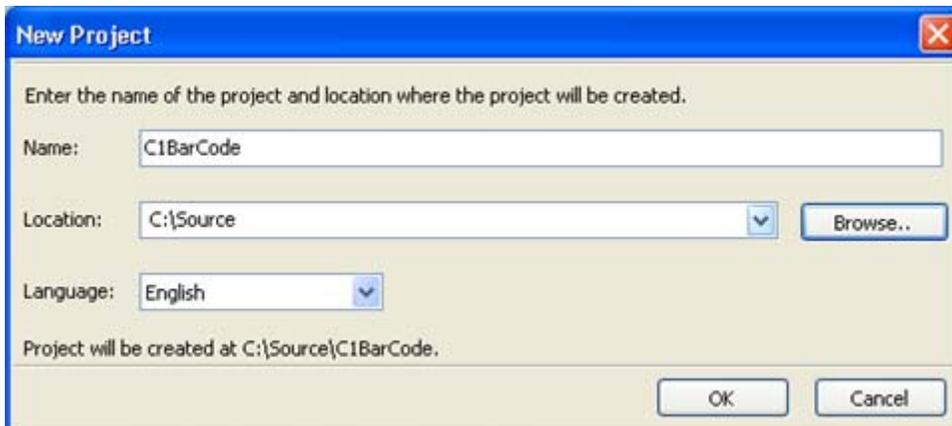
Creating a New Doc-To-Help Documenter Project

1. Create a directory named **Source** in a convenient location. This folder is used to store your .DLL and .XML files for the purposes of this tutorial.
2. Copy the **C1.Win.C1Barcode.DLL** file and **C1Barcode.XML** file from *C:\Program Files\ComponentOne\DocToHelp\Tutorial\BarCode\BarCode Assembly\Original* to your **Source** directory.
3. Start ComponentOne Doc-To-Help and select the Open Doc-To-Help Documenter for .NET option from the main screen.



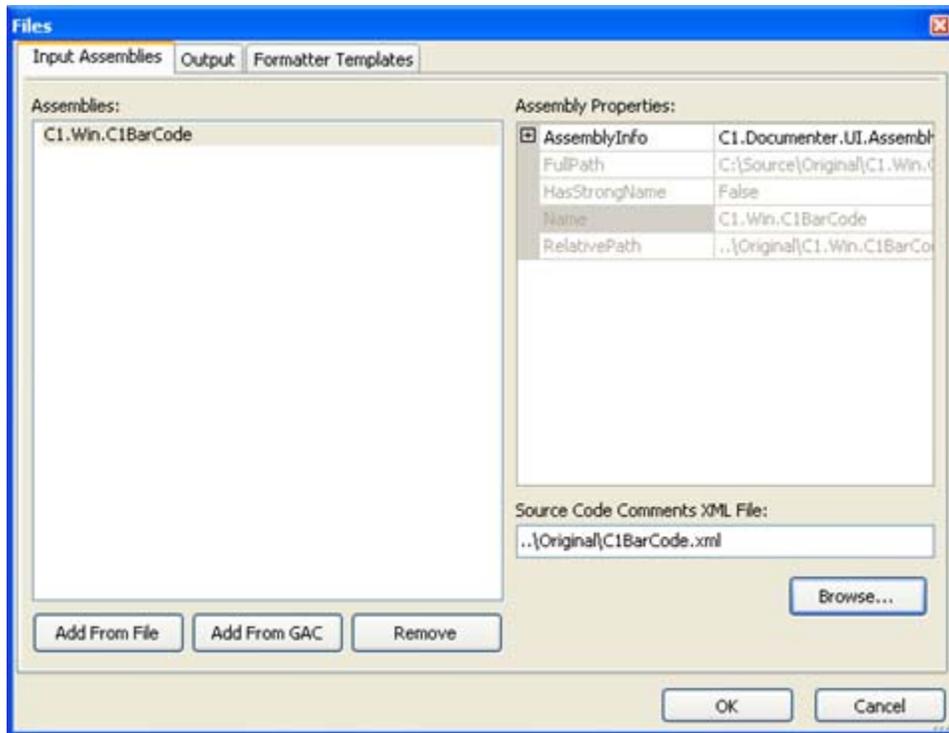
This will bring up the **Documenter** start dialog, where you can select to create a new project or open an existing one.

4. Click the **New Project** button, then select a name and location for the new documentation project.



5. Click **OK** and the new project will be created at the location you specified.

The **Files** dialog will prompt you for the assemblies you want to document and the location of the optional XML documentation file created by the C# compiler.



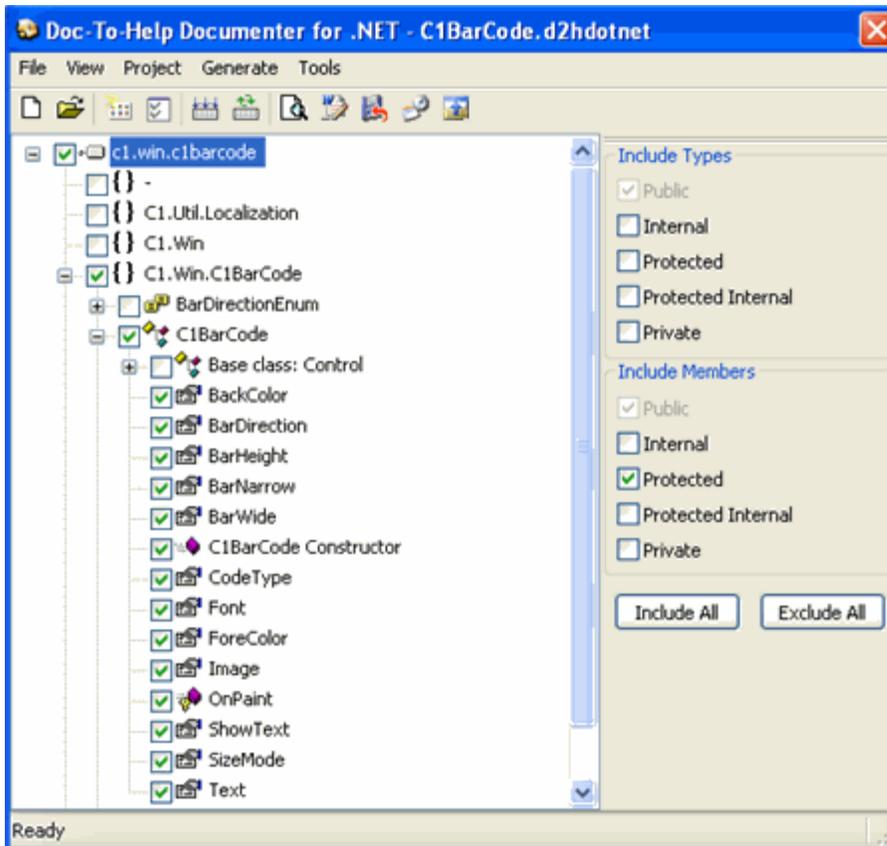
By default, the output document file uses the same name as the project. If you want to modify the name of the output document, select the **Output** tab and modify the document name in the **Generated Document | Name** textbox.

- Click **Add From File** and select **C1.Win.C1BarCode.dll** from your **Source** directory.

Note: You have the option to add from the Global Assemblies Cache (GAC). By clicking **Add From GAC**, Documenter exposes all assemblies stored in the local GAC. "Strongly Named" assemblies must reside in the GAC to be documented.

- Under *Source Code Comments XML File*, click **Browse**, select **C1BarCode.xml** from your **Source** directory and click **Open**.
- After selecting the assembly and .XML file, click **OK**.

Documenter opens a second dialog that allows you to specify which elements you want to include in the documentation.



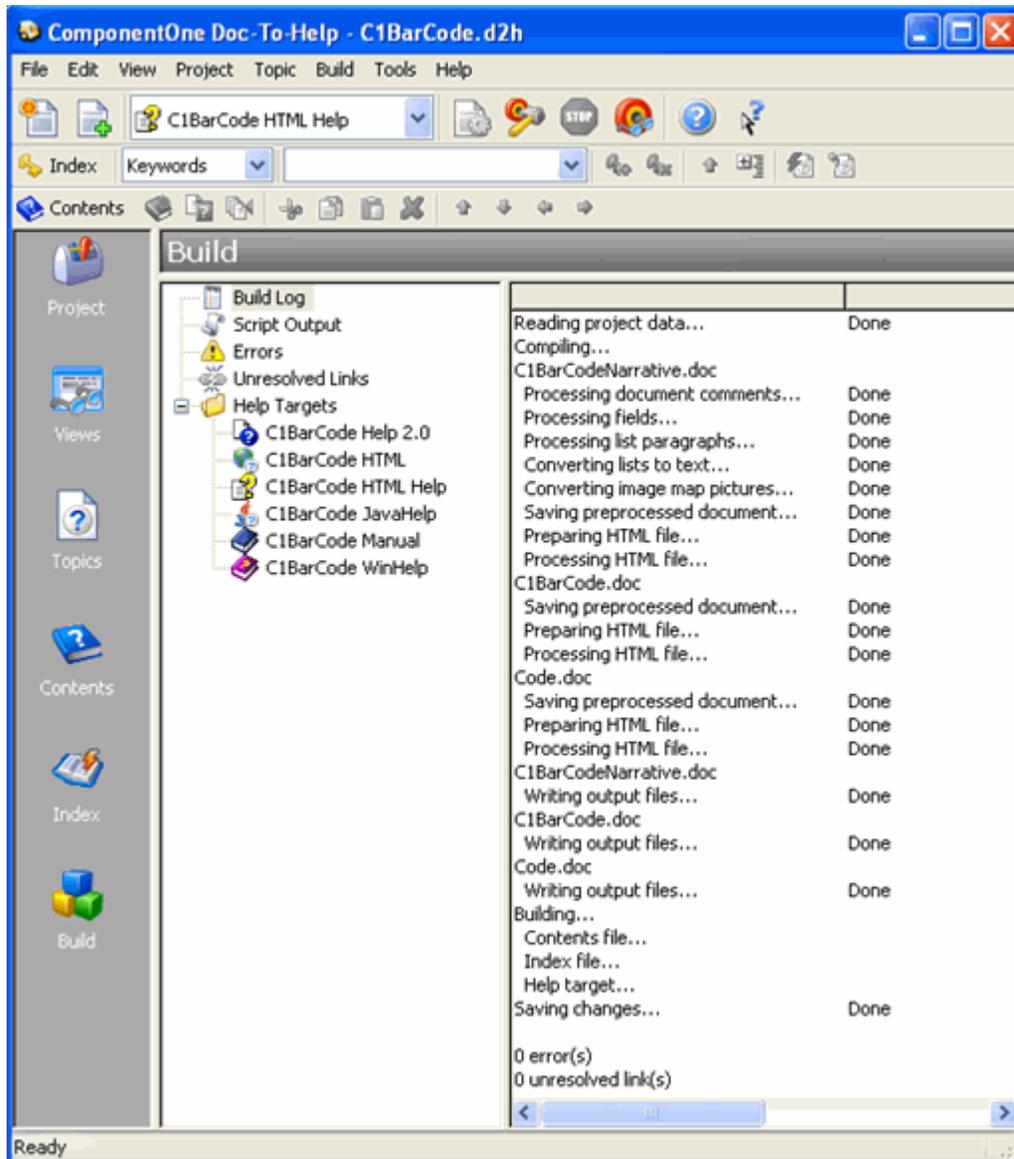
Note that in this case, we unchecked the node that contains the base class. That was done because `C1Barcode` inherits from the .NET `Control` class, and that is documented in MSDN. We will only document the members that are declared in our own assembly.

9. Once you have selected the members you want to document, choose the **Generate | Generate New Document** menu option. **Documenter** creates the Doc-To-Help project files. This may take a few minutes, depending on the size of the files. You can then use the **Tools** menu to view a list of elements without a description, incorrect user changes or invalid XML comments.
10. Click **Close**. The Doc-To-Help **Project Editor** opens, with the new project loaded.

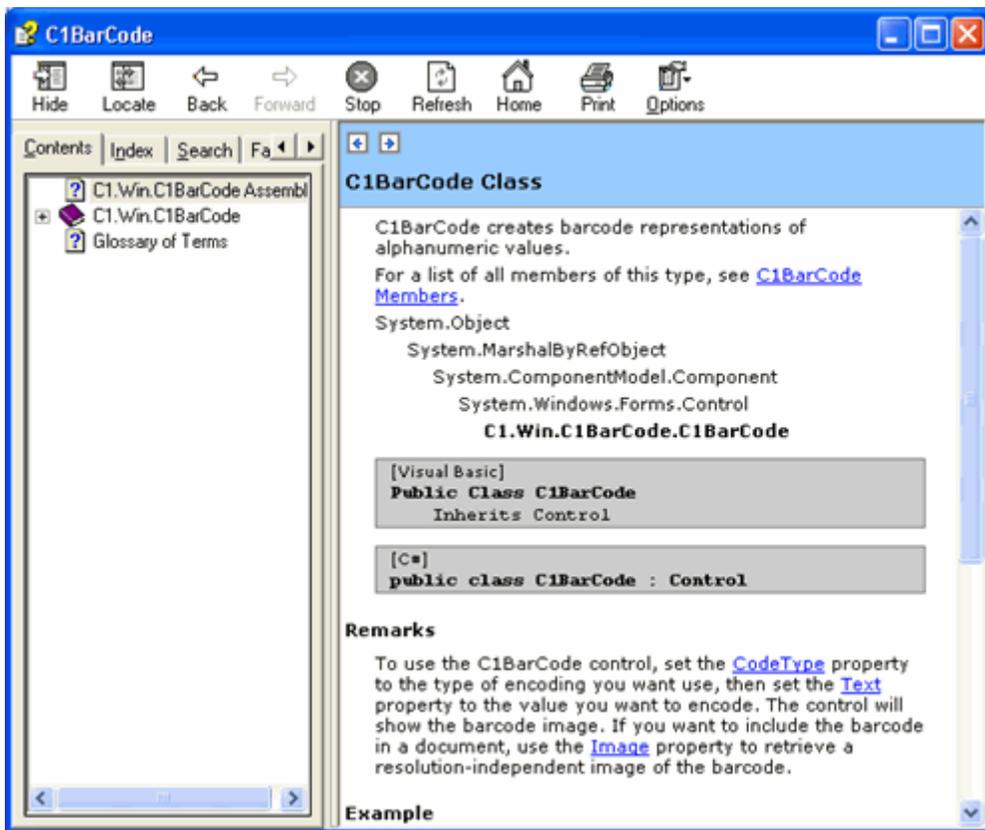
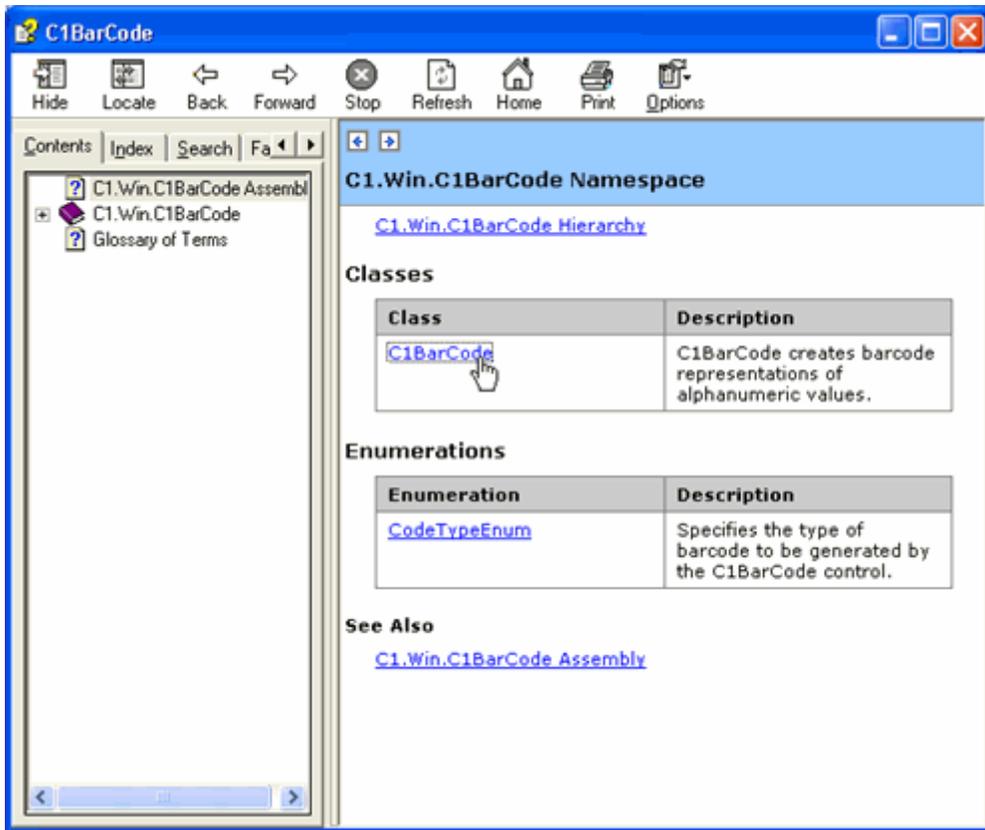
Building the Documentation

At this point, you can use all the regular **Doc-To-Help** commands to work on the project.

1. Select the **HTMLHelp** target on the **Project** toolbar, then use the **Build | Make Target** menu option to build the documentation.



2. Select the **Build | View Target** menu to preview the documentation. Notice how it is formatted and cross-referenced following the Visual Studio standard:



The reference part of the documentation is ready. If you modify your assembly by adding new classes, properties, methods, etc, simply repeat the process to have **Documenter** update the help project.

Note: If you are creating HTMLHelp documentation, be aware that the project path should not contain any folders with periods in their names. For example, if your project is in a folder called "c:\help projects\.net\c1barcode", then **Doc-To-Help** will build the project as usual, but the Microsoft help compiler will not create the .chm help file. This is a limitation of the help compiler, not related to the **Documenter** or **Doc-To-Help**. To work around this problem, you have to rename the folder that contains the period or move the project to a new folder.

Add Narrative Topics to the Documentation

Most documentation projects consist of more than just a reference section. There are also custom topics such as tutorials, how-to sections, and frequently asked questions. You may also want to add boilerplate items such as company information, copyright, and licensing. These topics are not embedded in assemblies, and must be added manually.

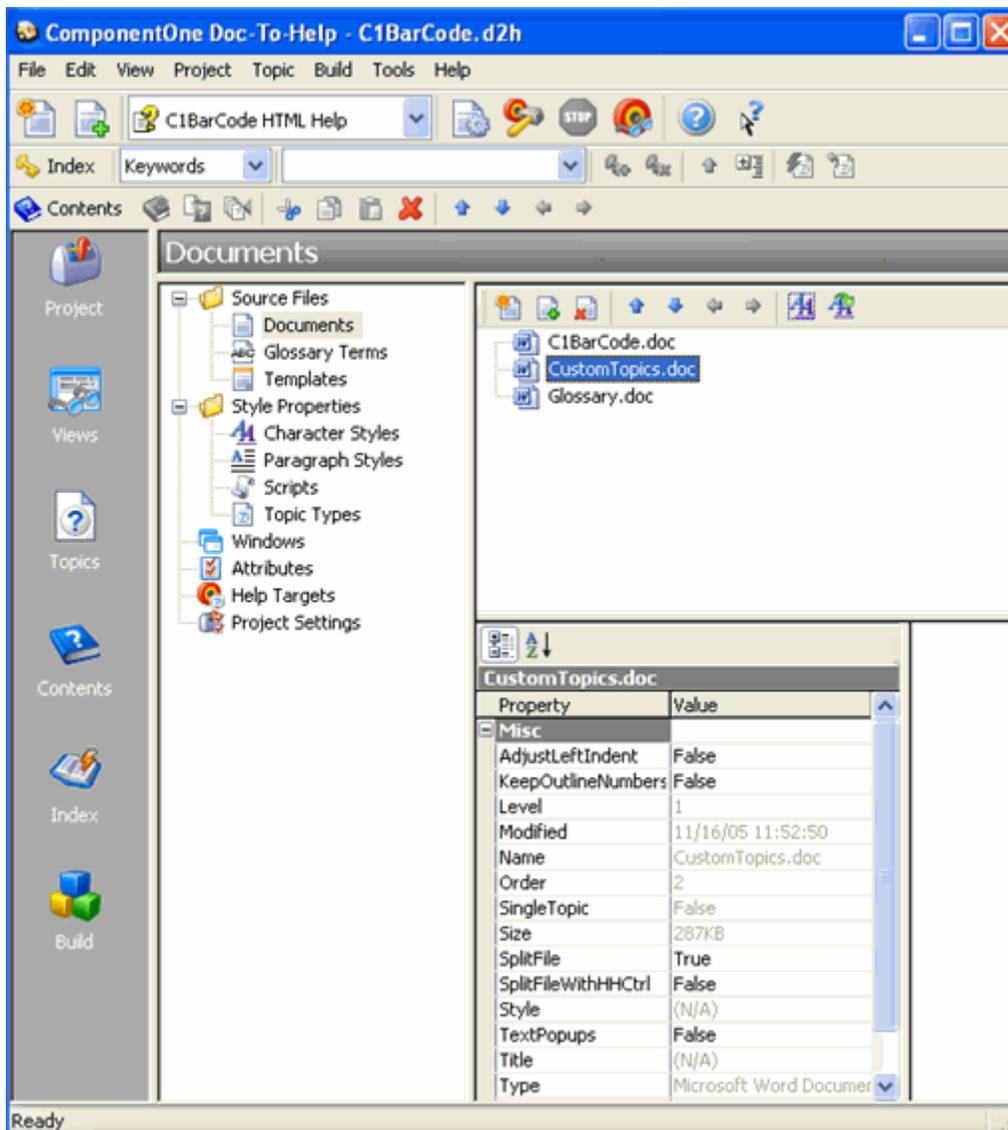
You can make these additions using the standard **Doc-To-Help** features. When your assembly changes and you regenerate the reference part of the documentation, **Documenter** will detect and preserve your changes. This is one of the most important features in **Documenter**: you can integrate the reference and narrative portions of the documentation and always keep it always up-to-date and accurate with minimum effort.

Adding a New Document to the Documenter Project

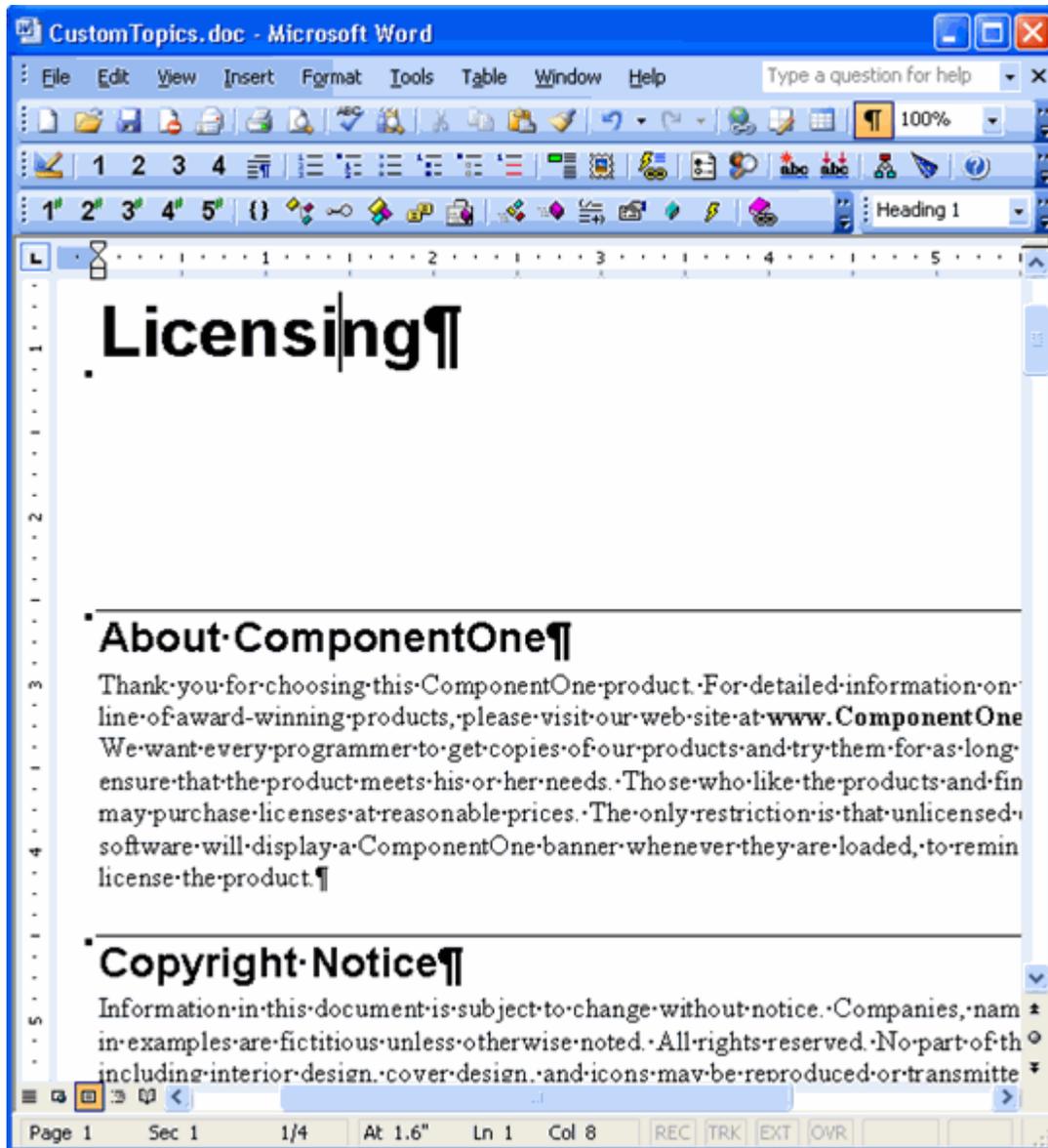
Let's add a new document containing legal and licensing information. This type of information is not product-specific. It is usually written by lawyers and included in every product.

1. Copy the **CustomTopics.doc** file from C:\Program Files\ComponentOne\DocToHelp\Tutorial\BarCode\BarCode Assembly\Document to your **Source** directory.
2. To add the new document, select the **Project | Add Documents** menu.
3. Browse for **CustomTopics.doc**, then click **OK**.

The **CustomTopics.doc** is added to your project.



If you examine the **CustomTopics.doc**, you will note that it already has a title formatted using the default styles and formatting used by **Doc-To-Help**.



Because CustomTopics.doc contains only general information, you can include it in all your help projects. If you want to change the information in that file (to introduce a new licensing policy for example), all you need to do is update CustomTopics.doc file and rebuild your help projects.

Adding Content to the Documenter Produced Document

You can also add custom content to the files that were generated by the **Documenter**.

1. Going back to our example, open the **C1Barcode.doc** and add a paragraph below the sample code under the **C1Barcode Class** topic.

Example

This sample shows how to use the Text property to encode the value "123456" as a Code39-encoded barcode.

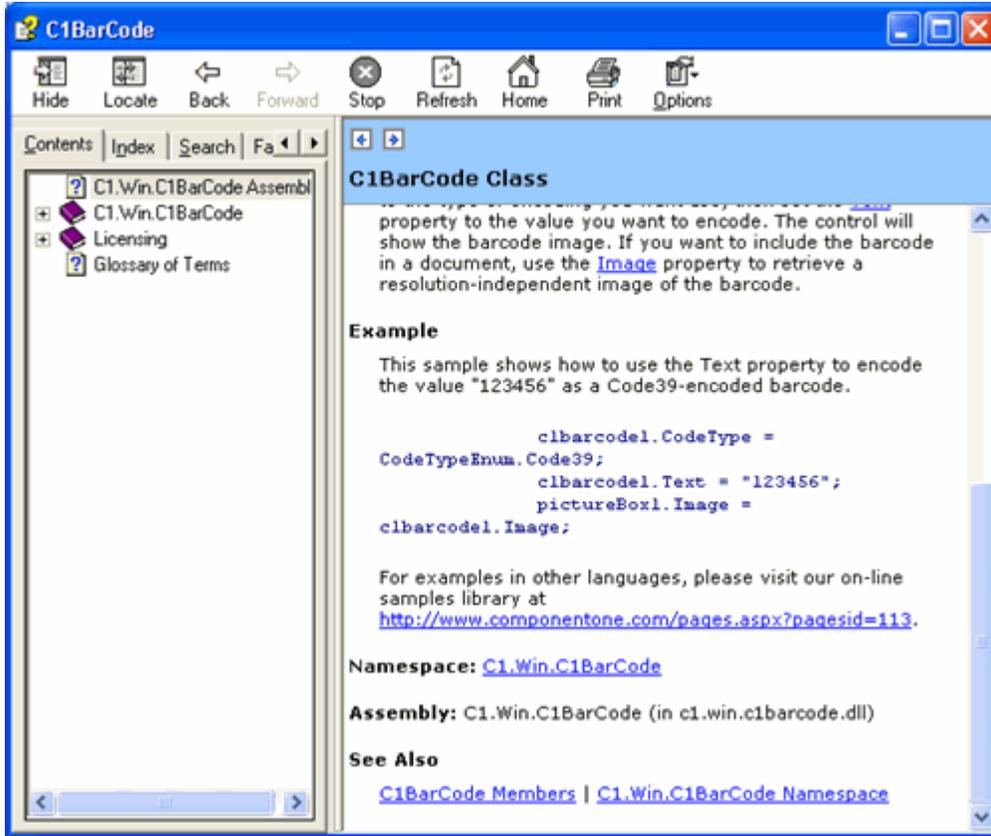
```

c1barcode1.CodeType = CodeTypeEnum.Code39;
c1barcode1.Text = "123456";
pictureBox1.Image = c1barcode1.Image;

```

For examples in other languages, please visit our on-line samples library at <http://helpcentral.componentone.com/ProductResources.aspx?View=Samples>.

2. Save and close **C1BarCode.doc**.
3. Rebuild your Help project. Note that the new content has been added to your Help file.



4. Close the help file and Doc-To-Help project.

Note: **Documenter** is able to preserve the changes you make by tagging automatically generated paragraphs with a hidden header. You can add and remove as many paragraphs as you like, but you should not make changes to the automatically generated paragraphs. If you do, your changes will be lost.

Updating the Assembly and the Documentation

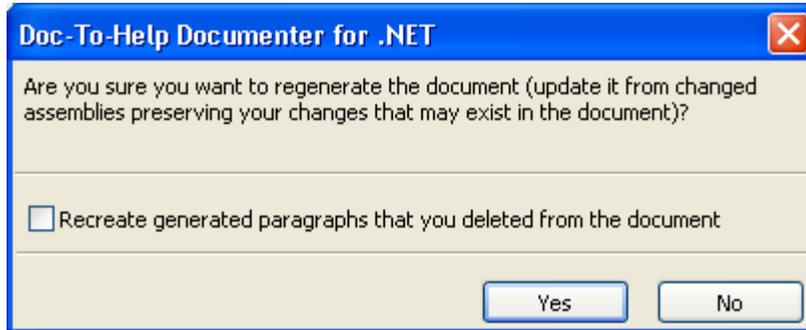
Software evolves. As you release new versions of your assemblies, with new properties, events, and methods, keeping the documentation up-to-date can be a challenge. **Documenter** addresses this by allowing you to regenerate the documentation while keeping the changes you made to the automatically generated files.

To demonstrate this, we will use a new version of the **C1BarCode** assembly and update the documentation to reflect the changes. The new **C1BarCode** assembly has an additional property called **ShowText**. The new assembly is located in *C:\Program Files\ComponentOne\DocToHelp\Tutorial\BarCode\BarCode Assembly\Updated*.

1. Copy (and overwrite the existing) **C1.Win.C1BarCode.DLL** file and **C1BarCode.XML** file from *C:\Program Files\ComponentOne\DocToHelp\Tutorial\BarCode\BarCode Assembly\Updated* your **Source** directory.
2. Open Doc-To-Help.
3. From the main screen, start the **Documenter**, then load the **C1BarCode** project by clicking the **C1BarCode** link. You can also click the **Open Project** button and then locate and open the .d2hdotnet file.

Notice that **Documenter** remembers where to find the assembly, the XML help file, and which elements to include in the documentation. All that information is stored as part of the project.

4. Select **Generate | Regenerate Existing Document**. A dialog box appears asking if you want to regenerate the document.
5. Click **Yes**.



The Documenter will repeat the process of analyzing the assembly and XML help file, and will merge the changes with the existing documentation.

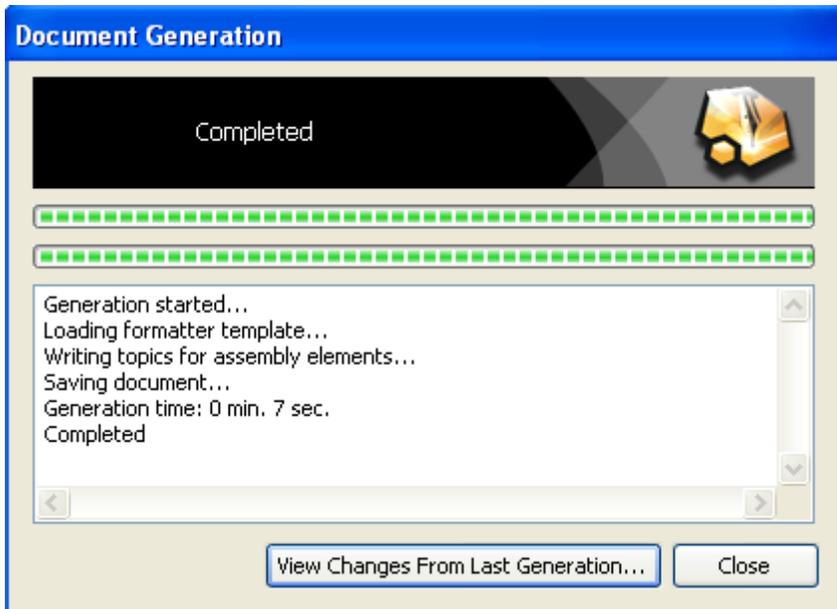
A Note About Regeneration:

Regeneration (rebuilding the automated documentation) removes generated topics and paragraphs that are no longer relevant (part of the assembly), adds new topics and paragraphs as needed, and modifies existing generated paragraphs if they need modification because of assembly changes. Documenter will preserve your changes: user added paragraphs will remain in their places or they will be moved to different places, if the assembly change necessitates the move. Documenter tries to do it in the best possible way, but sometimes there can be user-added paragraphs that cannot be placed in the document after regeneration. Such paragraphs will be lost, removed from the document. For example, you may have a user added paragraph describing a method that is no longer present in the assembly. In this case, your paragraph becomes orphaned, so it is removed from the document.

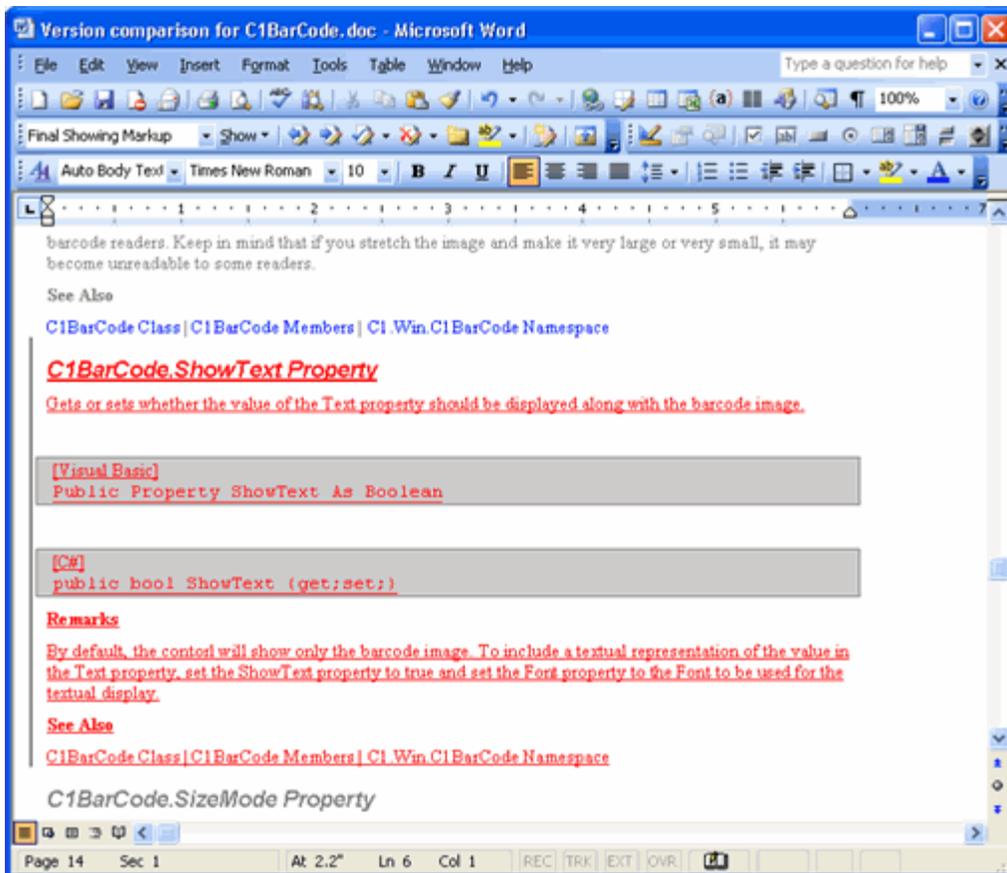
After each regeneration, you should press **View Changes From Last Generation** to examine the changes made by Documenter in regeneration and to fix any issues you find. When fixing any issues, make sure you: only add and delete paragraphs, don't modify generated paragraphs, do not use styles starting with "Auto" in your added paragraphs.

In addition to preserving user-added paragraphs, Documenter "preserves" deletions made by the user, when it regenerates the document. This means that if a generated paragraph is deleted by the user before the regeneration, it won't appear in the regenerated document as well. Sometimes (not often) you may need to suppress this behavior, for example, when you deleted some auto-generated paragraphs by mistake and want the Documenter to generate them again, to restore them for you. To do this, you can check the checkbox **Recreate generated paragraphs that you deleted from the document** in the confirmation dialog shown when you start regeneration. But bear in mind that in this case Documenter will ignore all your previous deletions, that is, it will recreate all paragraphs that you deleted if they are still relevant in the current assembly contents.

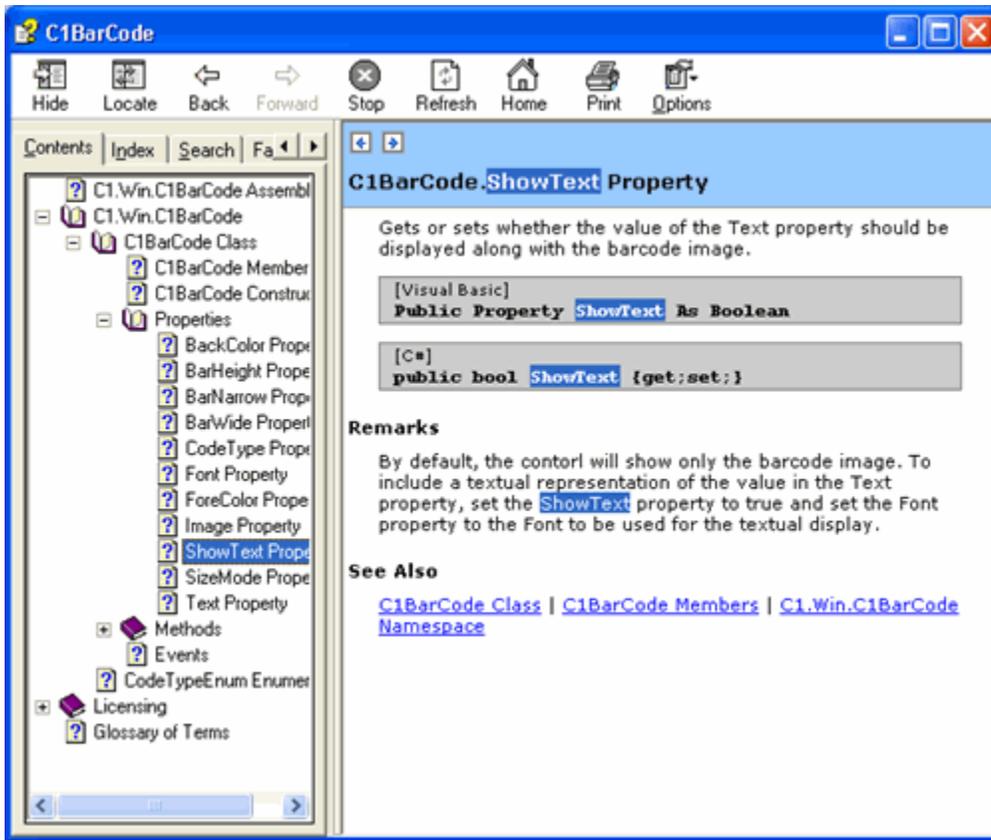
6. Click the **View Changes From Last Generation** button.



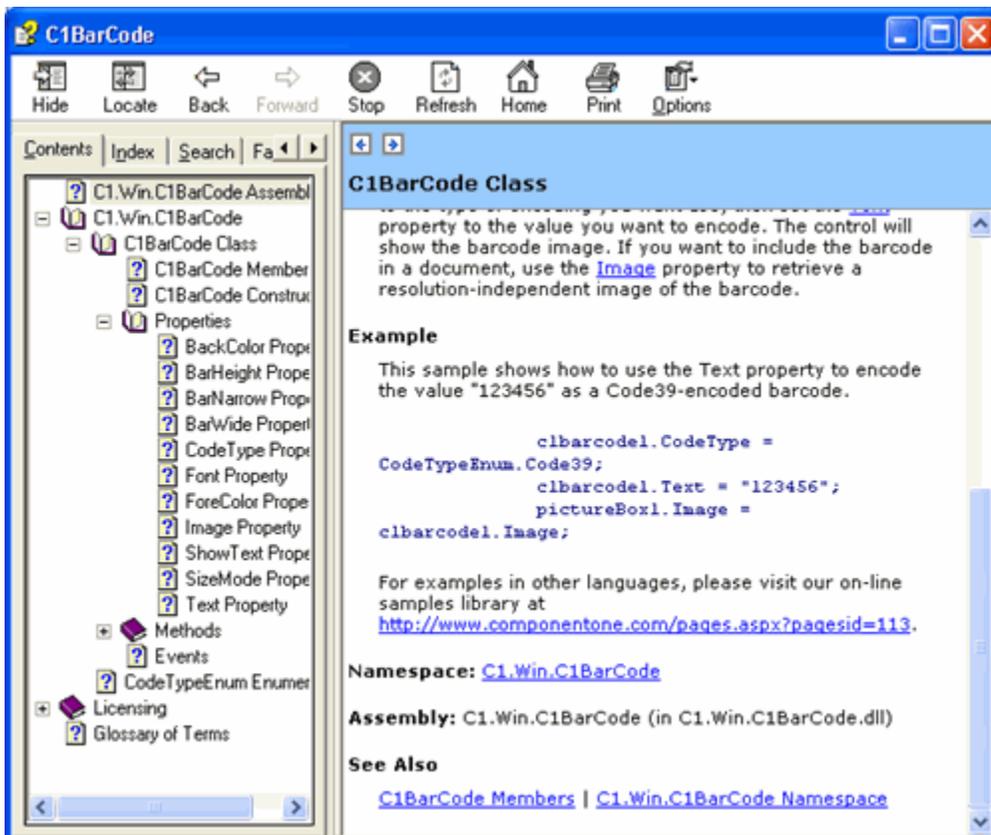
Note that the added property is highlighted in red.



7. Close the Word document displaying the changes and click the **Close** button on the **Document Generation** dialog.
8. Rebuild the help project. Notice the new topics that were added for the new properties, which were also included in the documentation outline.



Notice also how the changes that you made in the previous step were preserved.



This completes the Documenter for .NET Guided Tour.

Documenter Controls and Toolbars

Documenter for .NET uses a set of control icons when producing and updating the automatically generated source document. Additionally, **Documenter** displays a Word toolbar for easy text formatting and style defined link control. For more information on using styles to create automatic links, see [Using Documenter Styles to Create Links](#) (page 484).

The following two topics give you a brief overview of the functions related to each icon.

Exploring the Documenter Controls

The **Documenter** controls are used while defining and building the automated source documentation produced for Doc-To-Help. These controls are located at the top of the Documenter user interface. To learn more about these controls, see [Documenter for .NET Guided Tour](#) (page 449).



	New Project	Start a new Documenter project.
	Open Project	Open an existing Documenter project.
	Project Files	View and edit the assembly and XML files for the current project.
	Project Options	View and edit project options including generated syntax, including/excluding elements and generating additional links.
	Generate New Document	Generate a new Documenter source document.
	Regenerate Existing Document	Regenerate an existing Documenter source document. For information on regenerating a document, see "A Note About Regeneration" in the Updating the Assembly and the Documentation (page 466) topic.
	View Document	View the current Documenter source document.
	Document Changes in Last Generation	View a document showing the changes from the last document generation.
	Backup Document	Backup your current source document (before regeneration).
	Document Backup Versions	View the document backup versions.
	Show Lost Changes in Last Generation	View previously made changes that did not appear in the last document generation.

Some of the controls above also appear on Documenter's menu bar, along with additional controls:

File

New	Start a new Documenter project.
Open	Open an existing Documenter project.
Close	Close the current Documenter project.
Recent Projects	List projects that have recently been open.

Exit Exit Documenter.

View

Open Generated Document Open the generated document.

Document Changes in Last Generation View a document showing the changes from the last document generation.

Document Backup Versions View the document backup versions.

Lost Change in Last Generation View previously made changes that did not appear in the last document generation.

Project

Files View and edit the assembly and XML files for the current project.

Options View and edit project options including generated syntax, including/excluding elements and generating additional links.

Generate

Generate New Document Generate a new Documenter source document.

Regenerate Existing Document Regenerate an existing Documenter source document. For information on regenerating a document, see "A Note About Regeneration" in the [Updating the Assembly and the Documentation](#) (page 466) topic.

Backup Document Backup your current source document (before regeneration).

Tool

List Elements Without Description Lists assembly elements included in the element tree that do not have description neither in the assembly itself (Description attribute) nor in the XML comments. This may be intentional, in which case you probably do not want include those elements in the generated document. To make sure they do not appear in the generated document, check the **Exclude elements without description** check box in the Project | Options dialog. Some elements may be missing description by mistake. Listing them using this menu item helps you to detect such mistakes.

View Incorrect User Changes Opens the document and allows you to see incorrect changes you did in the document. After the document is generated by Documenter, you can make changes to it and they will be preserved when Documenter regenerates the document if those changes satisfy the rule: you can add and remove paragraphs but you should not make changes to the automatically generated paragraphs. This menu item allows you to check that your changes do not break this rule. You can navigate through incorrect changes using a special **Incorrect Change** toolbar with **Next** and **Previous** buttons.

View Invalid XML Comments Shows unsupported tags in the XML comment file.

Exploring the Documenter Style Toolbar

The **Documenter** toolbar allows you to easily apply a link style, making link creation a matter of two clicks. Select a word or phrase and click a tool button to format it with a link style. Doc-To-Help takes care of the rest when you build a Help target. For more information on the link styles, see [Using Documenter Styles to Create Links](#) (page 484).

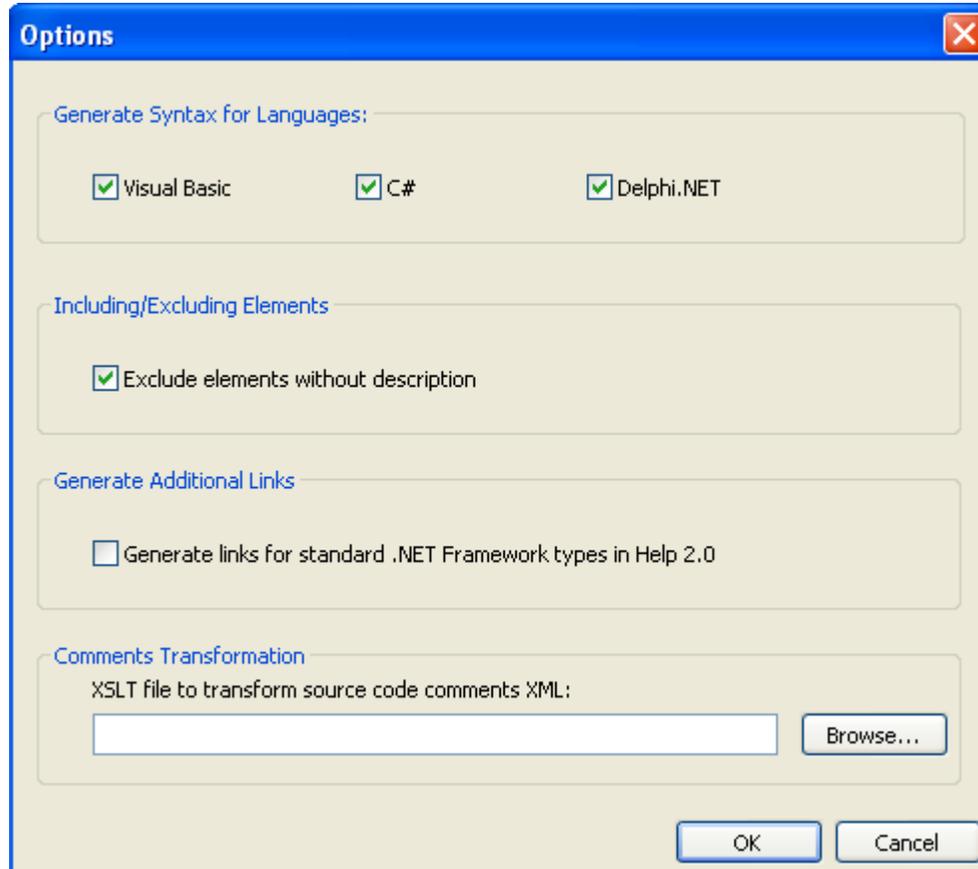


	Heading 1 Reference	Format text with Heading 1 Reference style.
	Heading 2 Reference	Format text with Heading 2 Reference style.
	Heading 3 Reference	Format text with Heading 3 Reference style.
	Heading 4 Reference	Format text with Heading 4 Reference style.
	Heading 5 Reference	Format text with Heading 5 Reference style.
	Link Namespace	Format text with Link Namespace style.
	Link Class	Format text with Link Class style.
	Link Interface	Format text with Link Interface style.
	Link Structure	Format text with Link Structure style.
	Link Enumeration	Format text with Link Enumeration style.
	Link Delegate	Format text with Link Delegate style.
	Link Constructor	Format text with Link Constructor style.
	Link Method	Format text with Link Method style.
	Link Operator	Format text with Link Operator style.
	Link Property	Format text with Link Property style.
	Link Field	Format text with Link Field style.
	Link Event	Format text with Link Event style.
	Link Topic	Format text with Link Topic style.
	Link Tag	Format text with Link Tag style.
	Show Hide Tag of Link	Show or Hide the tag segment of the link.
	Clear Formatting	Removes any formatting, returning the text to the style it was before it was formatted.

Changing the Project Options

Using Documenter's project **Options** dialog box, you can specify for which programming languages you want syntax automatically generated, whether to exclude elements that do not have descriptions provided in the assembly or XML file, and if you want links generated for standard .NET Framework types when building a Help 2.0 target.

To access the **Options** dialog box, in your Documenter project, select **Options** from the **Project** menu.



By default, Doc-To-Help automatically generates syntax for your reference section in Visual Basic and C# programming languages. You may want syntax generated for Delphi.NET as well or only in one language. Check the desired language(s) under *Generate Syntax for Languages*.

Documenter excludes elements without a description in the assembly or XML documentation file from the automatically generated reference document. To include all elements regardless of whether a description has been provided, uncheck the *Exclude elements without description* checkbox under **Including/Excluding Elements**.

Documenter automatically generates links for the standard .NET Framework classes and types, such as the **System.String** class, in Help 2.0 targets by default. When integrating your Help 2.0 targets with Microsoft Visual Studio, the links jump to the MSDN Help. If you do not plan to integrate your Help 2.0 Help file with Visual Studio, you must uncheck the *Generate links for standard .NET Framework types in Help 2.0* under **Generate Additional Links** in the **Options** dialog box. Otherwise, when the Help 2.0 target is built, the links will be broken because they cannot jump to the MSDN Help since your Help 2.0 is not integrated.

If you want to use non-standard tags or attributes in your source code XML comments, you can create an XSLT transformation file (.XSLT) and attach it here under **Comments Transformation**. For more information, see [Adding User-defined Tags and Attributes to XML Comments](#) (page 477).

Examining the Automatically Generated Document and Project

Once Documenter produces a reference document, it is a good idea to take a look at it to see just what is included. This topic describes the major automation features included in the output document.

After opening the document, place your cursor on any heading text and click the style drop-down. Note that all of the paragraphs are formatted with a style starting with **Auto**. Documenter uses these styles during the regeneration process to distinguish auto-generated paragraphs from those added by the user.



Note: It is the user's responsibility to avoid using styles starting with **Auto** in user added paragraphs. You can delete any auto-generated paragraphs you don't want to appear in the document, but cannot modify auto-generated paragraphs. More exactly, Documenter will overwrite modifications you perform to auto-generated paragraphs if it changes those paragraphs during regeneration. You are also not allowed to break generated paragraphs by adding your paragraph in the middle of an auto-generated paragraph. You must always add your paragraphs before or after an auto-generated paragraph.

Place your cursor on any link text (blue text) and click the style drop-down. Note that all of the topic links are formatted with styles starting with **Link**. These styles, and the related [toolbar and a menu items](#) (under the **Formatting** menu) are used to format link styles and to perform other frequently needed actions. These controls are provided in the **C1H_dotnet_src.dot** author template. Corresponding target templates for building different Doc-To-Help targets are **C1H_dotnet_html.dot** (HTML-based targets), **C1H_dotnet_hlp.dot** (WinHelp), **C1H_dotnet_prn.dot** (Manual).

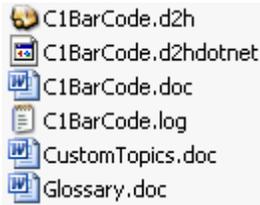


Every paragraph in the generated document starts with a hidden text that looks like this:

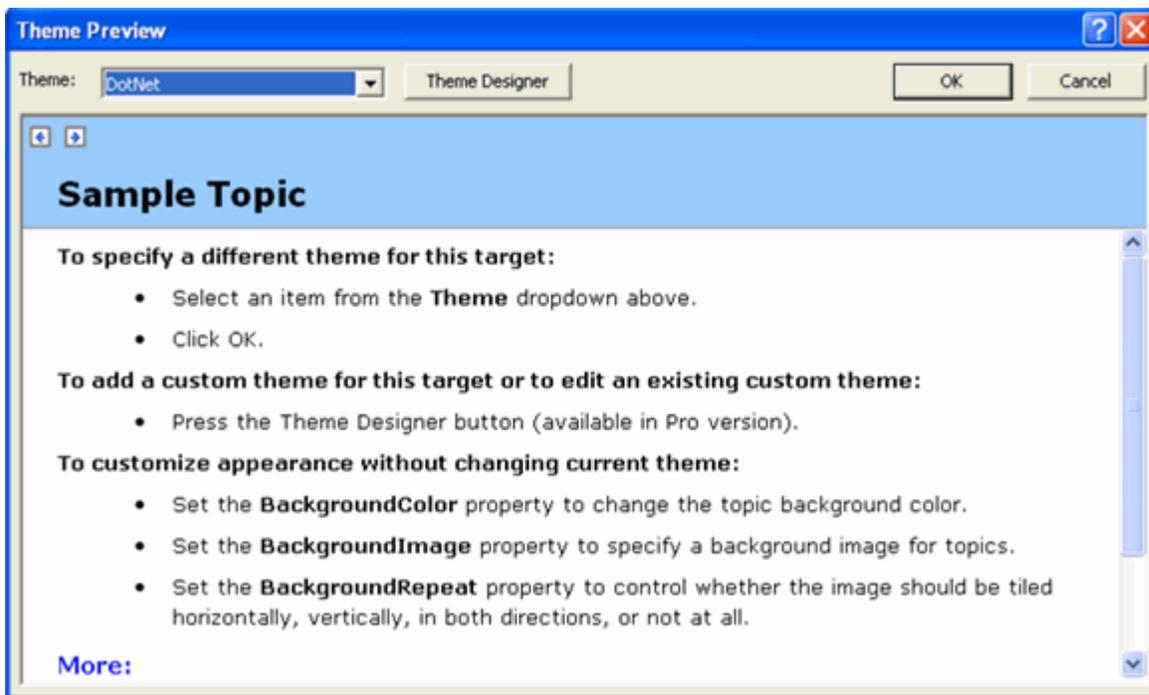
```
D2HDCmtr_: Types_Type: TargetAssemblyT: TargetAssembly. AFirstClassDerived%
```

This is the "block key". Documenter uses it for internal purposes, to make document regeneration possible. Normally, the block key should be ignored by the user. Just hide hidden text to remove it from the screen.

In addition to the .doc file, Documenter also creates a .log file in the project directory each time it generates or regenerates a document. This file must be kept unchanged in the project directory. Documenter needs it to be able to regenerate the document.



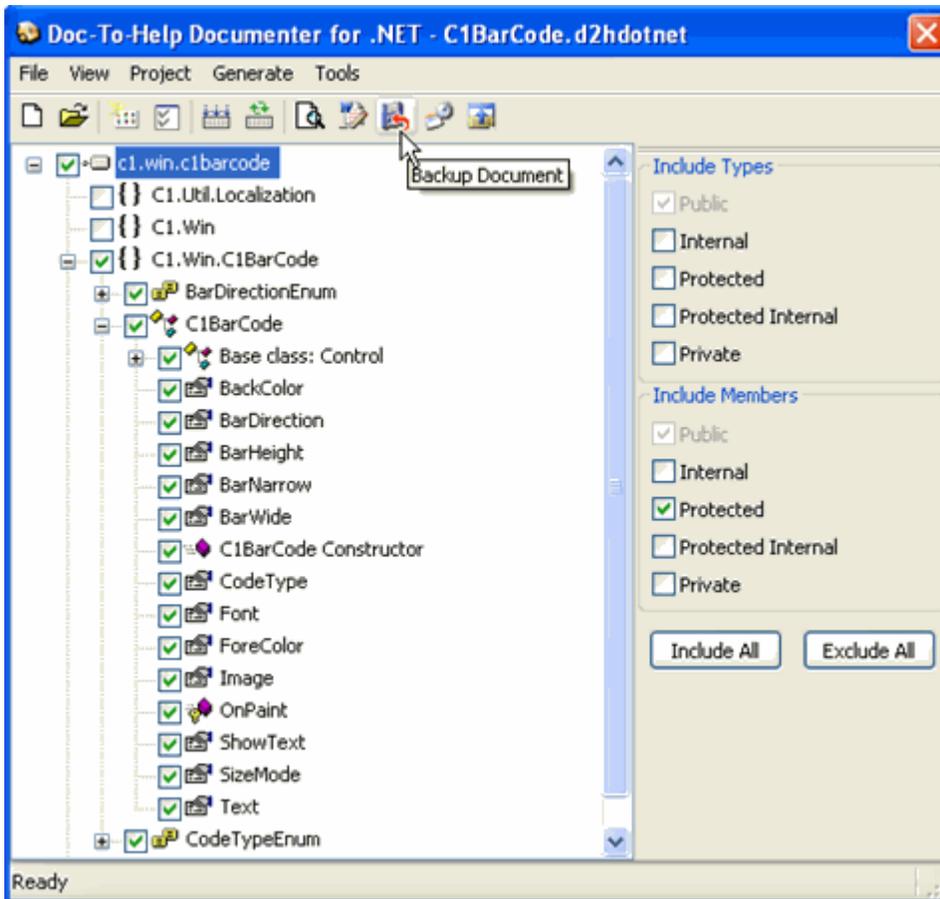
The targets in the Doc-To-Help project have their Theme property set to a special theme ".NET". You can use Doc-To-Help Theme Designer to modify that theme or to create a new theme and use it in the Doc-To-Help project.



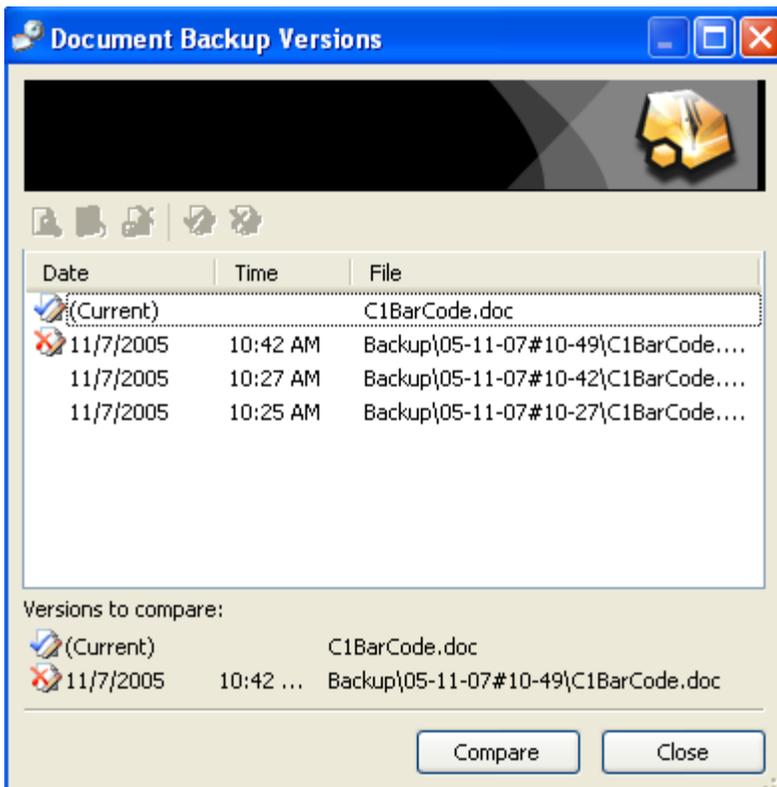
Adding Narrative to a Documenter Project

Documenter allows you to add narrative topics to the documentation and keeps those sections intact when your component evolves and you rebuild the documentation (by selecting the updated assemblies). This is a unique feature that provides seamless integration between the reference and narrative parts of the documentation.

Another important feature of **Documenter** is the ability to archive and compare versions of your automatically produced documentation. By simply clicking the **Backup Document** button , you can store a "snapshot" of your document before rebuilding with an updated .NET assembly.



By clicking the **Document Backup Version** button , you can view and compare your stored back up versions.



In addition to adding, retaining and archiving the narrative text in your Doc-To-Help project, you can insert additional Word documents into your Doc-To-Help project. Any Word document added to a Doc-To-Help project has complete access to all of the Doc-To-Help features, including the topic link styles. For more information on using the topic link styles, see [Adding Topic Links to a Doc-To-Help Project](#) (page 479) and [Using Doc-To-Help Styles to Create Links](#) (page 484).

Adding Formatting and Links to XML Comment Text

Writing comments in your source code that are included in the documentation through the XML comment file, you can use special tags supported by Doc-To-Help to make your comment text richly formatted and include any links and other help hot spots supported by D2HML.

To format a range of text in your XML comment with any style available in the generated document, use the `` tag with `style` attribute. Since it allows you to use all styles, you have the full power of D2HML for creating hot spots of any kind. You can create topic links, keyword links, expanding text and other kinds of hot spots, see D2HML description for details. Note that although standard XML comment tag `<see>` allows you to create links to assembly element topics, it does not allow you to create links to narrative topics in your help. Using the `` tag you can easily overcome this limitation; in fact, the `` tag allows your XML comment text to use all Doc-To-Help features available in Word and HTML source documents via D2HML.

For example, you can add a link to a Doc-To-Help topic like this:

```
<remarks>These are my remarks and there is a link to <span style="C1H Jump">this topic|tag=mytopic</span> here</remarks>
```

You can also use the `` tag to format your text with styles to give it the appearance you desire.

Doc-To-Help also supports two special tags for formatting for appearance, `` and `<i>`, used for bold and italic text. For example,

```
<remarks>These are my <b>bold</b> remarks</remarks>
```

You can also add your own tags and attributes to the XML comments. See [Adding User-defined Tags and Attributes to XML Comments](#) (page 477) for more information.

Supported XML Comment Tags

Doc-To-Help supports all standard XML comment tags, allowing you to easily format text in the XML file used to generate your reference source document. For a complete list of the tags, see the [Microsoft documentation](#).

In addition, Doc-To-Help also supports the non-standard XML comment tags supported by the free tool NDoc. NDoc, like Doc-To-Help, uses .NET assemblies and XML documentation files to automatically generate MSDN-style reference documentation. The non-standard tags supported by NDoc include:

<note>	The <code><note></code> tag produces a formatted note block.
<overloads>	The <code><overloads></code> tag provides documentation that applies to all the overloads of a member.
<exclude>	The <code><exclude/></code> tag directs NDoc to exclude the current item from documentation.
<preliminary>	The <code><preliminary></code> tag marks the documentation for the current item as preliminary.
<event>	The <code><event></code> tag describes which events can be raised by the current method.
<threadafety>	The <code><threadafety></code> tag is used to describe how a class or structure behaves in multi-threaded scenarios.

Adding User-defined Tags and Attributes to XML Comments

Doc-To-Help for .NET supports any custom user-defined tags and attributes in XML comments. This is done through an optional XSLT transformation that can be specified in a Doc-To-Help project.

If you want to use some non-standard tags or attributes in your source code XML comments, you can create an XSLT transformation file that transforms the source code comments XML file containing the non-standard tags and attributes into an XML file that does not, or one that contains only tags and attributes supported by Documenter for .NET. If you specify such an XSLT transformation file in your Documenter project, Documenter will apply it to the source code comments XML file before using the XML file for document generation. You can see the transformed XML file after generation in the same directory with the source code XML file; its extension is .XMLT.

To specify an XSLT transformation file in a Documenter project:

1. In Documenter, select **Options** from the **Project** menu. The **Options** dialog box appears.
2. Under **Comments Transformation**, click **Browse** and, once located, select the .XSLT file.
3. Click **Open** to close the **Open** dialog box and click **OK** to close the **Options** dialog box.

Documenter for .NET comes with one example of an XSLT, c1documenter.xslt, which can be found in the C:\Program Files\ComponentOne\DocToHelp\Documenter\Transforms folder. Note that this folder may be located elsewhere if you changed the default path during the Doc-To-Help installation.

This XSLT allows you to use a custom 'lang' attribute in the <code> tag. For example, the following XML comments:

```
<code lang="Visual Basic">
    C1Barcode1.CodeType = C1.Win.C1Barcode.CodeTypeEnum.Code39
    C1Barcode1.Text = "123456"
    PictureBox1.Image = C1Barcode1.Image
</code>
<code lang="C#">
    c1barcode1.CodeType = CodeTypeEnum.Code39;
    c1barcode1.Text = "123456";
    pictureBox1.Image = c1barcode1.Image;
</code>
<code lang="Delphi">
    begin
        c1barcode1.CodeType := CodeTypeEnum.Code39;
        c1barcode1.Text := '123456';
        pictureBox1.Image := c1barcode1.Image;
    end;
</code>
```

will produce the following content in the generated document:

- [Visual Basic](#)

```
C1Barcode1.CodeType = C1.Win.C1Barcode.CodeTypeEnum.Code39
C1Barcode1.Text = "123456"
PictureBox1.Image = C1Barcode1.Image
```
- [C#](#)

```
c1barcode1.CodeType = CodeTypeEnum.Code39;
c1barcode1.Text = "123456";
pictureBox1.Image = c1barcode1.Image;
```
- [Delphi](#)

```
begin
    c1barcode1.CodeType := CodeTypeEnum.Code39;
    c1barcode1.Text := '123456';
    pictureBox1.Image := c1barcode1.Image;
end;
```

which will be shown in the help target as three dropdown text hotspots that, when clicked, show code examples in three different programming languages:

- [Visual Basic](#)



```
C1Barcode1.CodeType = Cl.Win.C1Barcode.CodeTypeEnum.Code39
C1Barcode1.Text = "123456"
PictureBox1.Image = C1Barcode1.Image
```

- [C#](#)

```
clbarcode1.CodeType = CodeTypeEnum.Code39;
clbarcode1.Text = "123456";
pictureBox1.Image = clbarcode1.Image;
```

- [Delphi](#)

```
begin
  clbarcode1.CodeType := CodeTypeEnum.Code39;
  clbarcode1.Text := '123456';
  pictureBox1.Image := clbarcode1.Image;
end;
```

Adding Topic Links to a DocuMenter Project

Unlike other documentation tools, Doc-To-Help DocuMenter for .NET automates the entire process of creating program documentation, not just the reference materials, but the entire document set. Other documentation tools help to generate the reference sections of your documentation only. But quality software documentation contains additional documentation (narrative sections, conceptual sections, tutorials, etc.) in addition to reference materials. DocuMenter takes the next step by helping you produce these additional documents (which frequently have multiple links to the reference materials) by automating the topic link process.

By using the new [link styles](#), creating topic links is a simple matter of formatting text. Because narrative sections usually have numerous links to the reference sections of the documentation, using the link style feature makes it easy for you to add topic jumps as you write, without worrying about dialog boxes, selecting topics, etc.

1. To add a link to the topic below:

C1Barcode.BackColor Property

Gets or sets the background color for the control.

2. Select the location and text for the link.

C1Barcode.BarNarrow Property

Gets or sets the width of the narrow bars in screen pixels.

For more information on the colors, see [C1Barcode.BackColor](#).

3. Click the **Link Property** style button .

In this case, we selected the **Link Property** button because the topic we are linking to describes a property. If we were linking to a topic that describes a method, we would select **Link Method** button , for a delegate, we would select **Link Delegate** button , etc.

4. Build the Help target.

Note that the link is in the Help target, minus the **C1Barcode**, which is stripped out by DocuMenter.

C1Barcode.BarNarrow Property

Gets or sets the width of the narrow bars in screen pixels.
For more information on colors, see [BackColor](#).

[Visual Basic]
Public Property BarNarrow As Integer

[C#]
public int BarNarrow {get;set;}

Remarks

By default, narrow bars are one screen pixel wide (1/96") and wide bars are two pixels wide (1/96"). You should never have to change these values, they are exposed only in case you need to make adjustments for special barcode reader hardware.

See Also

[C1Barcode Class](#) | [C1Barcode Members](#) | [C1.Win.C1Barcode Namespace](#)

For more information on Link Styles, see [Using Documenter Styles to Create Links](#) (page 484).

Using the Link Tag Style

In some instances, it may be necessary to insert a topic link using text that does not correspond to the topic text. In this case, you can use the **Link Tag** style.

1. To add a link to the topic below:

C1.Win.C1Barcode Hierarchy

.....
System.Object
.....
System.MarshalByRefObject

2. Select the location and text for the link.

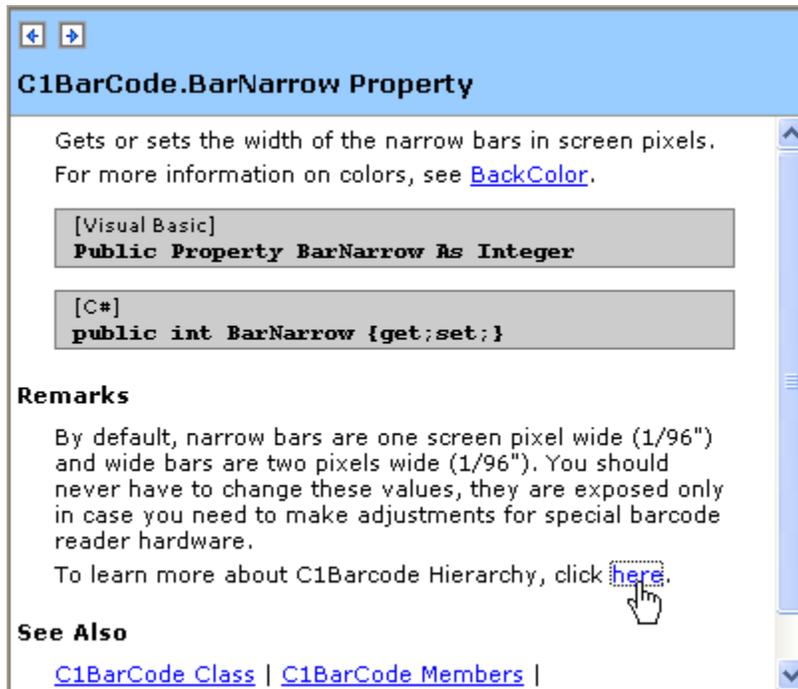
To learn more about C1Barcode Hierarchy, click [here@C1 Win C1Barcode Hierarchy](#).

You will note that we have specially formatted this text to use the **Link Tag** style. The formatting is:

```
j ump text@j ump topi c
```

In this case **Here** is the jump text, then the @ symbol (without spaces), then **C1 Win C1Barcode Hierarchy** is the jump topic. (Replace all "dots" (periods) with spaces.)

3. Click the **Link Tag** style button .
4. Build the Help target. Note that the link is in the Help target, minus **@C1 WinC1Barcode Hierarchy**, which is stripped out by Documenter.



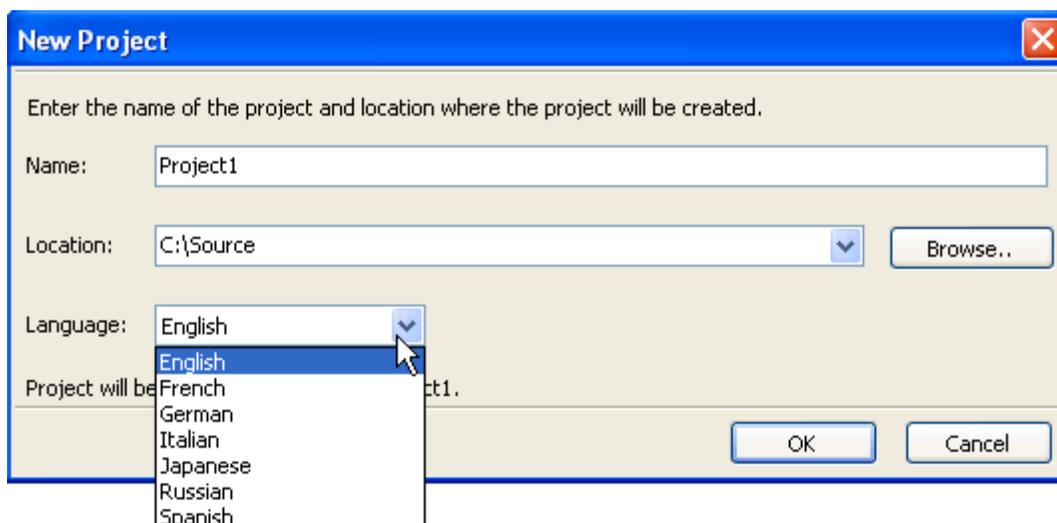
In cases where you want to see how your document will look without the links, select the link tag and click the **Show/Hide Tag of Link**  on the **Doc-To-Help Docucenter for .NET** toolbar. To make the tag part reappear, press the **Show/Hide Tag of Link** button again.

To hide or show the tag parts for all Link Tag links in the document, press **Show/Hide Tag of Link** while holding the **Ctrl** key.

For more information on Link Styles, see [Using Docucenter Styles to Create Links](#) (page 484).

Multilanguage Support in Docucenter for .NET

Docucenter for .NET can generate documentation in English, French, German, Spanish, Italian, Russian and Japanese. And with custom localization, you can generate output in any other language, see [Customization and Localization of Docucenter for .NET](#) (page 482). When creating a Docucenter project (or updating from a previous Docucenter version), you can choose the language of your project. The figure below illustrates this.

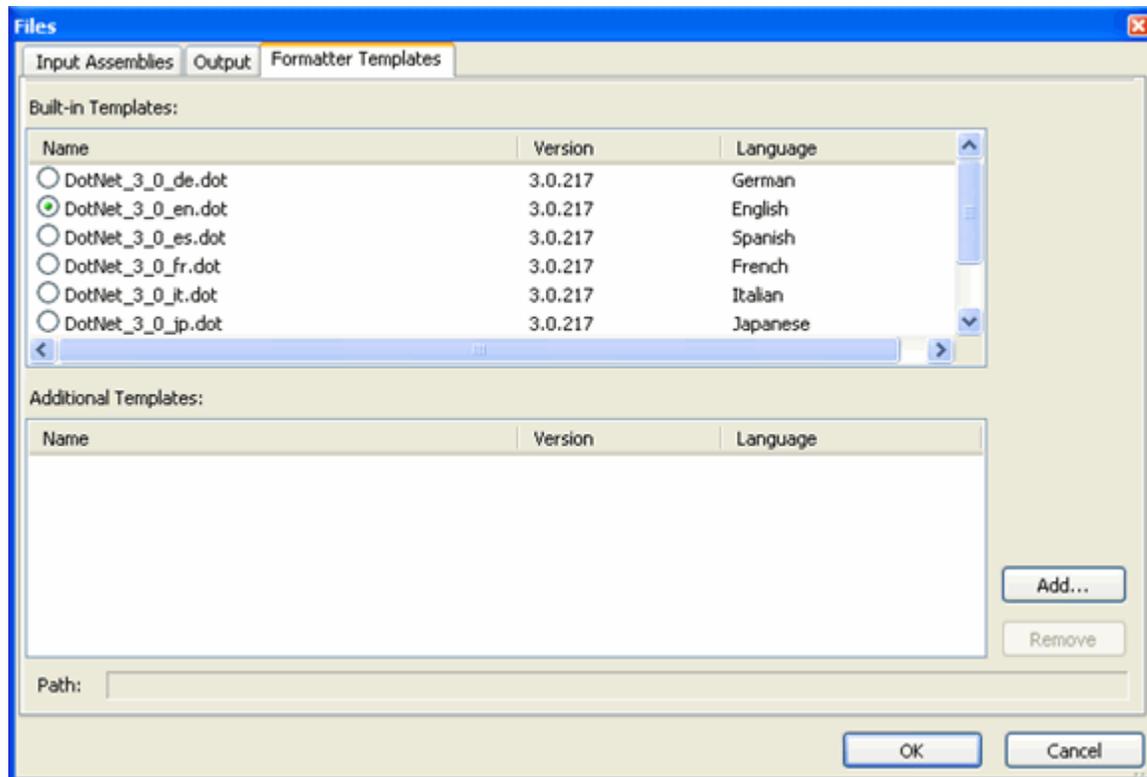


Customization and Localization of Documenter for .NET

You can customize Documenter for .NET output by modifying special templates called *formatter templates*. You can change and add text, formatting, graphics, etc. You can also use this customization feature to localize the output so it is produced in a language not covered by the built-in choice of languages (see [Multilanguage Support in Documenter for .NET](#) (page 481)).

Understanding the Formatter Templates

Documenter for .NET uses a “formatter template” to generate the document. The formatter template used in a project is specified in the project Files dialog on the Formatter Templates tab.



Documenter for .NET comes with a built-in formatter template:

- **Formatter Version 3.0 (all supported languages, each language in a separate template):** this template is enabled in projects created with this formatter version, version 3.0.

In addition to the built-in template, you can create your own formatter templates to customize **Documenter** output. You can start with the built-in template and modify it as necessary.

Do not change a built-in template. Always copy it to a new file and rename it.

Documenter addresses formatter templates by file name alone. The file names (without path) of all formatter templates on your computer must be unique. When you have created a new formatter template (or received it from someone else), you must add it to the Documenter in order to use it. This is done in the Files dialog’s Formatter Templates tab with the **Add** button adding templates to the Additional Templates list. This list is global to all Documenter projects on your computer. If you want to copy a Documenter project that uses a custom formatter template to another computer, you must copy the template with the project and then add it to the Files dialog on the destination computer.

Understanding Formatter Template "Blocks"

A Formatter template consists of multiple “blocks”. Each block is marked with a bookmark (the bookmark contains the entire block). Bookmark names must be the same as in the original built-in template. A customized template must contain all bookmarks (blocks) present in the original. Bookmark names have the prefix “D2HDCmtr_” and the

remainder of the name represents the part of the output (usually one paragraph) generated using this block of the template.

To see the blocks with their bookmarks, check the **Bookmarks** check box on the **View** tab of the **Word Tools | Options** dialog. Use the **Bookmark** dialog (**Insert | Bookmark** in Word menu) if you need to locate a bookmark.

Customizing Formatter Template

ShowExample Button:

The Formatter template has a **Formatter Help** toolbar that contains the **ShowExample** button. By using this button, you can easily find out, for any block, the place in the generated document that this particular block generates. Position on the block you are interested in and press the **ShowExample** button. This will show a sample generated project in a separate Word window with selection highlighting text generated by this block.

Variables:

Most blocks contain “variables”. Variables are enclosed in angle brackets (see figure).

<Name>

When generating the output, DocuMenter substitutes values for these variables. **Do not change the variables in the formatter template. They are hardcoded in the software; changing them will disrupt generation.**

You may delete a variable if you need to (although it is rarely necessary), but do not change them. All constant words, phrases and other content that may need to be changed to customize or localize a template is in the constant parts of the blocks, not in the variables. The styles with which variables are formatted in the template are also essential for proper generation. You can modify style attributes (font, etc.), but do not change the name of a style with which a variable is formatted. All other styles (in the constant parts of the blocks) can be used without restrictions. However, you must ensure that all paragraph styles used in the formatter template start with “Auto” because all generated content must be formatted with “Auto” styles to distinguish it from paragraphs added by the user.

Localizing Formatter Template

If you customize a formatter template in order to localize it, translate it to a language other than languages built-in in the DocuMenter, then you should consider using the `FormatterLocalizer` utility supplied with DocuMenter for .NET. Using this utility, localization is much easier than changing a formatter template by hand. This utility helps you to specify and maintain a list of translations for English phrases, and it automatically produces a localized formatter template using these translations.

The `FormatterLocalizer` utility and a document explaining how to use it are located in the Doc-To-Help installation directory in `DocuMenter\Customization\FormatterLocalizer`.

Read-made translations for all built-in languages are provided with the utility.

Restrictions in Changing Text in Formatter Template

Changing text in a formatter template, you must be aware that some text is used to create links, so it can not always be changed arbitrarily without breaking the links. This mostly concerns text used for topic titles. Some blocks are interdependent in the sense that changing a block you must change its dependent blocks correspondingly, to avoid breaking links. Some (not many) blocks can only be changed if you make corresponding changes in the scripts in the Doc-To-Help project. For example, if you change title “MyProperty Property” to “MyProperty (property)”, then links for MyProperty in the text will be broken, because the scripts in the Doc-To-Help project assume a property topic title has a certain format.

For the complete list of rules and restrictions in changing formatter template text, with complete lists of dependent blocks, see `DocuMenter\Customization\Restrictions.doc` in the Doc-To-Help installation directory.

Formatter Versions in Documenter for .NET

When a new major version of Documenter for .NET is released, it usually has new features and improved formatting that you want to use in your projects. This happens transparently when you create a new project with the new version, you just benefit from new features and enhancements. However, there are some issues in upgrading an existing project to a new major Documenter version.

Upgrading

When you open an old project with the new **Documenter** version, you will be prompted to upgrade by creating a new project.

Normally, the upgrade is chosen, in which case a new project will be created with all your settings copied to the new project. The new project belongs to the new formatter version. Retain the old project for backup purposes.

The only time you would consider not upgrading to the new formatter version is when you have already entered many paragraphs of your own text to the generated document and do not want to repeat that in the new version. Because you have to generate the project from scratch after upgrading the project, your changes will not appear in the document until you re-add them. If your project does not contain manually-added content or if you do not mind re-adding the material, the upgrade is always recommended.

Should you choose not to upgrade, **Documenter** will continue working on your project just as the previous version did. **Documenter** supports several formatter versions, including the previous version allowing you to continue working on an old document without upgrading if you so choose.

Viewing a Formatter Version of Your Project

To view the formatter version of your project, open the **Files** dialog box, select the **Formatter Templates** tab and look at the selected formatter template. It shows Version in a three-number format: <major version>.<minor version>.<build number>. Formatters with different major version numbers are incompatible. However, both are supported by the same **Documenter** version. Incompatibility refers only to manually-added content in a document created with an old formatter. The manually-added content will not be automatically retained by the newer formatter (Regenerate Existing Document is disabled). Minor version and build number increase do not break formatter compatibility.

Using Documenter Styles to Create Links

ComponentOne Documenter for .NET automatically generates links in reference documentation using the Doc-To-Help character style mechanism. This allows the links to be specified as simple text in the document. The fact that a text is actually a link is determined by formatting the text with a pre-defined style. There is no additional information stored anywhere in the document or in the Doc-To-Help project that is needed to create the link. When compiling the help target, Doc-To-Help recognizes the link style and creates the required connection automatically. You can see links created by Documenter for .NET in the generated document as words and phrases formatting with styles such as Link Class, Link Method, Link Property, etc.

Documenter for .NET does not require your participation in or even knowledge of the process of creating links in the text it generates. However, you will need link styles if you want to use one of the most powerful features of Documenter for .NET, the ability to add your paragraphs, topics and whole documents and be able to easily specify links from your added contents to Documenter generated contents.

To create a link from your text to a generated topic (or to an added topic), all you need to do is to format a word or a phrase in your text (usually it is a name of a property, method, class, etc.) with an appropriate style. You don't need to search for a topic in some huge list box, or even ensure that your link text corresponds exactly to the target topic name. For example, to create a link from the text **MyProperty** to the topic **MyClass.MyProperty Property**, you simply format the word **MyProperty** in your text with **Link Property** style. Given that such links are usually very numerous in narrative topics, you can see how Documenter for .NET makes the task of writing narrative documentation much easier. It completely automates creating reference parts, and for narrative parts it automates the mundane task of creating links. It frees you up to do what is most important, write the narrative itself.

If you use the correct style and, optionally, specify your preferences using one of a few special characters described in [Special Characters in Link Text](#) (page 492), Doc-To-Help will automatically determine the link target topic and will optionally modify the link text according to your preferences. All you need to do to create a link is choose the correct link style and apply it to the text. Doc-To-Help makes it even easier for you providing a special toolbar so you can easily apply a link style, making creating a link really a matter of two clicks, double-click a word, say, a method name, to select it, and click a toolbutton to format it with a link style. For more information, see [Documenter Controls and Toolbars](#) (page 470).

The following topics provide more information on the **Documenter for .NET** link styles and link functionality.

The Link Tag Style

Link Tag style is used to link to any topic regardless of the link text. The target topic title is specified in the second part of the link text, called the *tag part*, invisible in the help file.

Using this style, link text must consist of two parts separated with '@'. The first part is the link text as it will appear in the help file. The second part is the topic title specifying the link's target topic. Doc-To-Help removes the second part when it builds the help file.

To create a link using this style, select the link text and press **Link Tag**  on the **Doc-To-Help Documenter for .NET** toolbar. Then add '@' to the end and type the full topic title. Or you can first type '@' in the end and paste the topic title and then select the whole text and press the **Link Tag** toolbar button.

If you want to see how your document will look without the links, select the link tag and click the **Show/Hide Tag of Link**  on the **Doc-To-Help Documenter for .NET** toolbar. To make the tag part reappear, press the **Show/Hide Tag of Link** button again.

To hide or show the tag parts for all Link Tag links in the document, press **Show/Hide Tag of Link** while holding the **Ctrl** key.

The following table gives you some examples of using the Link Tag style:

In document	In help file	Links to topic
This is a link to FileInfo@FileInfo Class	This is a link to FileInfo	FileInfo Class
Another link to FileInfo@FileInfo Class	Another link to FileInfo	FileInfo Class
public FileInfo CopyTo(string);@FileInfo.CopyTo Method (String)	public FileInfo CopyTo(string);	FileInfo.CopyTo Method (String)

The Link Topic Style

Link Topic style is used to link to the topic whose title is an exact textual match with the link text.

To create a link using this style, select the link text and press **Link Topic**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, Doc-To-Help applies only the nested type separator substitution to the link text, see [Special Characters in Link Text](#) (page 492). Other special characters are not applicable to this style.

The following table gives you some examples of using the Link Topic style:

In document	In help file	Links to topic
FileInfo Class	FileInfo Class	FileInfo Class
System.Data Namespace	System.Data Namespace	System.Data Namespace
IResourceReader Members	IResourceReader Members	IResourceReader Members
Environment+SpecialFolder Enumeration	Environment.SpecialFolder Enumeration	Environment.SpecialFolder Enumeration

The Link Namespace Style

Link Namespace style is used to link the text representing a namespace name to the topic describing that namespace.

To create a link using this style, select the link text and press **Link Namespace**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, Doc-To-Help compiler will add the word **Namespace** to the text and establish a link to the topic with the title coinciding with the resulting text.

The following table illustrates the usage of the Link Namespace style:

In document	In help file	Links to topic
System	System	System Namespace
System.Data	System.Data	System.Data Namespace

The Link Class Style

Link Class style is used to link a class name to the topic describing that class.

To create a link using this style, select the link text and press **Link Class**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Class** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the qualifying namespace, if present, from the class name in the link text. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Class style:

In document	In help file	Links to topic
FileInfo	FileInfo	FileInfo Class
System.IO.FileInfo	FileInfo	System.IO.FileInfo Class
!System.IO.FileInfo	System.IO.FileInfo	System.IO.FileInfo Class
(System.IO.)FileInfo	System.IO.FileInfo	FileInfo Class

The Link Interface Style

Link Interface style is used to link an interface name to the topic describing that interface.

To create a link using this style, select the link text and press **Link Interface**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Interface** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the qualifying namespace, if present, from the interface name in the link text. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Interface style:

In document	In help file	Links to topic
IResourceReader	IResourceReader	IResourceReader Interface
System.Resources.IResourceReader	IResourceReader	System.Resources.IResourceReader Interface
!System.Resources.IResourceReader	System.Resources.IResourceReader	System.Resources.IResourceReader Interface
(System.Resources.)IResourceReader	System.Resources.IResourceReader	IResourceReader Interface

The Link Structure Style

Link Structure style is used to link a structure name to the topic describing that structure.

To create a link using this style, select the link text and press **Link Structure**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Structure** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes qualifying namespace, if present, from the structure name in the link text. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Structure style:

In document	In help file	Links to topic
Point	Point	Point Structure
System.Drawing.Point	Point	System.Drawing.Point Structure
!System.Drawing.Point	System.Drawing.Point	System.Drawing.Point Structure
(System.Drawing.)Point	System.Drawing.Point	Point Structure

The Link Enumeration Style

Link Enumeration style is used to link an enumeration name to the topic describing that enumeration.

To create a link using this style, select the link text and press **Link Enumeration**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Enumeration** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes qualifying namespace, if present, from the enumeration name in the link text. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Enumeration style:

In document	In help file	Links to topic
TraceMode	TraceMode	TraceMode Enumeration
System.Web.TraceMode	TraceMode	System.Web.TraceMode Enumeration
!System.Web.TraceMode	System.Web.TraceMode	System.Web.TraceMode Enumeration

In document	In help file	Links to topic
(System.Web.)TraceMode	System.Web.TraceMode	TraceMode Enumeration

The Link Delegate Style

Link Delegate style is used to link a delegate name to the topic describing that delegate.

To create a link using this style, select the link text and press **Link Delegate**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Delegate** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes qualifying namespace, if present, from the delegate name in the link text. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Delegate style:

In document	In help file	Links to topic
TypeFilter	TypeFilter	TypeFilter Delegate
System.Reflection.TypeFilter	TypeFilter	System.Reflection.TypeFilter Delegate
!System.Reflection.TypeFilter	System.Reflection.TypeFilter	System.Reflection.TypeFilter Delegate
(System.Reflection.)TypeFilter	System.Reflection.TypeFilter	TypeFilter Delegate

The Link Type Style

Link Type style is used to link a type name to the topic describing that type. A type can be a class, an enumeration, an interface, a structure or a delegate.

Note: Using this style is not necessary and not recommended. You can always use a specialized style instead, one of the following: **Link Class**, **Link Enumeration**, **Link Interface**, **Link Structure**, or **Link Delegate**. The **Link Type** style is used by Documenter for .NET for internal purposes, to simplify the process of generating the document, and is documented here only for completeness.

The link text starts with a special character followed by colon. The special character specifies the kind of the type:

- – Class
- D** – Delegate
- E** – Enumeration
- I** – Interface
- S** – Structure

For this style, the Doc-To-Help compiler removes the special character and colon. The target topic title is determined by the remainder of the link text with the added type kind name, which is one of the words **Class**, **Delegate**, **Enumeration**, **Interface** or **Structure**.

To create a link using this style, prepend the text with one of the special characters (uppercase) and colon, select the text including the special character and format it with the **Link Type** style using standard Microsoft Word style formatting.

For this style, Doc-To-Help applies all transformations determined by the special characters '!', '()' and '+', as for the corresponding specialized styles for type links, see [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Type style:

In document	In help file	Links to topic
C:FileInfo	FileInfo	FileInfo Class
I:IResourceReader	IresourceReader	IResourceReader Interface
S:Point	Point	Point Structure
E:TraceMode	TaceMode	TraceMode Enumeration
D:TypeFilter	TypeFilter	TypeFilter Delegate

The following table demonstrates using special characters with the Link Type style:

In document	In help file	Links to topic
C:FileInfo	FileInfo	FileInfo Class
C:System.IO.FileInfo	FileInfo	System.IO.FileInfo Class
C:!System.IO.FileInfo	System.IO.FileInfo	System.IO.FileInfo Class
C:(System.IO.)FileInfo	System.IO.FileInfo	FileInfo Class

The Link Field Style

Link Field style is used to link a field name to the topic describing that field.

To create a link using this style, select the link text and press **Link Field**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Field** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the name of the type containing the field, if the type name is present. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Field style:

In document	In help file	Links to topic
Timeout.Infinite	Infinite	Timeout.Infinite Field
!Timeout.Infinite	Timeout.Infinite	Timeout.Infinite Field
System.Threading.Timeout.Infinite	Infinite	System.Threading.Timeout.Infinite Field
!System.Threading.Timeout.Infinite	System.Threading.Timeout.Infinite	System.Threading.Timeout.Infinite Field
(System.Threading.)Timeout.Infinite	System.Threading.Timeout.Infinite	Timeout.Infinite Field

The Link Event Style

Link Event style is used to link an event name to the topic describing that event.

To create a link using this style, select the link text and press **Link Event**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Event** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the name of the type containing the event, if the type name is present. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492).

The following table demonstrates the usage of the Link Event style:

In document	In help file	Links to topic
Form.Load	Load	Form.Load Event
!Form.Load	Form.Load	Form.Load Event
System.Windows.Forms.Form.Load	Load	System.Windows.Forms.Form.Load Event
!System.Windows.Forms.Form.Load	System.Windows.Forms.Form.Load	System.Windows.Forms.Form.Load Event
(System.Windows.Forms.)Form.Load	System.Windows.Forms.Form.Load	Form.Load Event

The Link Property Style

Link Property style is used to link a property name to the topic describing that property.

To create a link using this style, select the link text and press **Link Property**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Property** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the name of the type containing the property, if the type name is present. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492). You can also use the argument list for property overloads with arguments, see [Argument List in Links](#) (page 493).

The following table demonstrates the usage of the Link Property style:

In document	In help file	Links to topic
FileInfo.Exists	Exists	FileInfo.Exists Property
!FileInfo.Exists	FileInfo.Exists	FileInfo.Exists Property
System.IO.FileInfo.Exists	Exists	System.IO.FileInfo.Exists Property
!System.IO.FileInfo.Exists	System.IO.FileInfo.Exists	System.IO.FileInfo.Exists Property
(System.IO.)FileInfo.Exists	System.IO.FileInfo.Exists	FileInfo.Exists Property

The Link Method Style

Link Method style is used to link a method name to the topic describing that method.

To create a link using this style, select the link text and press **Link Method**  on the **Doc-To-Help Documenter for .NET** toolbar.

For this style, the Doc-To-Help compiler will apply the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492), add the word **Method** to the text, and establish a link to the topic with the title coinciding with the resulting text.

By default, Doc-To-Help removes the name of the type containing the method, if the type name is present. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492). You can also use the argument list for method overloads, see [Argument List in Links](#) (page 493).

The following table demonstrates the usage of the Link Method style:

In document	In help file	Links to topic
File.Delete	Delete	File.Delete Method
!File.Delete	File.Delete	File.Delete Method
System.IO.File.Delete	Delete	System.IO.File.Delete Method
!System.IO.File.Delete	System.IO.File.Delete	System.IO.File.Delete Method
(System.IO.)File.Delete	System.IO.File.Delete	File.Delete Method

The Link Constructor Style

Link Constructor style is used to link a constructor name to the topic describing that constructor.

For this style, link text must consist of a type name followed by the word **Constructor**.

To create a link using this style, select the link text and press **Link Constructor**  on the **Doc-To-Help Documenter for .NET** toolbar.

By default, Doc-To-Help applies the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492). It also removes qualifying namespace, if present, from the class name in the link text. You can cancel removing the namespace by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492). You can also use the argument list for constructor overloads, see [Argument List in Links](#) (page 493).

The following table demonstrates the usage of the Link Constructor style:

In document	In help file	Links to topic
FileInfo Constructor	FileInfo Constructor	FileInfo Constructor
System.IO.FileInfo Constructor	FileInfo Constructor	System.IO.FileInfo Constructor
!System.IO.FileInfo Constructor	System.IO.FileInfo Constructor	System.IO.FileInfo Constructor
(System.IO.)FileInfo Constructor	System.IO.FileInfo Constructor	FileInfo Constructor

The Link Operator Style

Link Operator style is used to link an operator name to the topic describing that operator.

For this style, link text must consist of an operator name followed by the word **Operator** or **Conversion** defining the kind of the operator. The operator name must exactly correspond to the operator name in the title of the topic describing the operator.

To create a link using this style, select the link text and press **Link Operator**  on the **Doc-To-Help Documenter for .NET** toolbar.

By default, Doc-To-Help applies the nested type separator substitution to the link text, if applicable, see [Special Characters in Link Text](#) (page 492). It also removes the name of the type containing the operator, if the type name is present. You can change this behavior by using the special characters, exclamation sign and parenthesis. See [Special Characters in Link Text](#) (page 492). You can also use the argument list for operator overloads, see [Argument List in Links](#) (page 493).

The following table demonstrates the usage of the Link Operator style for regular operators:

In document	In help file	Links to topic
Point.Addition Operator	Addition Operator	Point.Addition Operator
!Point.Addition Operator	Point.Addition Operator	Point.Addition Operator

In document	In help file	Links to topic
System.Drawing.Point.Addition Operator	Addition Operator	System.Drawing.Point.Addition Operator
!System.Drawing.Point.Addition Operator	System.Drawing.Point.Addition Operator	System.Drawing.Point.Addition Operator
(System.Drawing.)Point.Addition Operator	System.Drawing.Point.Addition Operator	Point.Addition Operator

The following table demonstrates the usage of the Link Operator style for conversion operators:

In document	In help file	Links to topic
Point.Point to Size Conversion	Point to Size Conversion	Point.Point to Size Conversion
!Point.Point to Size Conversion	Point.Point to Size Conversion	Point.Point to Size Conversion
System.Drawing.Point.Point to Size Conversion	Point to Size Conversion	System.Drawing.Point.Point to Size Conversion
!System.Drawing.Point.Point to Size Conversion	System.Drawing.Point.Point to Size Conversion	System.Drawing.Point.Point to Size Conversion
(System.Drawing.)Point.Point to Size Conversion	System.Drawing.Point.Point to Size Conversion	Point.Point to Size Conversion

Special Characters in Link Text

There are three special characters that you can use in link text, in addition to link styles, to control the process of creating links: '!' (exclamation sign), '(' (parenthesis), '+' (plus). These characters have special meanings to the Doc-To-Help compiler (more exactly, to the scripts in the Doc-To-Help project created by Documenter for .NET). They are removed or replaced in the resulting help file.

By default, Doc-To-Help removes the qualifying namespace, if present, from type names in links to types. It also removes the qualifying type name from member names in links to members. So, for example, a link **System.Windows.Forms.CheckBox** formatted with Link Class style will become simply **CheckBox** in the help file. And a link **DataSet.Clone** formatted with Link Method style will become simply **Clone** in the help file. This is done so you can use a namespace or type name to qualify your link so it points to the correct topic. For example, a class **CheckBox** also exists in the namespace **System.Web.UI.WebControls**, and many different classes have a **Clone** method, so you may need to qualify those links to ensure uniqueness, but you usually don't want the qualifier to appear in the help file text.

However, sometimes you may want the fully qualified name to appear in the link text. In that case, use one of the special characters, either exclamation sign or parenthesis.

An exclamation sign '!' at the start of the link text instructs Doc-To-Help to leave the full name as is, not to remove the namespace or type name qualifier. Special character '!' is supported in all link styles except **Link Namespace**, **Link Topic** and **Link Tag** where it is not applicable.

Enclosing the qualifier (including the dot in the end) in parenthesis has a similar effect. It also stops the qualifier from being removed from the link text. But, unlike the exclamation sign, parentheses specify a different link target topic: the link points to a topic whose name does not contain the qualifier.

The following table illustrates the use of special characters '!' and '(' (link text in the document is formatted with Link Method style):

In document	In help file	Links to topic
File.Delete	Delete	File.Delete Method
System.IO.File.Delete	Delete	System.IO.File.Delete Method
!System.IO.File.Delete	System.IO.File.Delete	System.IO.File.Delete Method
(System.IO.)File.Delete	System.IO.File.Delete	File.Delete Method

The special character '+' (plus) is used only in nested type names (nested types are classes, enumerations and other types defined inside a class). It must be used in every nested type name in any link, wherever a type name can occur

(that includes links to nested types, links to members of nested types if they are qualified with a type name, and links to method overloads that have nested type names in their argument list). It separates the type name from the parent type name. The normal separator is dot ('.'), and indeed Doc-To-Help replaces '+' with '.' in the help file text, but '+' is necessary to tell Doc-To-Help where the parent type name ends and the type name itself starts.

This behavior, called the *nested type separator substitution*, is supported in all link types except **Link Namespace** and **Link Tag** where it has no meaning.

The following table illustrates the use of the special character '+' for a **SpecialFolder** enumeration nested in the Environment class (all examples are from Microsoft .NET Framework). It also demonstrates the other two special characters, '!' and '()'. Link text in the document is formatted with Link Enumeration style.

In document	In help file	Links to topic
Environment+SpecialFolder	Environment.SpecialFolder	Environment.SpecialFolder Enumeration
System.Environment+SpecialFolder	Environment.SpecialFolder	System.Environment.SpecialFolder Enumeration
!System.Environment+SpecialFolder	System.Environment.SpecialFolder	System.Environment.SpecialFolder Enumeration
(System.)Environment+SpecialFolder	System.Environment.SpecialFolder	Environment.SpecialFolder Enumeration

Argument List in Links

The following link styles can contain an argument list, in case it is necessary to distinguish between overloads with different arguments:

- Link Method
- Link Property
- Link Operator
- Link Constructor

If there are nested types in the argument lists, they undergo nested type separator substitution, see [Special Characters in Link Text](#) (page 492), that is, '+' is replaced with '.'. This is the only transformation applied to types in argument lists. Any types that are not nested remain unchanged. It means that the argument list in the link must exactly correspond to the argument list in the target topic title (except for the '+' sign in nested types, if nested types are present).

Important: Argument lists must be separated with a space from the preceding member name in the link.

The following table illustrates the use of argument lists in List Method links:

In document	In help file	Links to topic
File.Create (String, System.Int32)	Create (String, System.Int32)	File.Create Method (String, System.Int32)
System.IO.File.Create (String, System.Int32)	Create (String, System.Int32)	System.IO.File.Create Method (String, System.Int32)
!System.IO.File.Create (String, System.Int32)	System.IO.File.Create (String, System.Int32)	System.IO.File.Create Method (String, System.Int32)
(System.IO.)File.Create (String, System.Int32)	System.IO.File.Create (String, System.Int32)	File.Create Method (String, System.Int32)

Link Colors and Appearance

By default, links use the color blue and underline. You can change this, as with many other appearance and formatting style attributes, by modifying the Help target template used by Doc-To-Help. These templates are installed by Doc-To-Help in the Microsoft Word template directory (to locate that directory, use Word menu **Tools | Options | File Locations | User Templates**).

To see what template is used in building a help target, select the target in the Doc-To-Help project and look at the value of the **Template** property in the Help Targets property pane. For example, to modify the color or other appearance attributes of the links using the Link Class style in HTML Help, open the C1H_dotnet_html.dot template in Microsoft Word, find the Link Class style and modify it. You can do this with any style in any help target template. Here is the list of standard Documenter for .NET target templates for different targets:

- C1H_dotnet_hlp.dot – WinHelp target
- C1H_dotnet_prn.dot – Manual target
- C1H_dotnet_html.dot – HTML-based targets (HTML Help, HTML, Help 2.0, JavaHelp)

Blue underlined links are an MSDN standard, so it was chosen as default in Documenter for .NET. But you can change the colors as described above, or you can make links look like standard links in the corresponding target. To use the standard link appearance instead of the one specified by styles in target templates, set the **AffectsAppearance** property of the link styles to **False**. You can locate that property in Doc-To-Help on the Project icon properties pane.

For HTML-based targets viewed with Internet Explorer (or standalone Microsoft viewers, such as Help 2.0 or HTML Help), standard link colors are determined in Internet Explorer settings **Internet Options | General | Colors | Links**. For WinHelp, the standard link color is green with an underline.

Doc-To-Help 2006 Features

The following features are available with Doc-To-Help 2006.

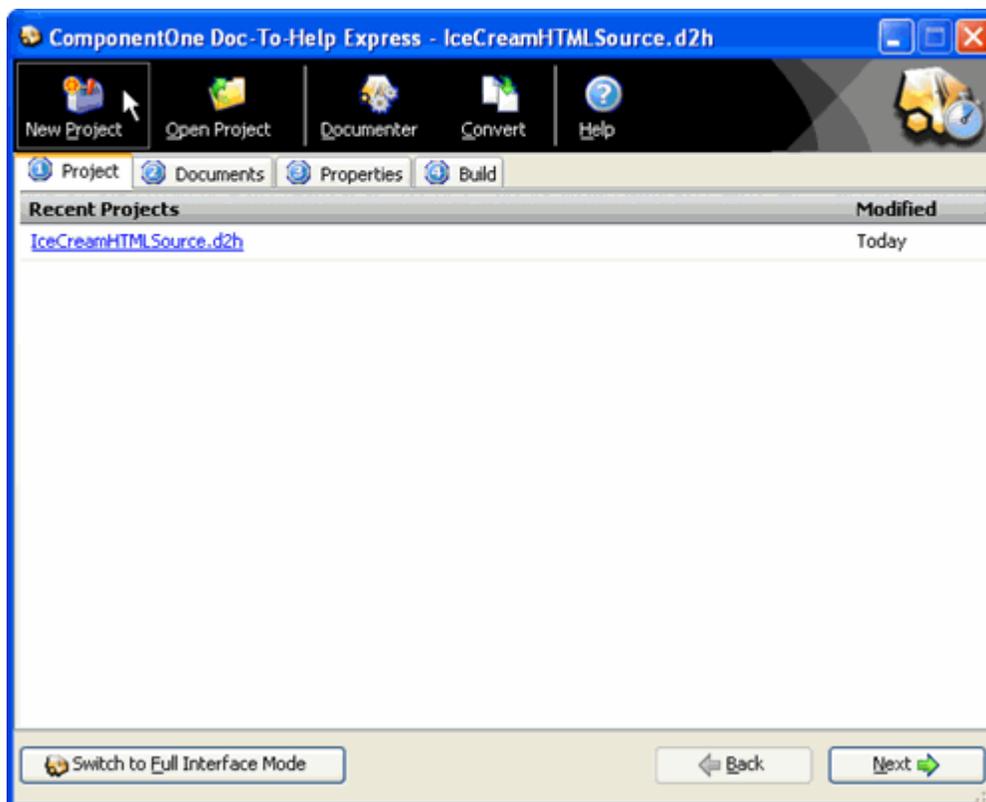
Doc-To-Help Express

Welcome to Doc-To-Help Express. Doc-To-Help Express is Doc-To-Help's simplified user interface. It uses most of the same features used in regular Doc-To-Help projects, providing you the functionality to easily create help systems. Doc-To-Help Express is designed to allow you to quickly transfer from the Express interface to the Doc-To-Help interface if you want the complete features of Doc-To-Help.

Creating a Doc-To-Help Express Project

Doc-To-Help Express allows you to open projects or create new ones.

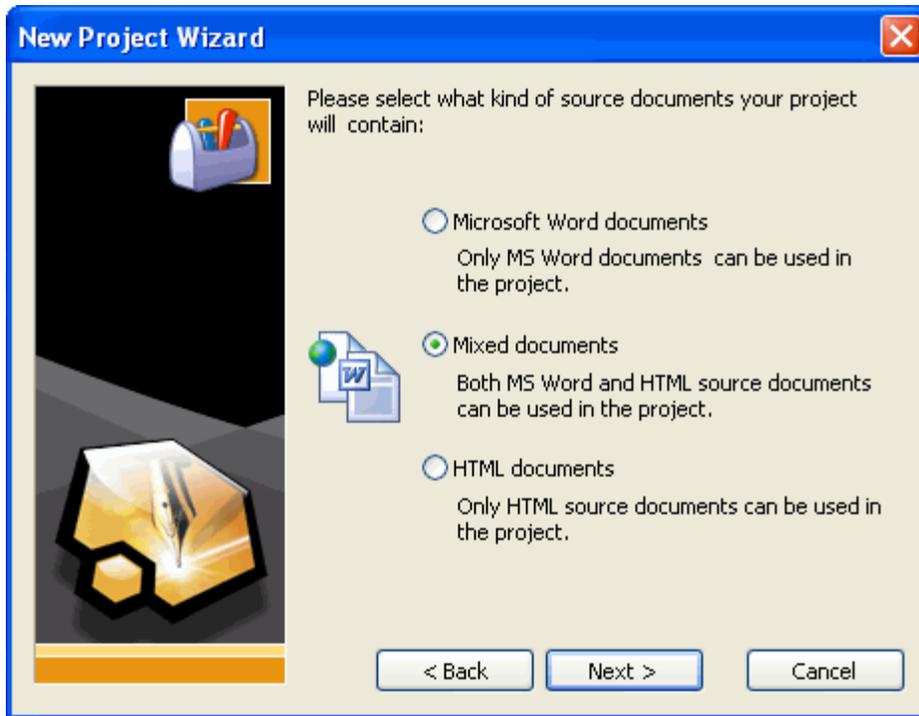
1. To create a new project, click the **New Project** button on the Doc-To-Help Express interface.



2. A dialog box appears asking you to browse to the location of your project.
3. Create a name for your Doc-To-Help Express project.

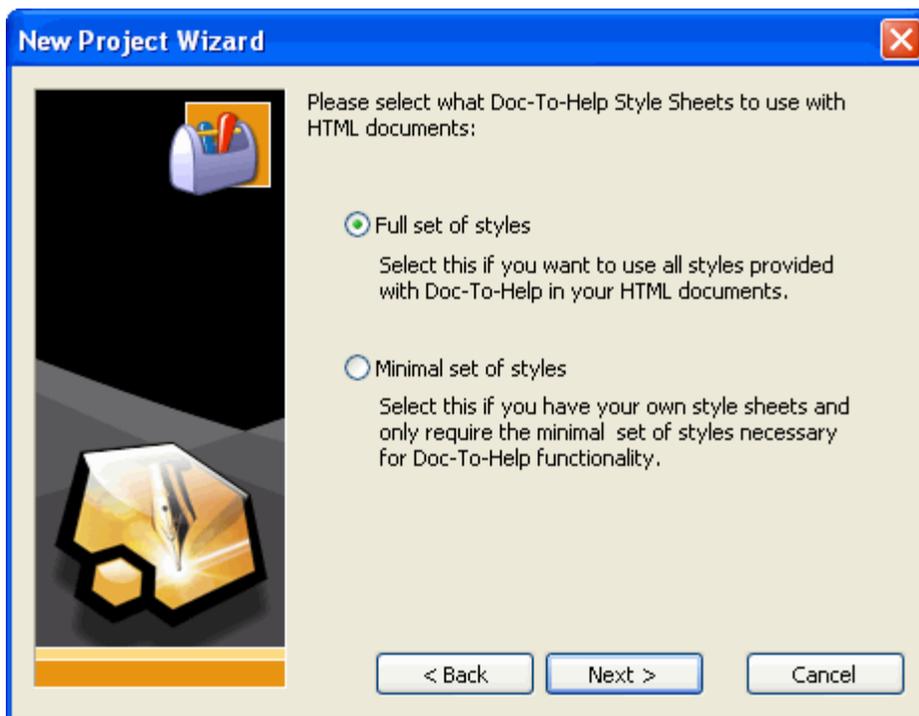
Note: No two Doc-To-Help projects can reside in the same folder; each project must have its own folder.

4. The **New Project Wizard** dialog asks you what source documents you plan to use in your project – **Microsoft Word documents** (Word documents only), **Mixed documents** (Microsoft Word documents and HTML source documents), or **HTML documents** (HTML documents only). For this particular project, select **Mixed documents**.



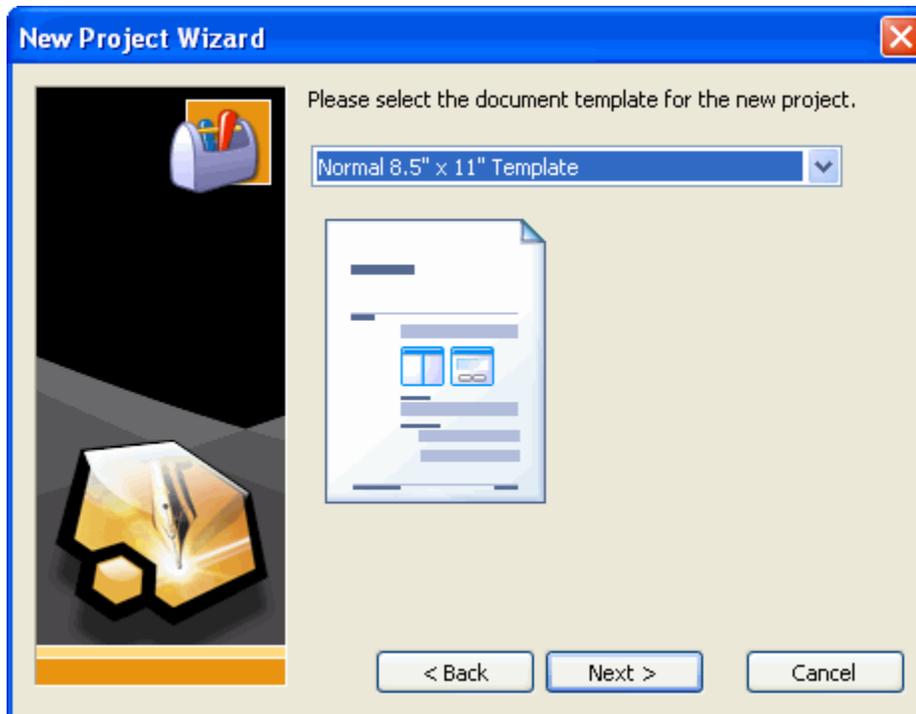
Note: If using mixed or HTML documents, Doc-To-Help will prompt you for the name of the default file for content. The **Adding HTML Documents** dialog box will appear allowing you to set the number of topics in the document and document style. For more information, see [Adding a Document to a Project](#) (page 191).

5. The **New Project Wizard** then asks the kind of Style Sheets you wish to use with the HTML documents – **Full set of styles** or **Minimal set of styles**.

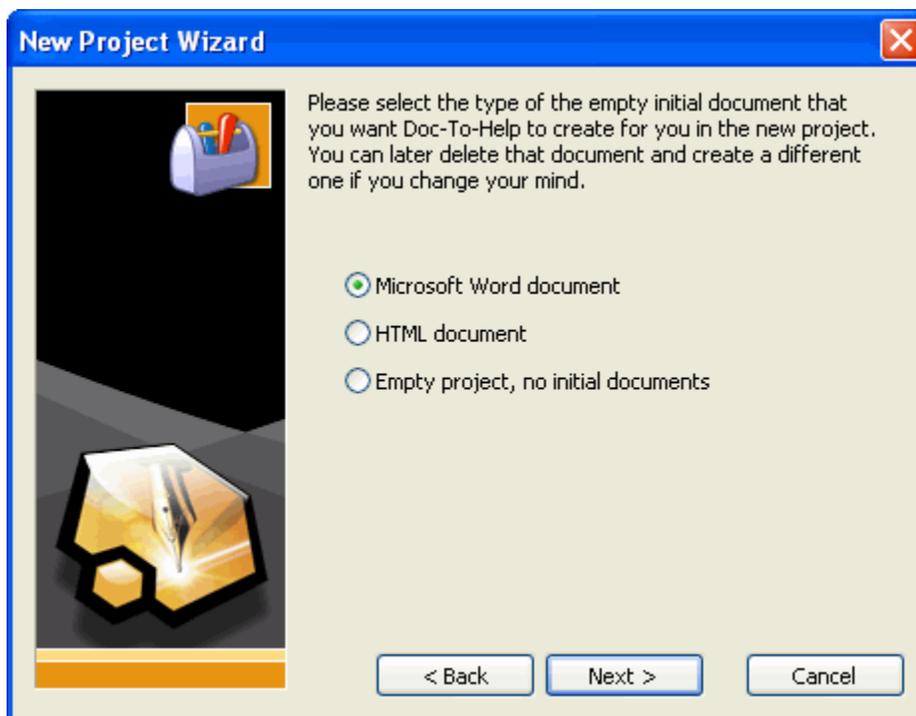


By selecting Full set of Styles, Doc-To-Help Express gives you full access to all of the styles available in Doc-To-Help.

- You are then instructed to select a template size and default help target.



- Select which type of file that you would like to begin working with, either a Microsoft Word document, an HTML document, or an empty project with no initial documents.



- Click **Finish** to create a new project file.

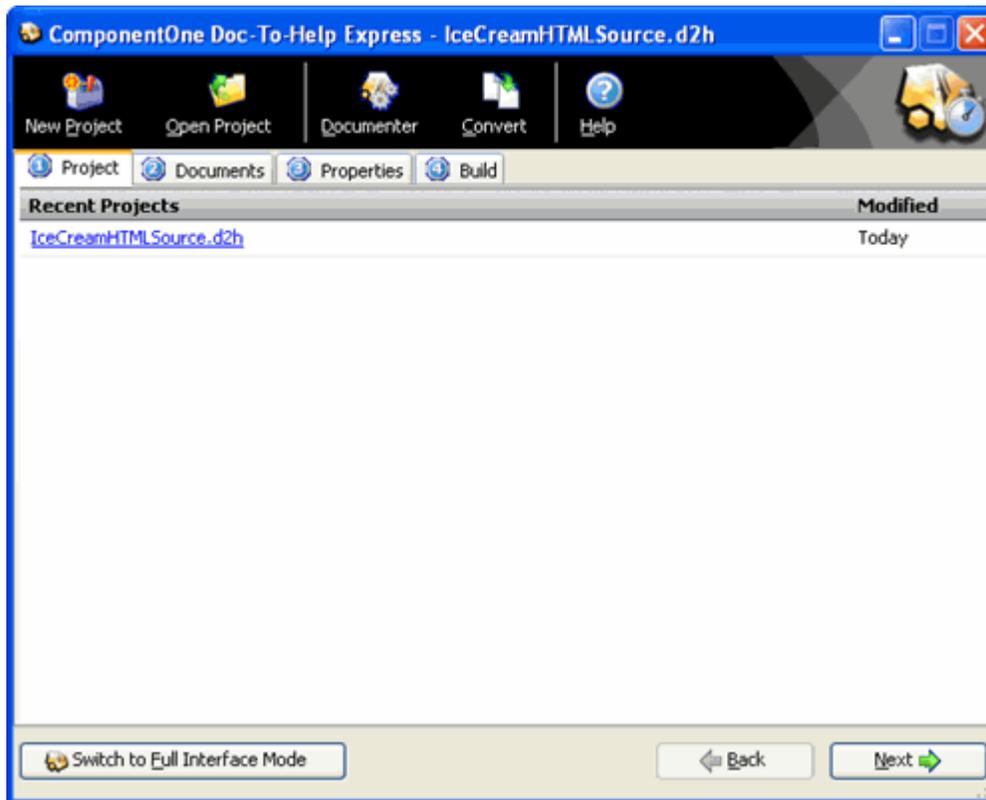
Using the Doc-To-Help Express Interface

Once you have created a new project, the Doc-To-Help Express Interface adjusts itself to include four tabs: **Project**, **Documents**, **Properties** and **Build**. The **Next** and **Back** buttons will navigate through each tab in creating a help target or you can skip ahead by clicking on the tab name.

Note: You are able to open Doc-To-Help 2006 at any point simply by clicking the **Switch To Full Interface Mode** button.

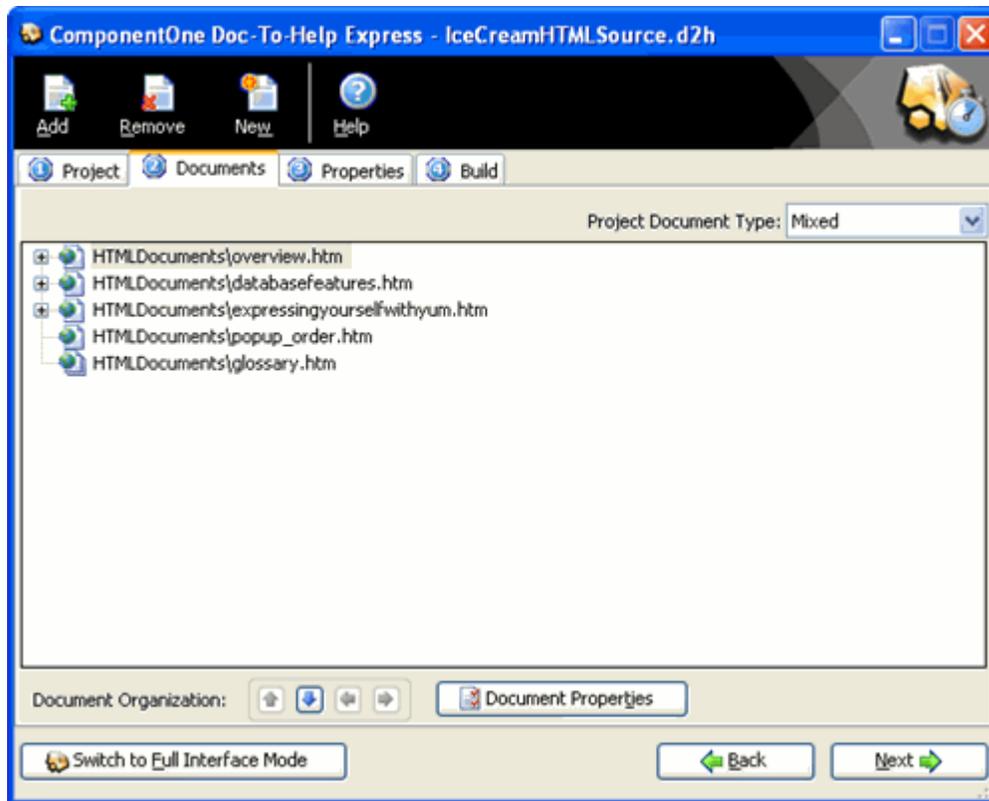
Project Tab

The **Project** tab lists all of the Doc-To-Help projects recently worked on, and gives the options of opening an existing Doc-To-Help project, creating a new one, opening Documenter for .NET or converting a RoboHelpHTML Help, or WinHelp project to Doc-To-Help.



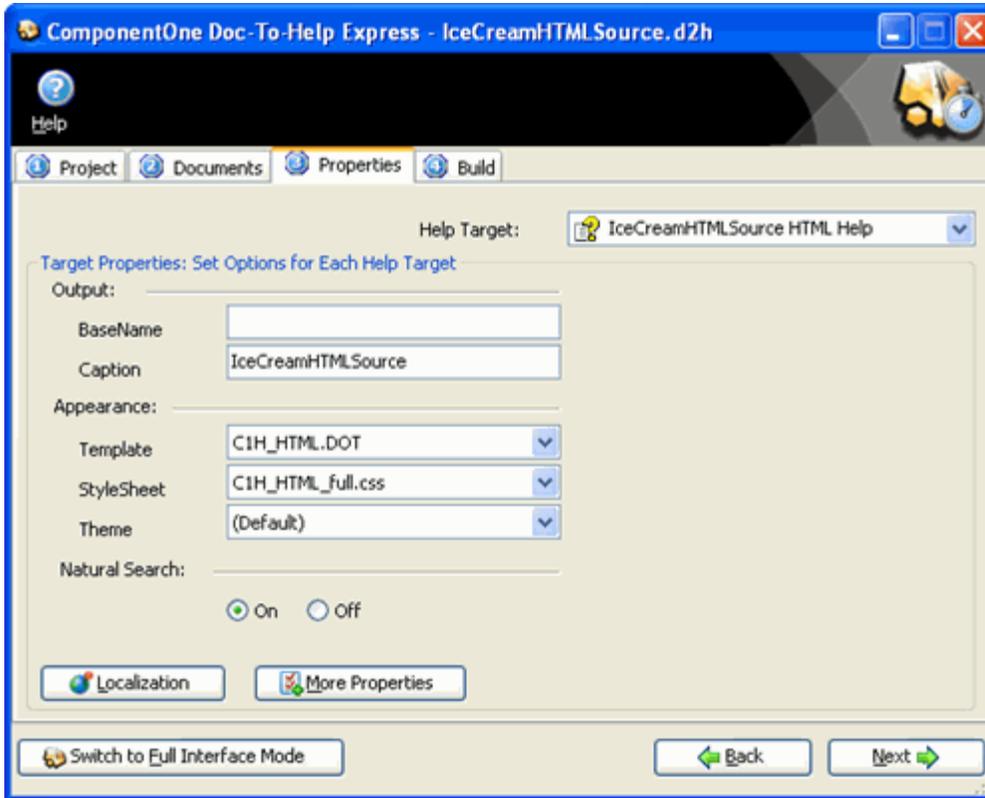
Documents Tab

The **Documents** tab lists all the source documents found in the project. On this tab, documents can be added, removed and created. The types of documents that the project can contain (HTML, Word or Mixed) can be changed to include or exclude a type of document. The document tree helps organize documents and topics in the table of contents.



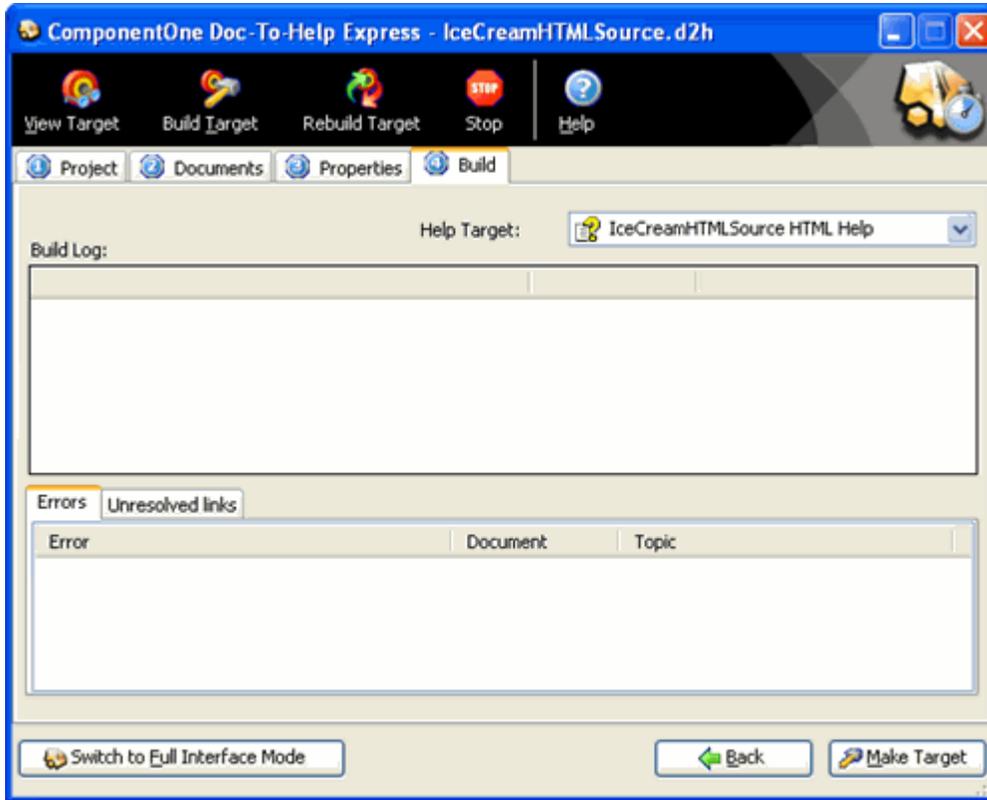
Properties Tab

The **Properties** tab allows modification of which type of help target will be created and setting the properties, such as caption, appearance and search. The **Localization** button provides additional control over what text strings appear in the help target. The **More Properties** button opens target specific options.



Build Tab

The **Build** tab builds the help target. On this tab, the help target can be viewed, rebuilt from scratch or built to incorporate updates to a document. Additionally, any errors and unresolved links that occurred during the build process will be listed on this tab.

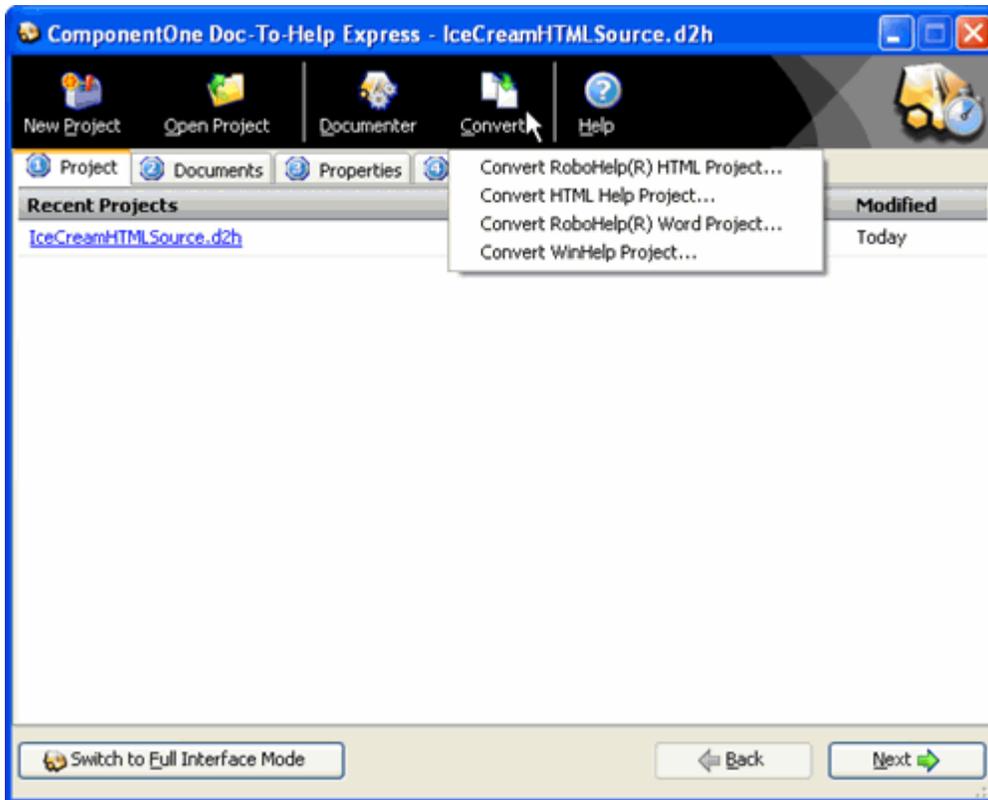


The Project Tab

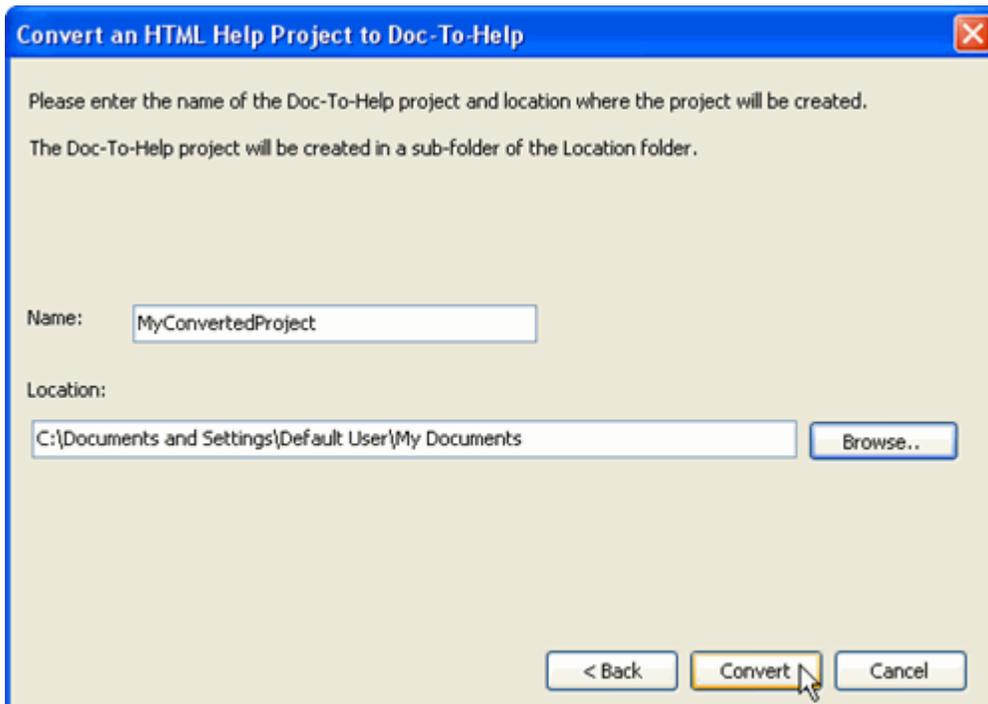
The **Project** tab appears when you start Doc-To-Help Express. On this tab, you will see all of the projects that you have recently worked on. Along the top of the window are buttons that allow you to open a project or create a new project. You can also open [Documenter for .NET](#) (page 449) or convert an existing RoboHelp, HTML Help, or WinHelp project to a Doc-To-Help project. To begin, simply click on the name of a recently modified project, open an existing project that is not already listed or create a new one.

Converting a RoboHelp, HTML Help or WinHelp Project

1. Click the **Convert** button and four options will appear: Convert RoboHelp Project, Convert HTML Help Project, Convert RoboHelp Word Project, and Convert WinHelp Project.



2. Choose the option for the file type that you want to convert. Browse to the location of the file for RoboHelp (.xpj, .mpj or .hpi), HTML Help (.hlp), or WinHelp (.hpi).
3. Next, enter the name of the project and the location of where the project will be created. Doc-To-Help Express will create the project in a subfolder of the location.



4. Click **Convert** and a window will appear showing the conversion process. When Doc-To-Help Express has finished converting, a dialog box will appear stating that the project has been successfully converted. You can now open the newly created project file in the **Project** tab by clicking the **Open** button. For more information on the conversion wizards see the Doc-To-Help documentation.

The Documents Tab

If you created a new project, DOC-TO-HELP Express automatically creates a new document and a glossary and places them in the document tree. If you are working with an existing project, the help target documents appear in the document tree in the same order they appear in the table of contents.

The document tree is designed to help organize the documents that will appear in the help target. Below the document tree are the **Document Organization** buttons, a series of arrow buttons that control the hierarchy of the table of contents in the help target. To organize the documents:

- Use the **Move Up** and **Move Down** buttons move the position of a document in the help target.
- Use the **Move Left** and **Move Right** buttons respectively create or remove document hierarchy.
- You can also move documents up and down in the document tree by dragging and dropping the document in the desired position.

Note: The hierarchy of the documents in the document tree does not affect the output in the help target. This is determined through the document type.

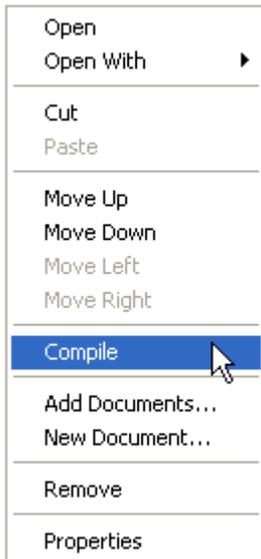
There are two types of documents in the document tree – [single and multiple topic documents](#) (page 183).

- Single Topic Documents inherit their hierarchy from the **Topic Style** property.
- Multiple Topic Documents inherit their hierarchy from topic header formatting.

Word documents are always considered multiple topic documents and their properties cannot be changed. Multiple topic documents can only have no Topic Style, which places them at the first level on the document tree. The Topic Style relates to the level of a single topic document in the document tree. It is helpful to keep both the position of the document in the document tree and the style level the same.

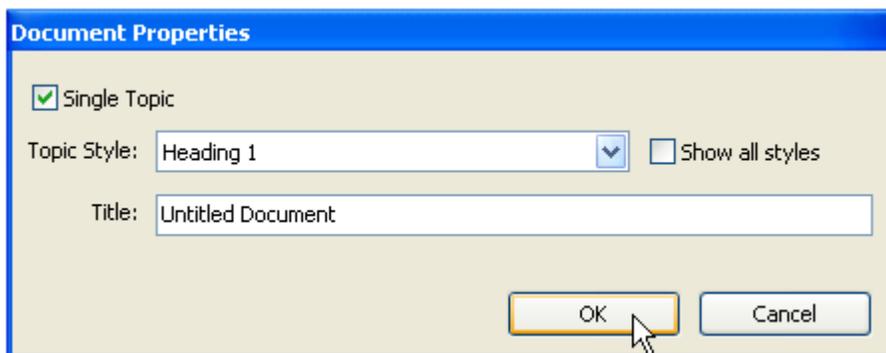
Since Word documents can only be multiple topic documents, Word documents cannot have children documents, nor can they be a child document either to another Word file or to an HTML document. Only single topic HTML documents can have children documents. There must be a single topic file within the document tree before a child document can be added. A child document can also have a child document attached to it.

To quickly move about the **Document** tab, simply right click on a document. This opens up a menu with the same commands as the buttons on the page, as well as the option to skip straight to the build process by clicking **Compile**.



The **Document Properties** button beneath the document tree allows you to modify the properties of HTML documents in the document tree. In the document tree, Single Topic documents inherit the title of the document. Selecting a single topic document requires that the title be specified in the **Title** box. The Topic Style relates to the document level in the document tree. You can choose a Topic Style that is not currently in the document tree by checking the **Show all styles** box.

Note: Selecting a style different from a level the document tree will result in a warning that the Style level does not correspond to the level of the document tree. It will ask you if you want to choose a different style level before proceeding. If you choose to continue without making changes, your help target will be built according to the Topic Style settings of each document.



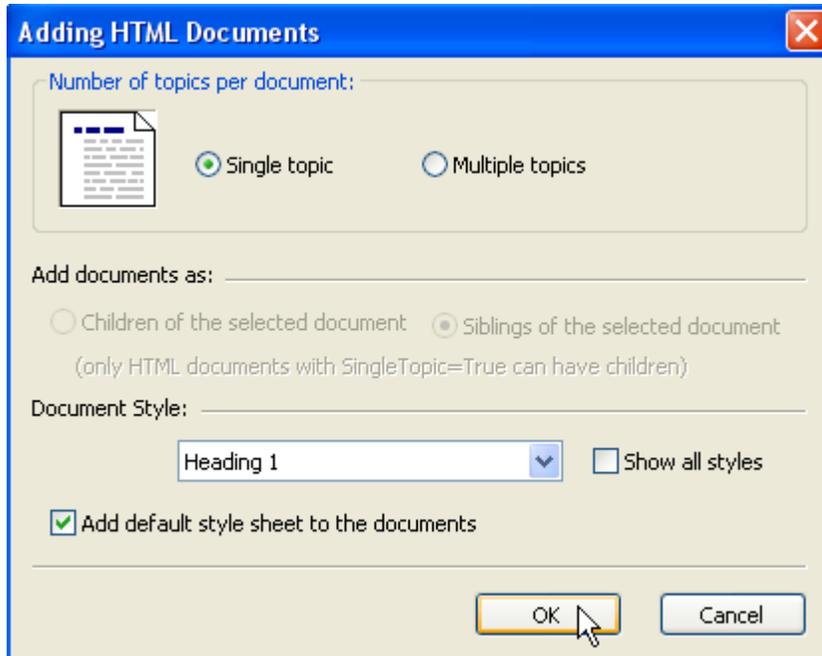
Adding Word Files

1. Select a file in the document tree that precedes the document that you want to add.
2. Click the **Add** button. The **Add Documents** dialog box opens. Make sure that the **Files of Type** drop-down is set to **Word Documents** or **Rich Text Documents**.
3. Browse to find the file that you want to add and select it.
4. Click **OK** and the file will be added after the file that you had selected in the document tree.

Adding HTML Files

1. Select a file in the document tree that precedes the documents you want to add.
2. Click the **Add** button. The **Add Documents** dialog box opens. Make sure that the **Files of Type** drop-down is set to **HTML Format**.
3. Browse to the file that you want to add.

- Another dialog box will appear where you can set the following options.



- Select the number of topics per document: single topic or multiple topics. HTML documents can contain a single topic or multiple topics. Single topic documents can have children documents, whereas multiple topic documents cannot.
- Select how to add the document to the document tree: as children or siblings of the selected document. Selecting children will place the current document as a subtopic of the selected document in the document tree. Selecting sibling will place the current document at the same level as the selected document.

Note: To add a child document to the document tree it must be attached to a single topic HTML file.

- Select the **Document Style**. The document style level corresponds to the level of the document in the document tree. Selecting the **Show all styles** checkbox will show all available styles, which may not correspond to the current document level in the document tree.

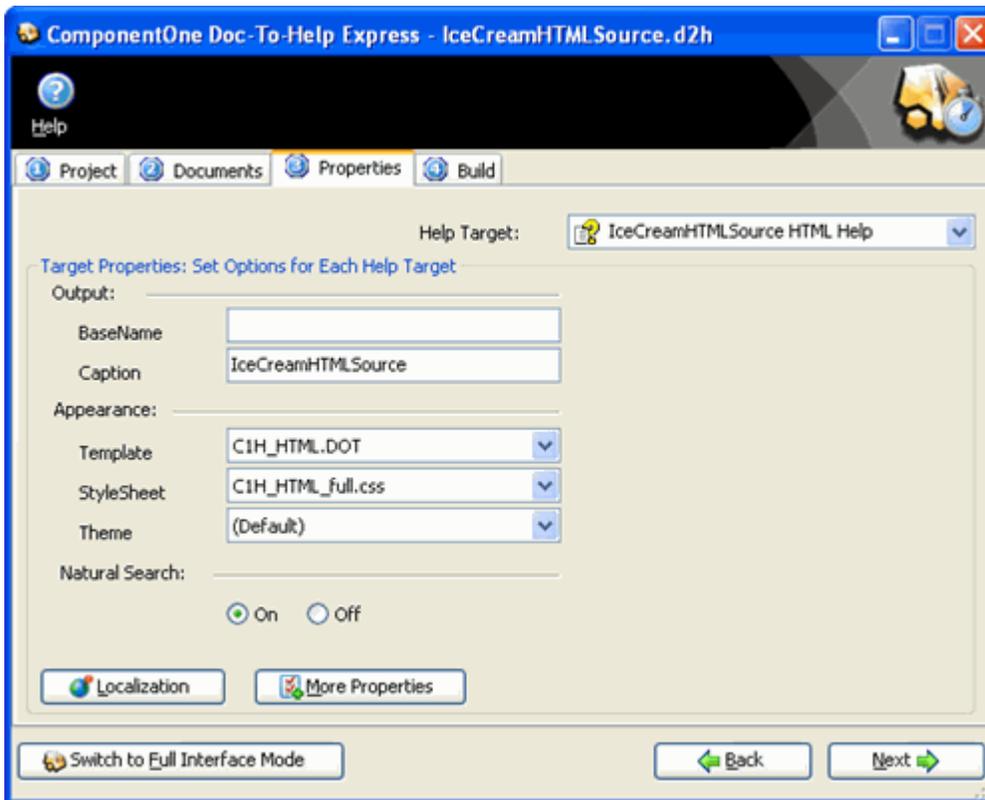
Note: Selecting a style that does not correspond to the document tree will result in a warning that the Style level does not correspond to the level of the document tree. It will ask you if you want to choose a different style level before proceeding. If you choose to continue without making changes, your help target will be built according to the Topic Style settings of each document.

- Check the box to add default style sheet to the document.
- Click **OK** and the file will be added after the file you had selected in the document tree.

Properties Tab

Help Target

When you created a new project, you specified what kind of help target you wanted to create. If you would like to change the type of help target, use the **Help Target** drop-down box to select your desired type of help target.

Target Properties*Output*

[BaseName](#) Base file name to be used for generating help target files.

[Caption](#) String to be displayed in the caption bar of the help window.

Appearance

[Template](#) Specifies the name of the document template applied to intermediate files during compilation.

[StyleSheet](#) Specifies the name of the style sheet applied to intermediate files during compilation.

[Theme](#) Determines the appearance of the Help pane and framesets for HTML-based targets.

Natural Search

On/Off Select **On** to generate Natural Search indexes for a Help target and add a custom tab (HTML Help) or button (WinHelp) to the Help file.

Localization

Clicking the **Localization** button opens the **Localization** window. This window allows you to modify the appearance of strings that appear in the help target. The options on this tab are specific to the type of target that you are creating. If you do not wish to modify the strings in the help target, you can continue on to the **Build** tab.

Help 2.0

Display	
ButtonLabel	More:
Manual	
OnPage	on page
Misc	
StringNext	Next
StringPrevious	Previous

- Display** [ButtonLabel](#) Specifies the text that precedes the subtopic buttons.
- Manual** [OnPage](#) Specifies the default text used when completing a cross-reference.
- Misc** [StringNext](#) Allows modification of the **Next** string value.
- [StringPrevious](#) Allows modification of the **Previous** string value.

HTML Help

Display	
ButtonLabel	More:
Manual	
OnPage	on page
Misc	
SearchCaption	N&atural Search
StringNext	Next
StringNoMatch	No matches found for your request.
StringPrevious	Previous

- Display** [ButtonLabel](#) Specifies the text that precedes the subtopic buttons.
- Manual** [OnPage](#) Specifies the default text used when completing a cross-reference.
- Misc** [SearchCaption](#) Specifies the text added to the tab caption when ComponentOne Natural Search is enabled.
- [StringNext](#) Allows modification of the **Next** string value.
- [StringNoMatch](#) Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.
- [StringPrevious](#) Allows modification of the **Previous** string value.

JavaHelp

[-] Display	
ButtonLabel	More:
[-] Manual	
OnPage	on page
[-] Misc	
StringContents	Contents
StringFavorites	Favorites
StringFoundMany	%d topics found:
StringFoundOne	1 topic found:
StringFoundZero	No topics found.
StringIndex	Index
StringNext	Next
StringPrevious	Previous
StringReturnToIndex	Return to Index
StringSearch	Search

Display	ButtonLabel	Specifies the text that precedes the subtopic buttons.
Manual	OnPage	Specifies the default text used when completing a cross-reference.
Misc	StringContents	Allows modification of the Contents string value.
	StringFavorites	Allows modification of the Favorites string value.
	StringFoundMany	Allows modification of the %d topics found: string value.
	StringFoundOne	Allows modification of the 1 topic found: string value.
	StringFoundZero	Allows modification of the No topics found. string value.
	StringIndex	Allows modification of the Index string value.
	StringNext	Allows modification of the Next string value.
	StringPrevious	Allows modification of the Previous string value.
	StringReturnToIndex	Allows modification of the Return to Index string value.
	StringSearch	Allows modification of the Search string value.

Manual

[-] Manual	
OnPage	on page
[-] Misc	
StringContents	Contents
StringIndex	Index

Manual	OnPage	Specifies the default text when completing a cross-reference.
Misc	StringContents	Allows modification of the Contents string value.
	StringIndex	Allows modification of the Index string value.

NetHelp

[-] Display	
ButtonLabel	More:
[-] Manual	
OnPage	on page
[-] Misc	
StringContents	Contents
StringFoundMany	%d topics found:
StringFoundOne	1 topic found:
StringFoundZero	No topics found.
StringIndex	Index
StringIndexCaption	Look for
StringNext	Next
StringNoMatch	No matches found for your request.
StringPrevious	Previous
StringReturnToIndex	Return to Index
StringSearch	Search
StringSearchCaption	Search for
StringSearchGo	Go
StringSyncToc	sync toc

Display	ButtonLabel	Specifies the text that precedes the subtopic buttons.
Manual	OnPage	Specifies the default text used when completing a cross-reference.
Misc	StringContents	Allows modification of the Contents string value.
	StringFoundMany	Allows modification of the %d topics found: string value.
	StringFoundOne	Allows modification of the 1 topic found: string value.
	StringFoundZero	Allows modification of the No topics found. string value.
	StringIndex	Allows modification of the Index string value.
	StringIndexCaption	Represents the string to be displayed in the Index window above the index text box.
	StringNext	Allows modification of the Next string value.
	StringNoMatch	Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.
	StringPrevious	Allows modification of the Previous string value.
	StringReturnToIndex	Allows modification of the Return to Index string value.
	StringSearch	Allows modification of the Search string value.
	StringSearchCaption	Represents the string to be displayed in the Search window above the search text box.
	StringSearchGo	Represents the string to be displayed in the search button within the Search window.
StringSyncToc	Allows modification of the Sync TOC string value.	

WinHelp

[-] Display	
ButtonLabel	More:
[-] Manual	
OnPage	on page
[-] Misc	
SearchCaption	N&atural Search
StringNoMatch	No matches found for your request.
Title	

Display	ButtonLabel	Specifies the text that precedes the subtopic buttons.
Manual	OnPage	Specifies the default text used when completing a cross-reference.
Misc	SearchCaption	Specifies the text added to the button caption when ComponentOne Natural Search is enabled.
	StringNoMatch	Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.
	Title	Represents the string to be displayed in the caption bar of the help window.

More Properties

The **More Properties** button opens the **Properties** window for the help target. This window allows you to customize the appearance of the help window. Like the **Localization** window, the options on this window are specific to the type of target that you are creating.

Help 2.0

There are no additional properties for Help 2.0

HTML Help

[-] Buttons	
ShowBackButton	True
ShowForwardButton	False
ShowHideShowButton	True
ShowNextButton	False
ShowPreviousButton	False
ShowPrintButton	True
[-] Tabs	
FavoritesTab	False
SearchTab	True
SearchTabAdvanced	True

Buttons	ShowBackButton	When selected, the Back button is included in the help window.
	ShowForwardButton	When selected, the Forward button is included in the help window.
	ShowHideShowButton	When selected, the Hide and Show buttons are included in the help window.

	ShowNextButton	When selected, the Next button is included in the help window. Note: The BinaryTOC property must be set to True for the navigation buttons to function.
	ShowPreviousButton	When selected, the Previous button is included in the help window. Note: The BinaryTOC property must be set to True for the navigation buttons to function.
	ShowPrintButton	When selected, the Print button is included in the help window.
Tabs	FavoritesTab	When enabled, the Favorites tab is added to the help window.
	SearchTab	When enabled, the Search tab is added to the help window.
	SearchTabAdvanced	Adds additional functionality to the search tab.

JavaHelp

Tabs	
ContentsTab	True
FavoritesTab	False
IndexTab	True
SearchTab	True

Tabs	ContentsTab	When enabled, the Contents tab is added to the help window.
	FavoritesTab	When enabled, the Favorites tab is added to the help window.
	IndexTab	When enabled, the Index tab is added to the help window.
	SearchTab	When enabled, the Search tab is added to the help window.

Manual

Title Page	
ByLine	By ComponentOne LLC
SuperTitle	Doc-To-Help
Title	Standard Template

Title Page	ByLine	Specifies the byline text used on the title page of printed manuals.
	SuperTitle	Specifies the text that appears above the title string on the title page of printed manuals.
	Title	Represents the title string of an object. Note: This property sets the text that appears below the SuperTitle string on the title page.

NetHelp

HTML	
DefaultFile	default.htm
Tabs	
ContentsTab	True
IndexTab	True
SearchTab	True

HTML	DefaultFile	File name to be used as part of the base URL for the HTML target. (Usually default.htm or index.html)
Tabs	ContentsTab	When enabled, the Contents tab is added to the help window.
	IndexTab	When enabled, the Index tab is added to the help window.
	SearchTab	When enabled, the Search tab is added to the help window.

WinHelp

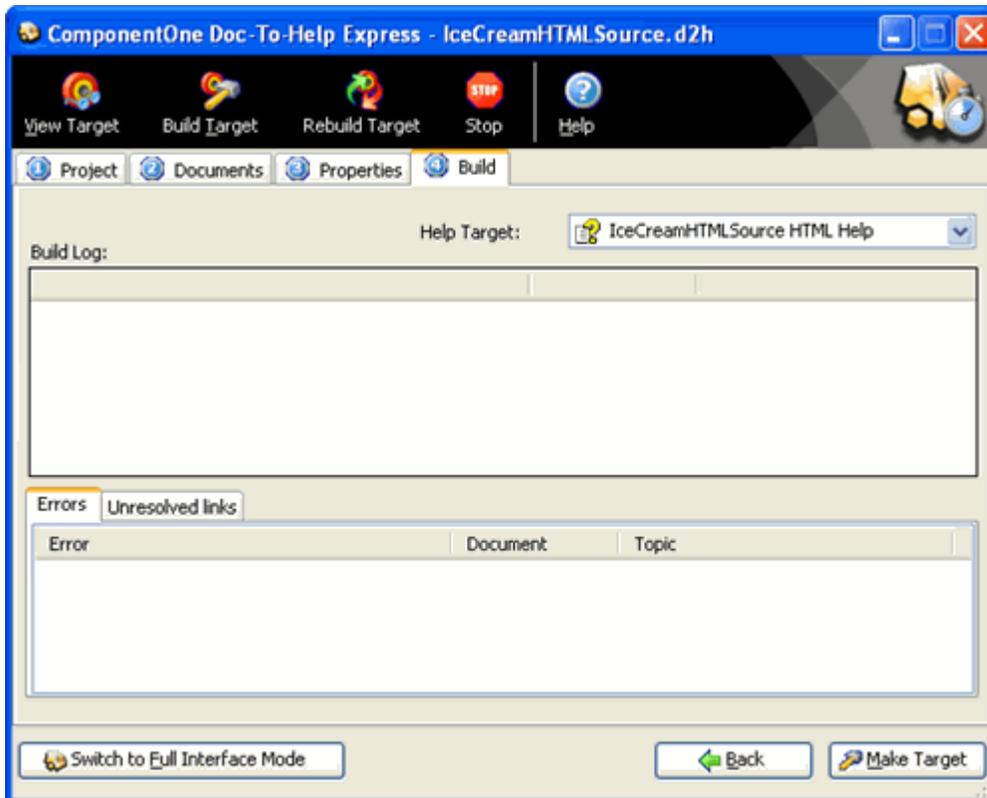
Buttons	
ShowBackButton	True
ShowBrowseButtons	True
ShowContentsButton	False
ShowFindButton	False
ShowHelpTopicsButton	True
ShowIndexButton	False
ShowOptionsButton	True
ShowPrintButton	True

Buttons	ShowBackButton	When selected, the Back button is included in the help window.
	ShowBrowseButtons	When selected, the Browse buttons are included in the help window.
	ShowContentsButton	When selected, the Contents button is included in the help window.
	ShowFindButton	When selected, the Find button is included in the help window.
	ShowHelpTopicsButton	When selected, the Help Topics button is included in the help window.
	ShowIndexButton	When selected, the Index button is included in the help window.
	ShowOptionsButton	When selected, the Options button is included in the help window.
	ShowPrintButton	When selected, the Print button is included in the help window.

Build Tab

The **Build** tab is the final step in creating your help target. The **Help Target** drop-down box at the top right allows you to change the type of help target you are about to create.

Note: If you choose to change the type of help target, you may want to return to the **Properties** tab and confirm the settings.



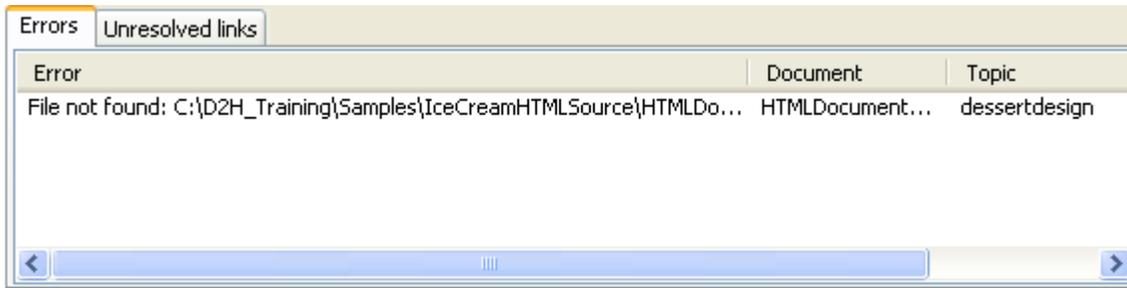
If you have used the **Next** button to move from step to step, you will notice clicking **Make Target** is your final step. You can also choose to make your target by clicking the **Build Target** button at the top of the window after **View Target** button. Once your target is built, you can view it by clicking the **View Target** button.

If you have made changes within your documents since the last time you built your help file, you can click **Make Target** or **Build Target** to include these changes into your help target. Clicking **Rebuild Target** gives you the option to rebuild your help target from scratch.

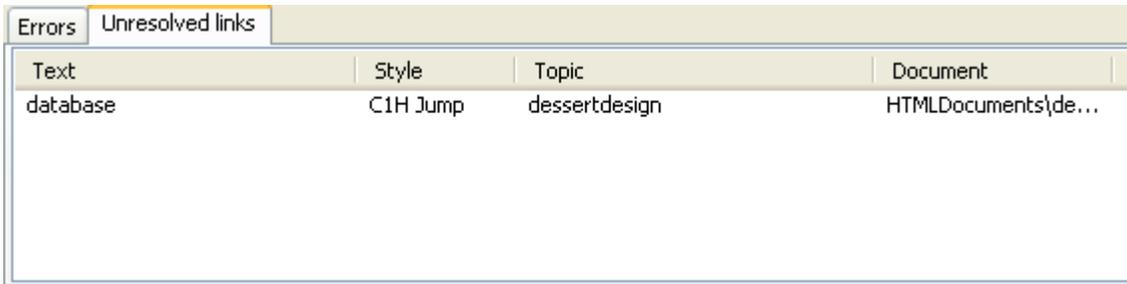
The **Build Log** displays the building process. It will indicate if there were any errors or unresolved links found during the process. No errors or unresolved links in the build log means that you have built your help target successfully and can now view it by clicking the **View Target** button.

Reading project data...	Done
Building...	
Contents file...	
Index file...	
Help target...	
Saving changes...	Done
0 error(s)	
0 unresolved link(s)	

Errors and unresolved links are displayed in the section below the **Build Log**. If an error occurs during the build process it is displayed, along with the document and topic where the error is located. Double-clicking the error opens the document.



The **Unresolved Links** tab displays any broken links in the help target. It specifies the text of the link, the link style and the topic and document where the broken link is located. Double-clicking the error opens the document.



Team Authoring

Doc-To-Help team authoring allows Help authors to work together on a single project; this includes editing, adding and removing documents, and performing all functions that can be done with a regular, single-user Doc-To-Help project. Project changes are available to the entire team, and one author's changes will not be overwritten by another author's changes. This feature is available in Doc-To-Help Enterprise 2006 only.

Team authoring is a basic source control feature in which authors work in their own local copy of a project on their machine, called the **working copy**, while the master project, or the **team project**, is located on the organization's network or on a Web server.

To set up a team-authoring project, you start with a regular, single-user Doc-To-Help project and make it available to other team members by uploading it to a central repository. Once the project is uploaded, it becomes the **team project**, and the project you started with becomes your **working copy** of the team project. Next, you tell the members of your team where the **team project** is located. Then they can connect to it and create working copies on their own machines.

Each author works on his or her own working copy. Changes made by an author are not automatically propagated to the team project in the central repository. Until the changes are checked in, they remain local to the author's machine, appearing only in his or her working copy of the project. Likewise, changes made by other authors cannot be seen until they are retrieved from the repository. Doc-To-Help provides special commands to send, or check in, your changes to the repository and to get other authors' changes from the repository.

Team Authoring Restrictions

A Doc-To-Help project must satisfy a few requirements in order to be used for team authoring:

- All documents and auxiliary files, such as graphics, must be located within subfolders inside of the project folder. Moreover, they must be located inside the folders designated for storing source documents, which are specified in the **WordSourceFolder** and **HTMLSourceFolder**; the default folder names are WordDocuments and HTMLDocuments. This folder structure for files is a requirement in team authoring, but it is also recommended practice for all Doc-To-Help projects.
- Templates, style sheets and themes are outside of team-authoring control. If your project uses customized templates, style sheets or themes, you must make sure that all team members have the same versions of those files in the same locations on each of their machines.

- All team members must have exactly the same version, or build number, of Doc-To-Help installed on their machine. Version incompatibility will be detected as an error.

Setting Up a Team Project

The first step in working on a project in a team environment is to make the project available to other users by uploading it to a central repository.

Note: Before setting up a Team Project, make sure your project satisfies all restrictions described in the **Team Authoring Restrictions** topic.

Doc-To-Help supports two repository types:

- File System Repository

File system repository is a location accessible from your file system. Usually, it is on a computer belonging to your organization's network. It is accessed through the regular operating system's file system, as you access any other files. No Web server is involved.

- Web Repository

Web repository is a location on the Internet, on a Web server. It can be accessed from anywhere on the Internet, including from behind a firewall. Doc-To-Help uses Web-based Distributed Authoring and Versioning (WebDAV) technology. WebDAV is a protocol that allows connectivity between a remote server and a local machine for easy file sharing among multiple users. WebDAV is supported by all major Web servers, but it is not necessarily enabled for any server location. It requires certain privileges, so you may need to consult your system administrator to find out the locations that you can use for a Web repository.

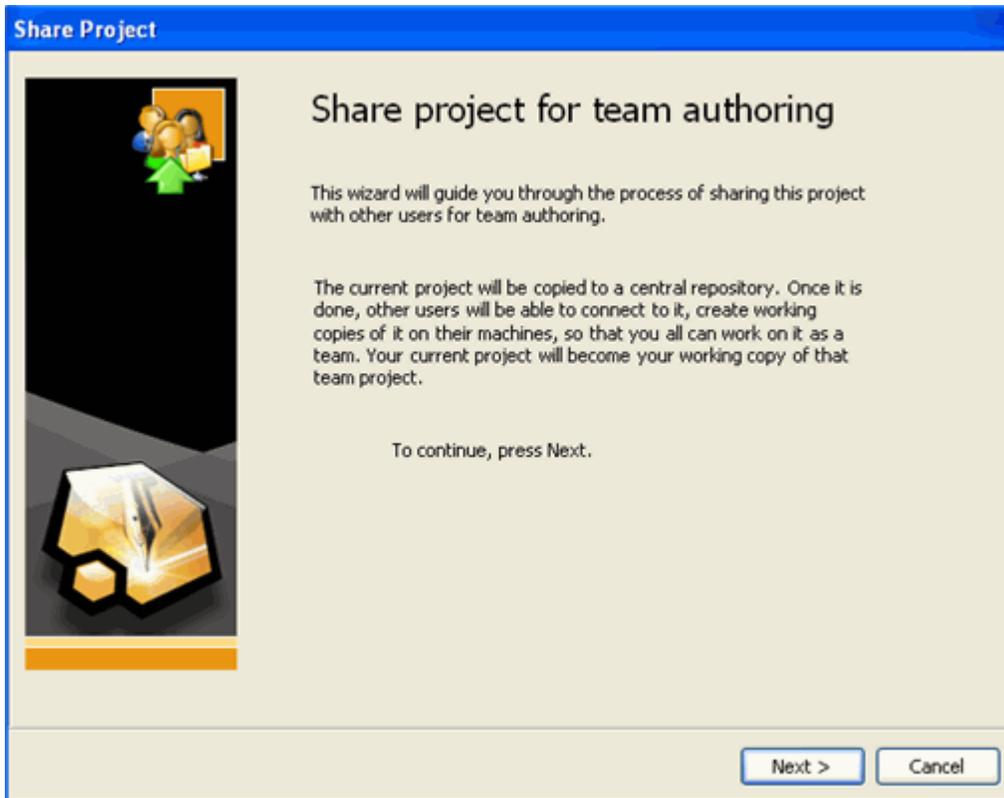
Sharing a Project with a File System Repository

Sharing a project with a file system repository is the easiest method for team authoring. It is recommended if all team members are connected to the same network.

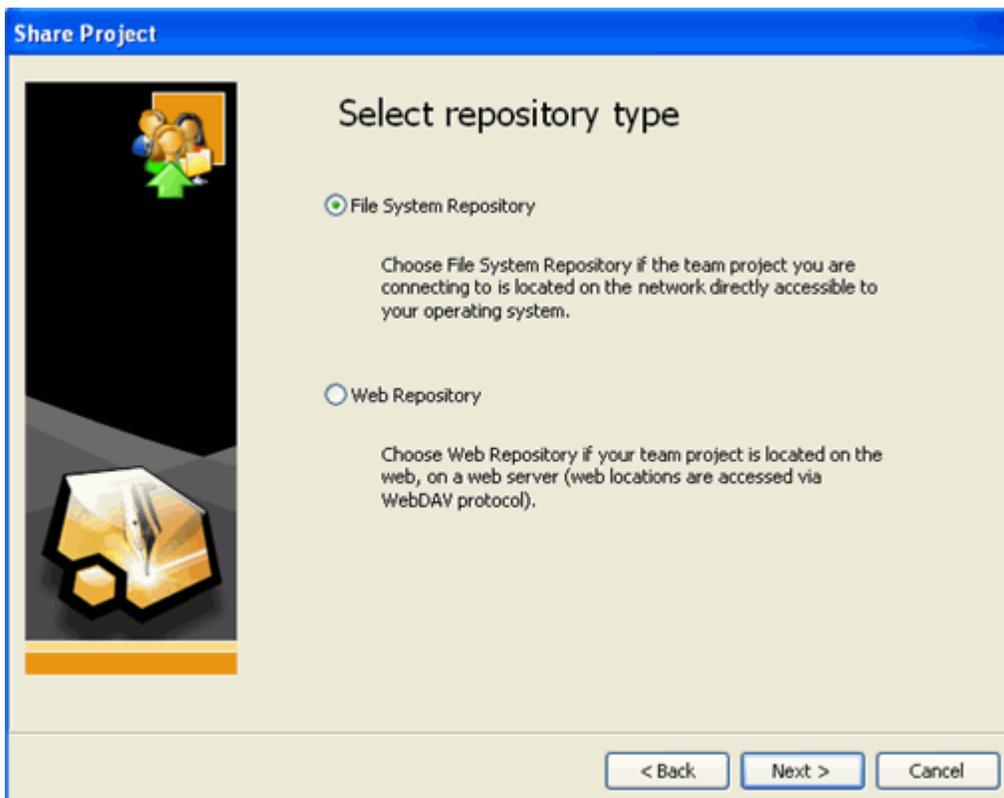
The only prerequisite for sharing a project with a file system repository is having access to a shared folder on the network.

To share a project for team authoring:

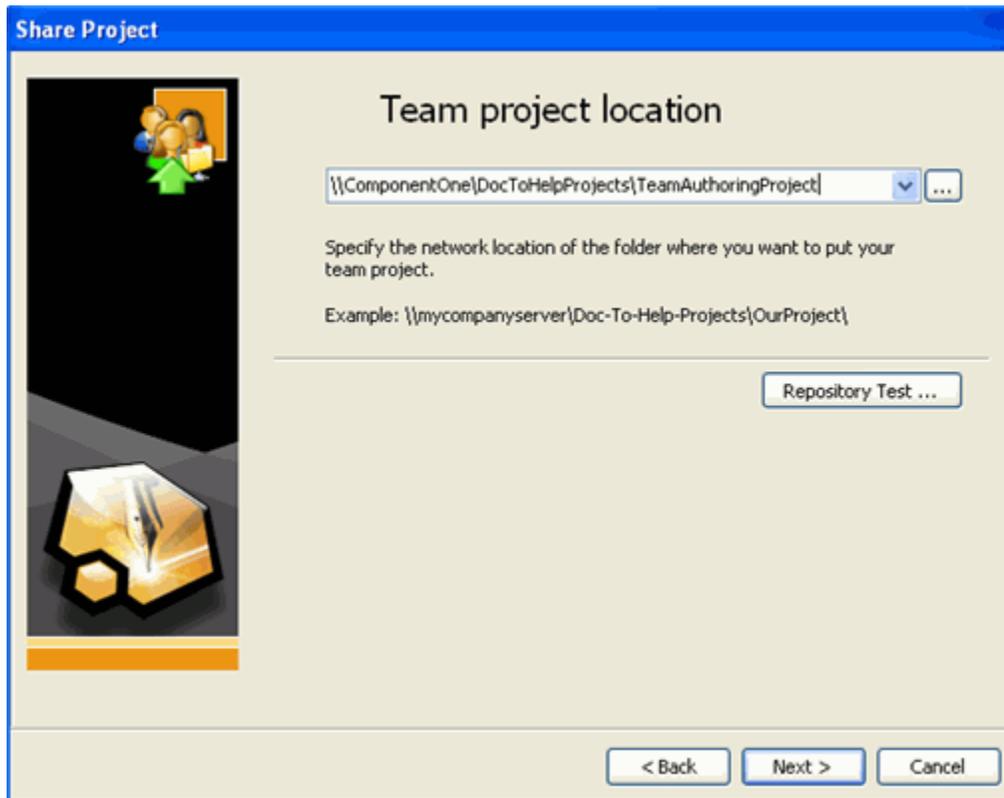
1. Create a new Doc-To-Help project or open an existing one.
2. Select **File | Share Project for Team Authoring**. The **Share Project** wizard appears.



3. Click **Next** to continue.
4. Under **Select repository type**, choose **File System Repository**. The team project will be located on your organization's network.



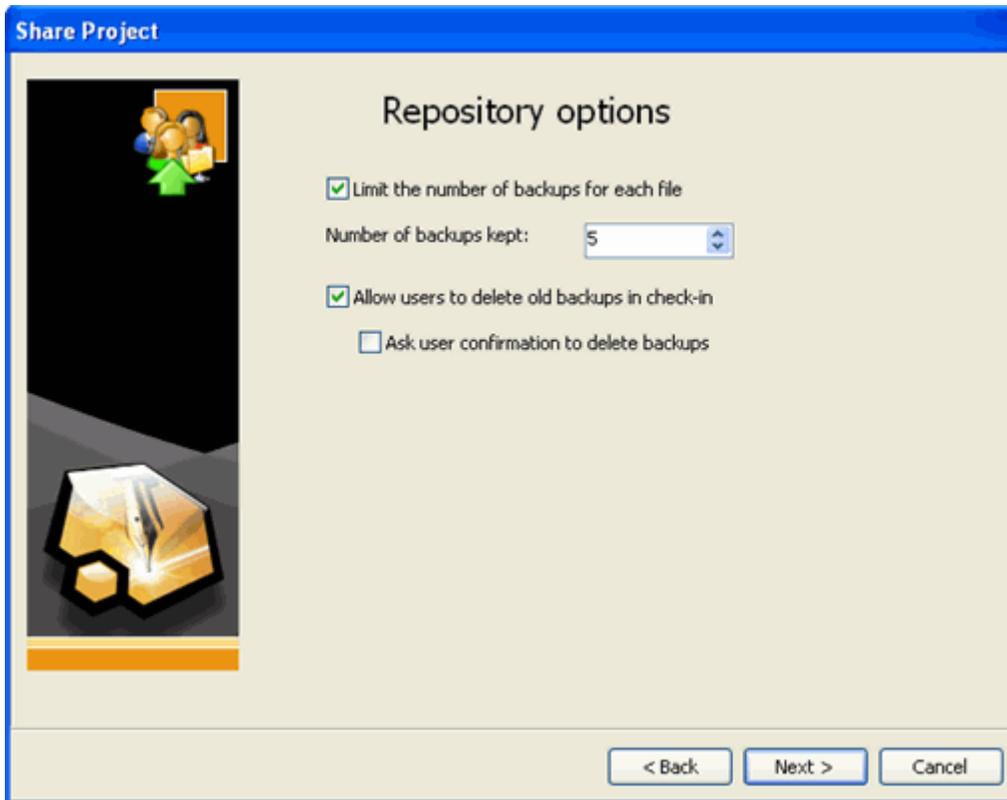
5. Click **Next** to continue.
6. Enter a network location for the team project or click the **ellipsis** button to browse for a folder on the network where you would like the team project to live.



7. Click the **Repository Test** button to test the connection to the repository and the files and subfolders.
8. Click **Next** to continue.
9. In the **Repository Options** window, you can specify the settings regarding file backups:
 - **Limit the number of backups for each file:** When checked, there is a limit on the number of backups kept for each file in the repository. This number can be specified in the **Number of backups kept** box; the default is **5**. If this option is unchecked, every time a file is checked-in, a new backup file is created in the repository, and there is no automatic cleanup of old backup files.
 - **Allow users to delete old backups in check-in:** Check this box if you want other team members to be able to delete old backup files when they check a newer version of the file into the repository when the limit of backups for that file has been exceeded.
 - **Ask user confirmation to delete backups:** Check this box if you want a confirmation dialog box to appear when the limit of backups for a file has been exceeded when a team member checks a newer version of the file into the repository. This option is only available when the **Allow users to delete old backups in check-in** checkbox has been checked.

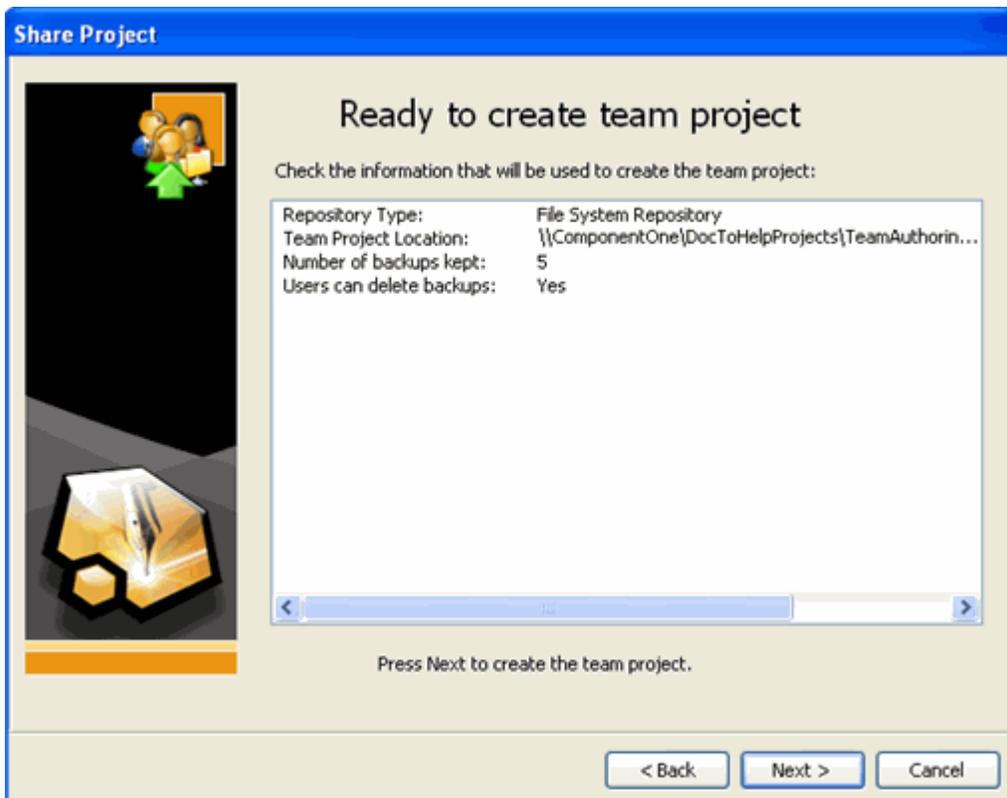
The **Ask user confirmation to delete backups** checkbox is unchecked by default, meaning the team member will not be asked to confirm the backup cleanup; it is performed automatically without any notification.

Note: These options can be changed using the team administration utility at any time after the team project is created. See [Changing Repository Settings](#) (page 550) for more information.



10. Click **Next** to continue.

11. Confirm your repository information and click **Next** to create the team project.



The **Upload project** dialog box appears.

12. Once the upload process is complete, click **Close**. At this point, your project has been uploaded to shared directory and the copy in which you were working is the **working copy**. This is the version in which you will perform all authoring tasks. The **team project** now on the network is the master copy.

Note: The **team project** should never be manually opened or altered. If you wish to make changes in the team project, make the changes locally in your working project, and then send, or check in, your changes to the central repository.

13. Click **Finish**.

Sharing a Project with a Web Repository

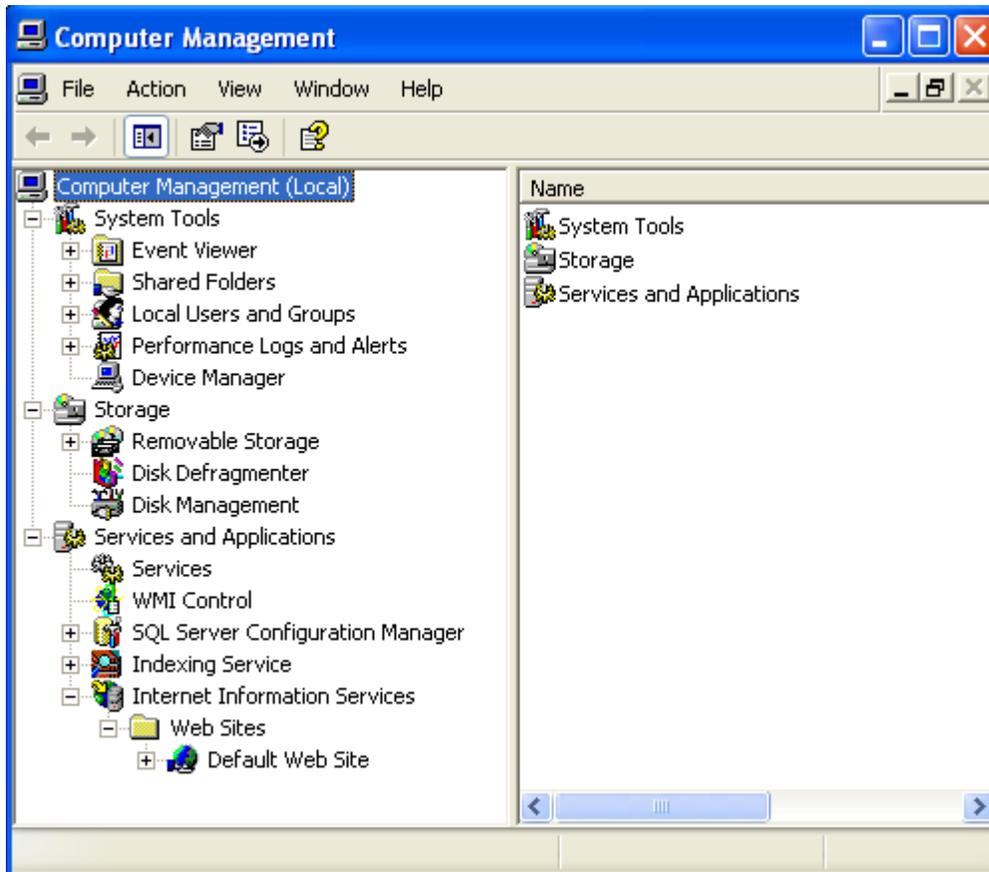
Sharing a project with a Web repository is the most advanced method for sharing a **team project**. It allows you to share your project with anyone connected to the Web.

When using a Web repository, Doc-To-Help uses Web-based Distributed Authoring and Versioning (WebDAV) technology. WebDAV is a protocol that allows connectivity between a remote server and a local machine for easy file sharing among multiple users. WebDAV is supported by all major Web servers, but it is not necessarily enabled for any server location.

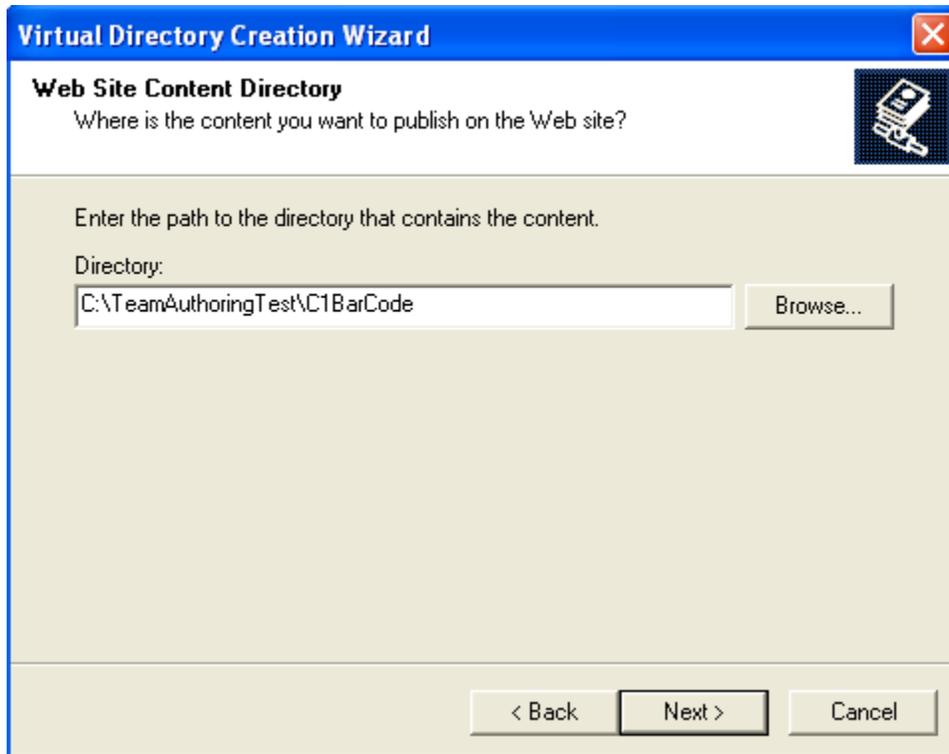
For example, Internet Information Services (IIS) is just one of the Web servers that you can use. If you are using IIS, first you must set up a virtual directory on the Web server in order to share a project using a Web repository. This virtual directory will be the location of the team project.

To set up a virtual directory on your Windows machine if you are using IIS:

1. Make sure IIS is installed.
2. Open the **Computer Management** utility:
 - a. Click the **Start** menu and select **Control Panel**.
 - b. Click **Performance and Maintenance** and then click **Administrative Tools**.
 - c. Double-click **Computer Management**.
3. Expand the **Services and Applications** node.
4. Expand the Internet Information Services and Web Sites nodes.



5. Right-click **Default Web Site** and select **New | Virtual Directory**. The **Virtual Directory Creation Wizard** opens.
6. Click **Next**.
7. Enter a name for your virtual directory in the **Alias** text box. This name will be part of the URL for your Web repository.
8. Browse to the physical directory on your machine that contains the content you want to share and click **Next**.



9. For the **Access Permissions**, make sure these items are checked: **Read**, **Run scripts**, **Write**, and **Browse**.
10. Click **Next** and then click **Finish** to complete the process. The virtual directory is now set up.

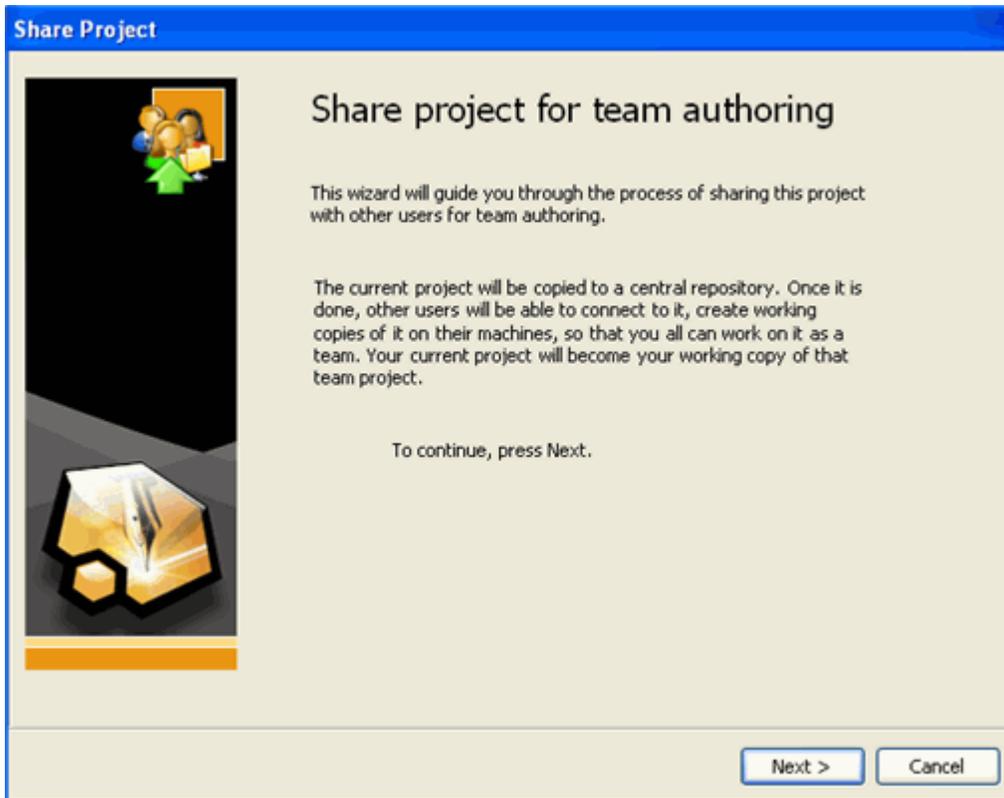
Here are some tips on setting up a virtual directory:

- Make sure the directory has Read, Write, and Browse privileges.
- You may need to consult your network administrator to set up the virtual directory.

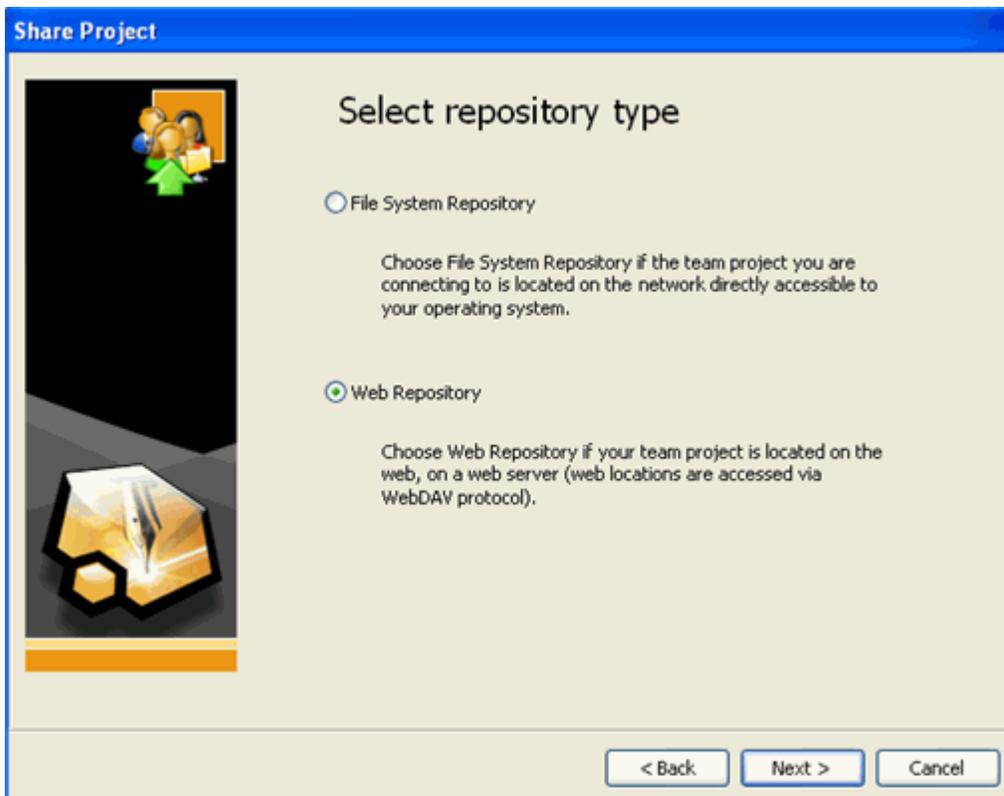
Once you have a virtual directory created, then you can share the project.

To share a project for team-authoring:

1. Create a new Doc-To-Help project or open an existing one.
2. Select **File | Share Project for Team Authoring**. The **Share Project** wizard appears.



3. Click **Next** to continue.
4. Under **Select repository type**, choose **Web Repository**. The team project will be located on a Web server.



5. Click **Next** to continue.
6. Enter the URL of the virtual directory you created.

Share Project

Team project location

Specify the web address (URL) of the folder on your web server where you want to put the team project.

Example: `http://www.yourcompany.com/Projects/YourProject/`

Authentication information for communication with your web server:

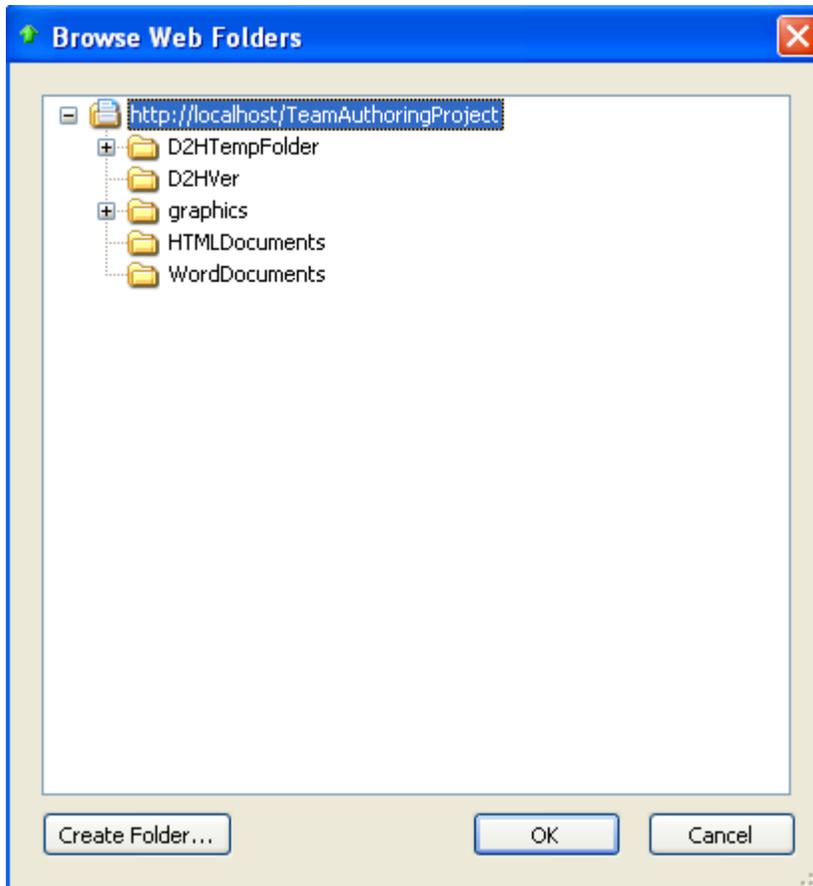
Use Windows Authentication

Domain:

User:

Password:

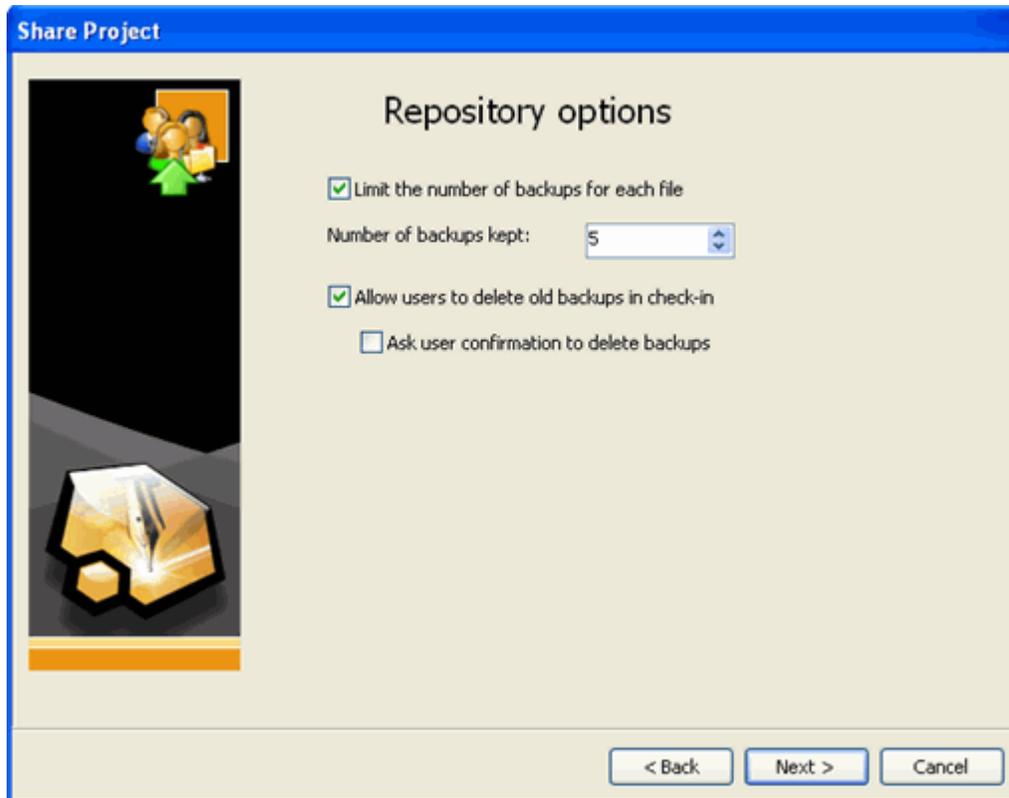
Notice that once you enter the URL, you can click the **ellipsis** button and browse the directory, as well as create a new folder, if necessary.



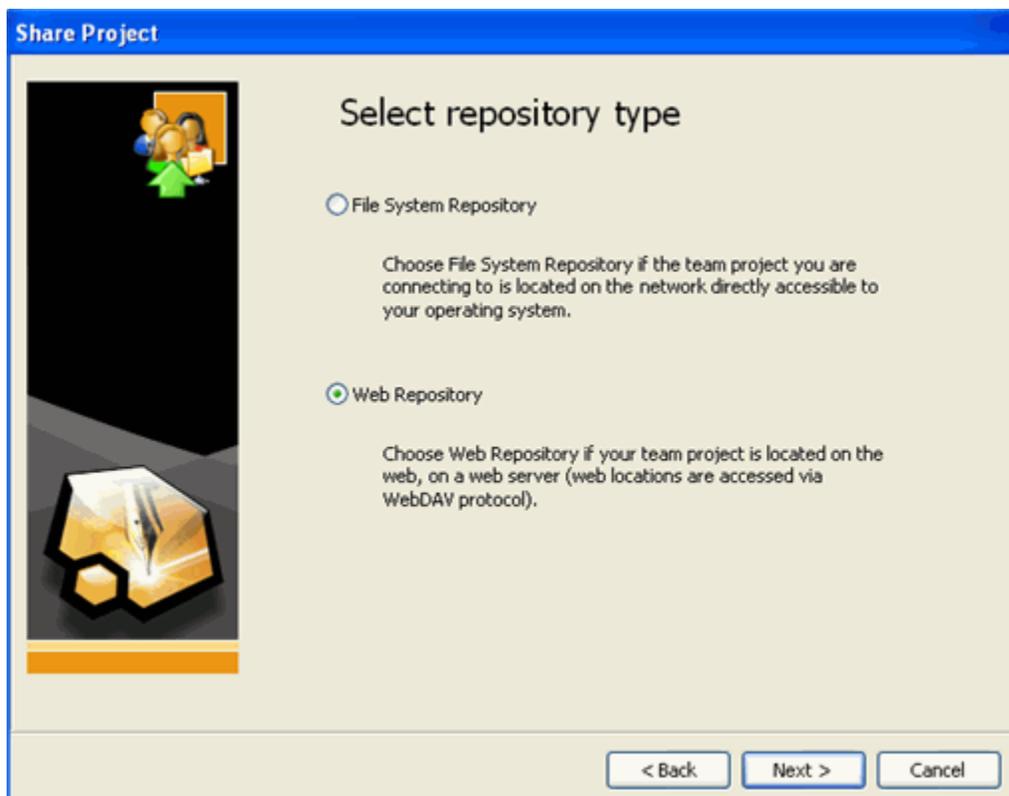
7. Choose your authentication method. If you are using Windows Authentication, leave **Use Windows Authentication** checked. If you are using another form of authentication, deselect the check box and enter your login credentials.
8. Click the **Repository Test** button to test the connection to the repository and the files and subfolders.
9. Click **Next** to continue.
10. In the **Repository Options** window, the two checkboxes regarding file backups are checked by default:
 - **Limit the number of backups for each file:** When checked, there is a limit on the number of backups kept for each file in the repository. This number can be specified in the **Number of backups kept** box; the default is **5**. If this option is unchecked, every time a file is checked-in, a new backup file is created in the repository, and there is no automatic cleanup of old backup files.
 - **Allow users to delete old backups in check-in:** Check this box if you want other team members to be able to delete old backup files when they check a newer version of the file into the repository when the limit of backups for that file has been exceeded.
 - **Ask user confirmation to delete backups:** Check this box if you want a confirmation dialog box to appear when the limit of backups for a file has been exceeded when a team member checks a newer version of the file into the repository. This option is only available when the **Allow users to delete old backups in check-in** checkbox has been checked.

The **Ask user confirmation to delete backups** checkbox is unchecked by default, meaning the team member will not be asked to confirm the backup cleanup; it is performed automatically without any notification.

Note: These options can be changed using the team administration utility at any time after the team project is created. See [Changing Repository Settings](#) (page 550) for more information.



11. Click **Next** to continue.
12. Click **Yes** to confirm your repository information and click **Next** to create the team project.



The **Upload project** dialog box appears.

- Once the upload process is complete, click **Close**. At this point, your project has been uploaded to shared directory and the copy in which you were working is the **working copy**. This is the version in which you will perform all authoring tasks. The **team project** now on the Web server is the master copy.

Note: The **team project** should never be manually opened or altered. If you wish to make changes in the team project, make the changes locally in your working project, and then send, or check in, your changes to the central repository.

- Click **Finish**.

Connecting to a Team Project

Team members must connect to and download a team project in order to create their own working copies of the project. Once they have a working copy, they can open and edit it each time they want to contribute to the project.

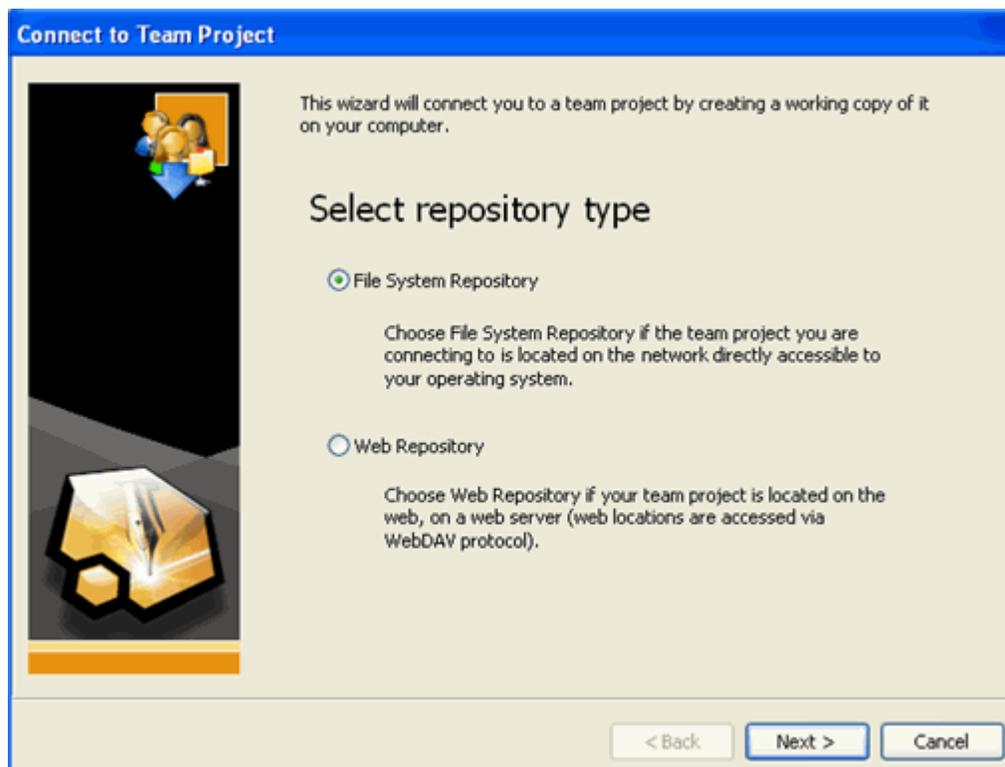
You can connect to a project in either a file system repository or in a Web repository. The project leader, or the person who starts the team project, should give team members the network path for the file system repository or the URL for the Web repository where the team project is located.

Connecting to a Project in a File System Repository

If you or another team member has shared a project with a file system repository, use the File System connection to download a working copy.

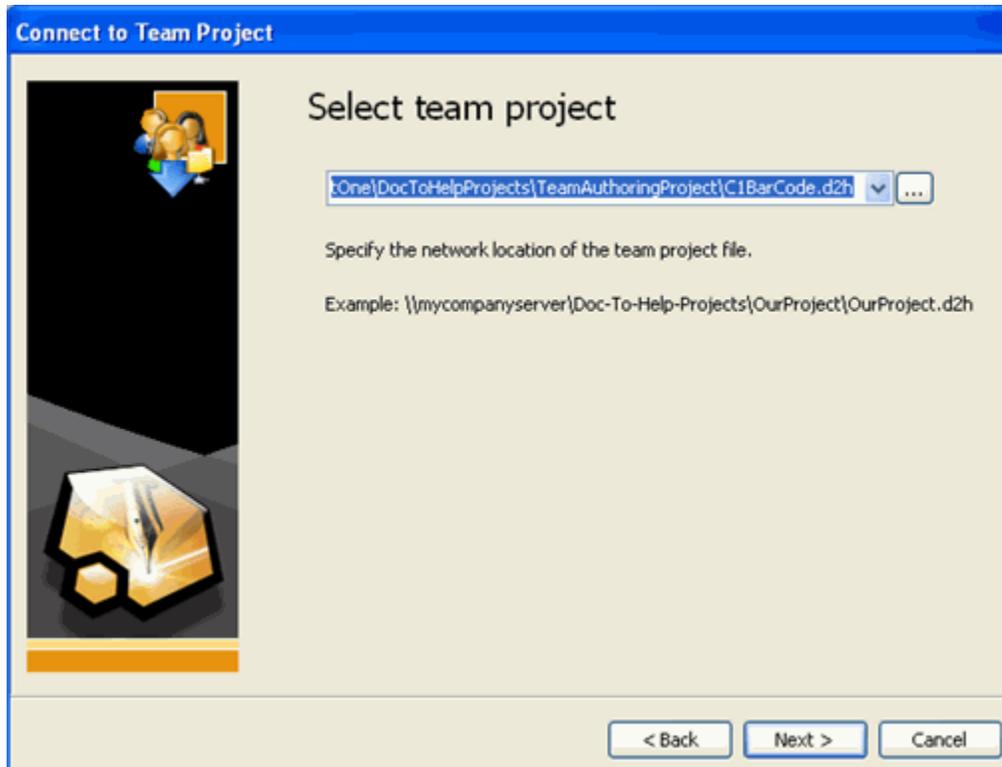
To connect to and download a team project from a file system repository:

- Open Doc-To-Help, click the **File** menu and select **Connect to Team Project**.
- Under **Select repository type**, choose **File System Repository**. The team project is located on your organization's network.

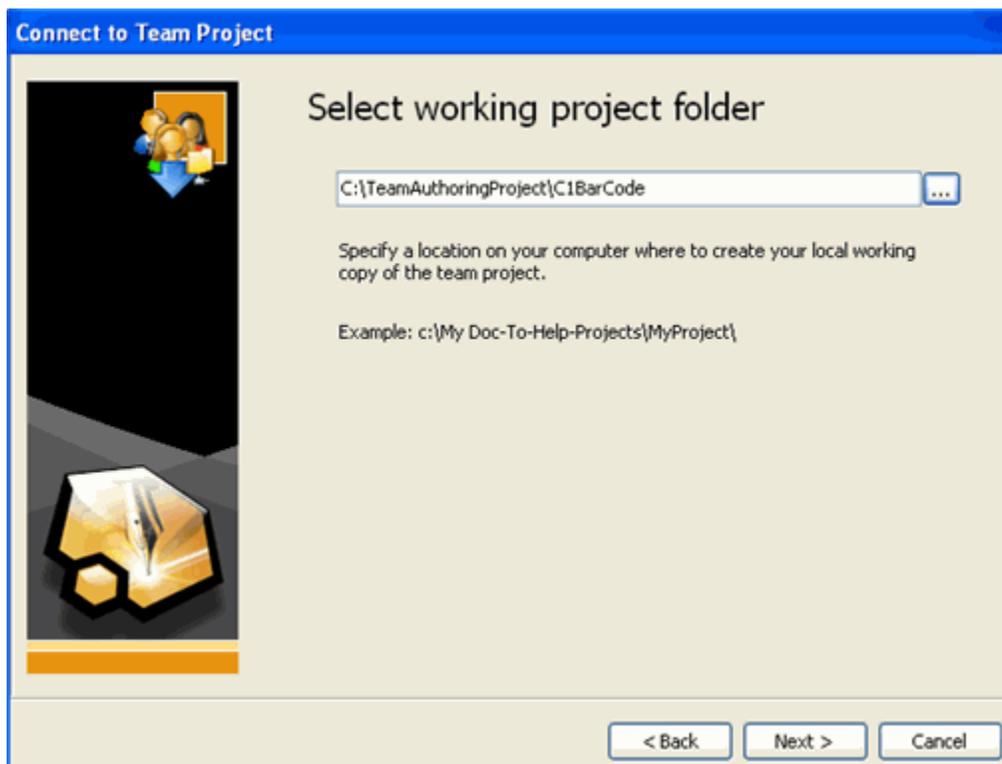


- Click **Next** to continue.

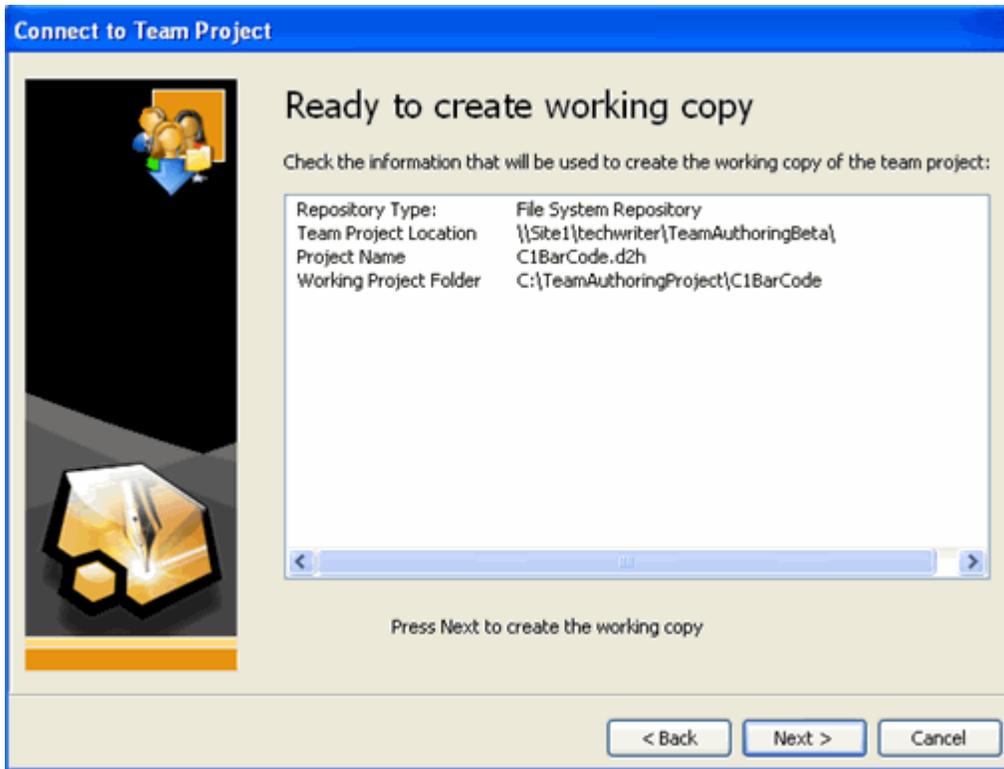
4. Enter the network location of the team project or click the **ellipsis** button to browse for the project.



5. Click **Next** to continue.
6. Enter a location for your working copy or click the **ellipsis** button to browse for a location. This is the project you will edit.



- Click **Next** and confirm the working copy information.



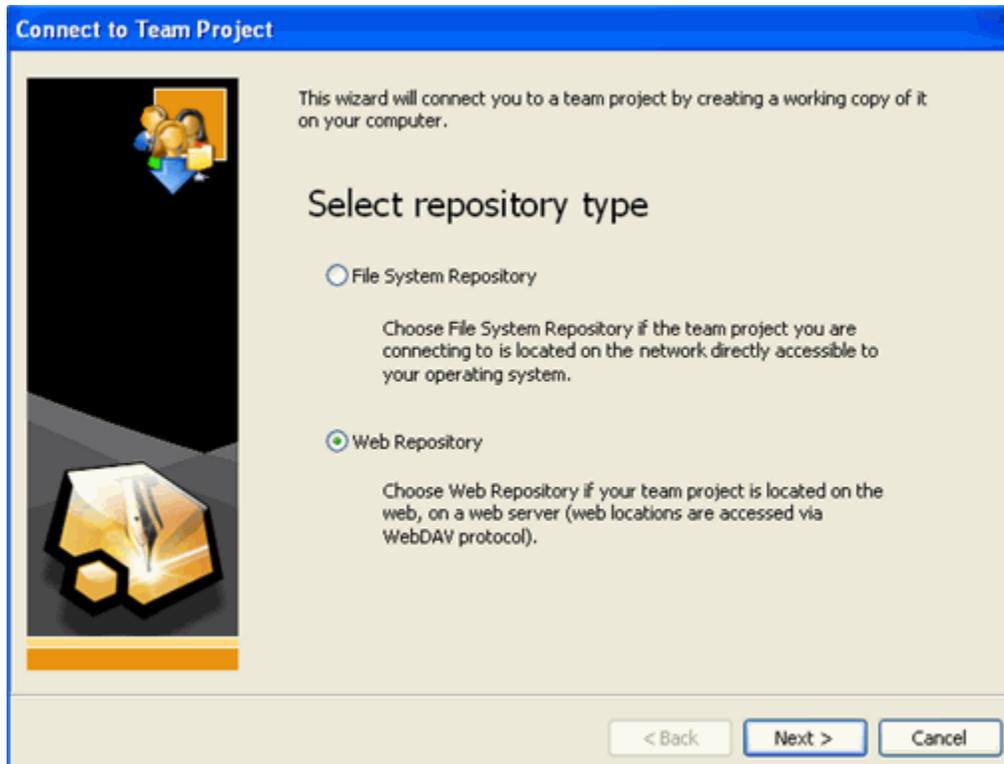
- Click **Next** to create the working copy. Once the files have been downloaded, you will have a local copy of the project in which to work. Simply open this working copy as you would any other Doc-To-Help project. You do not need to connect to and download the team project every time you open Doc-To-Help.
- Click **Close** and then click **Finish**. Your working copy of the project opens.

Connecting to a Project in a Web Repository

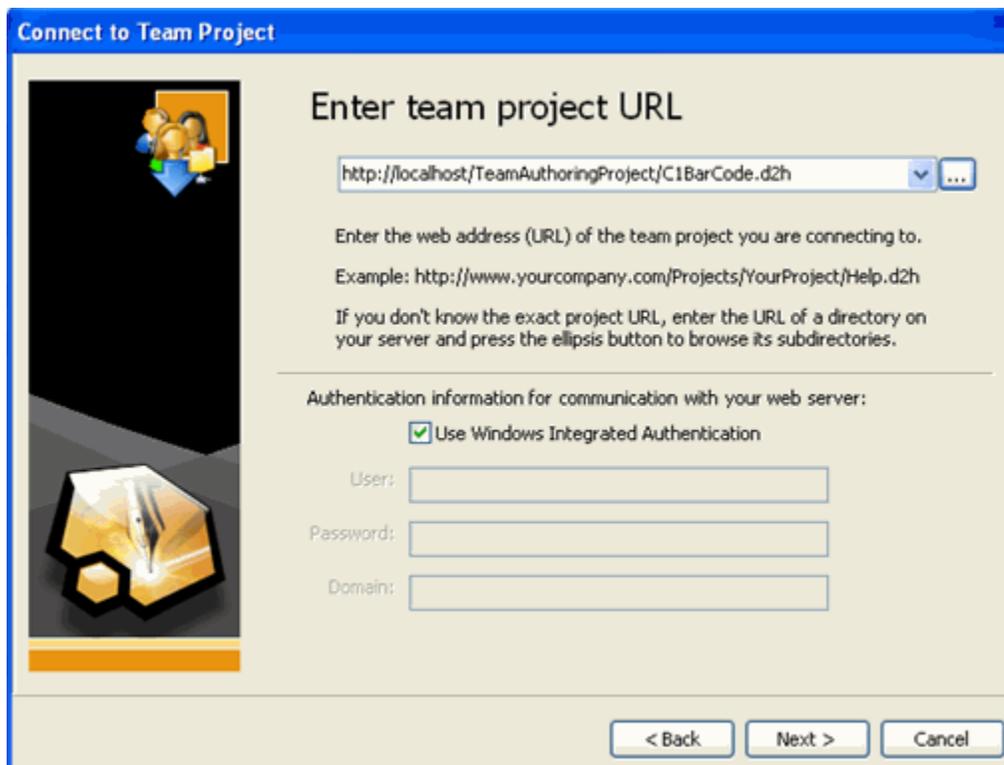
If you or another team member has shared a project with a Web Repository, use the Web connection to download a working copy.

To connect to and download a team project from a Web repository:

- Open Doc-To-Help and select **File | Connect to Team Project**.
- Under **Select repository type**, choose **Web Repository**. The team project is located on a Web server.

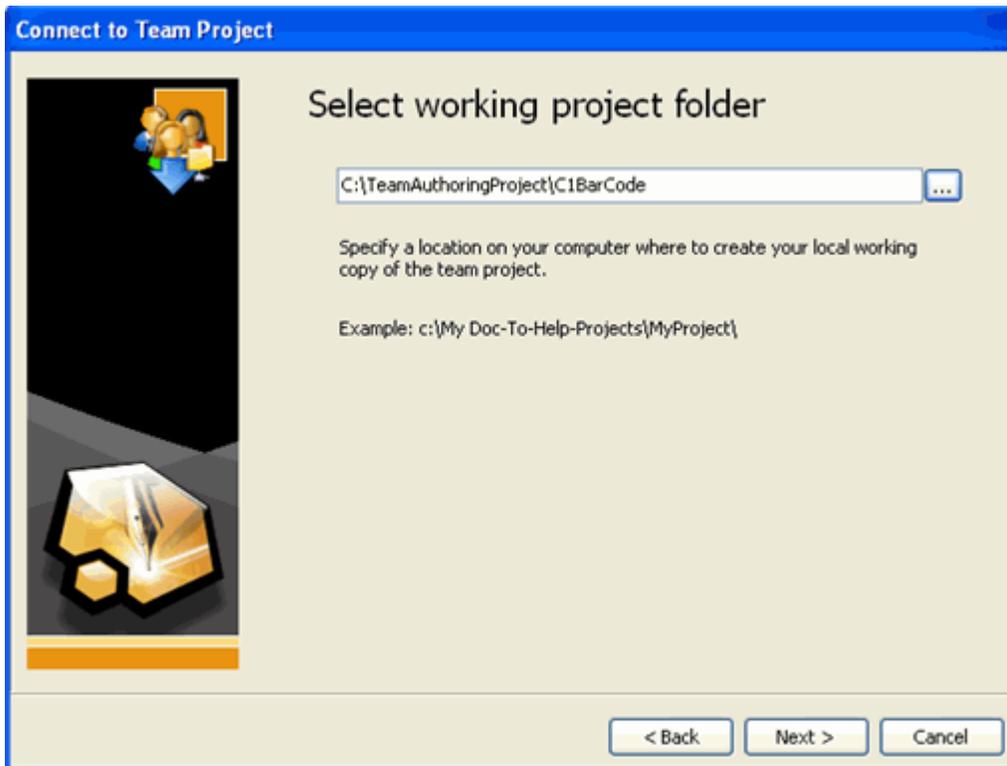


3. Click **Next** to continue.
4. Enter the URL of the virtual directory or location of the team project. Notice that you can click the **ellipsis** button and browse the virtual directory as well as create folders.

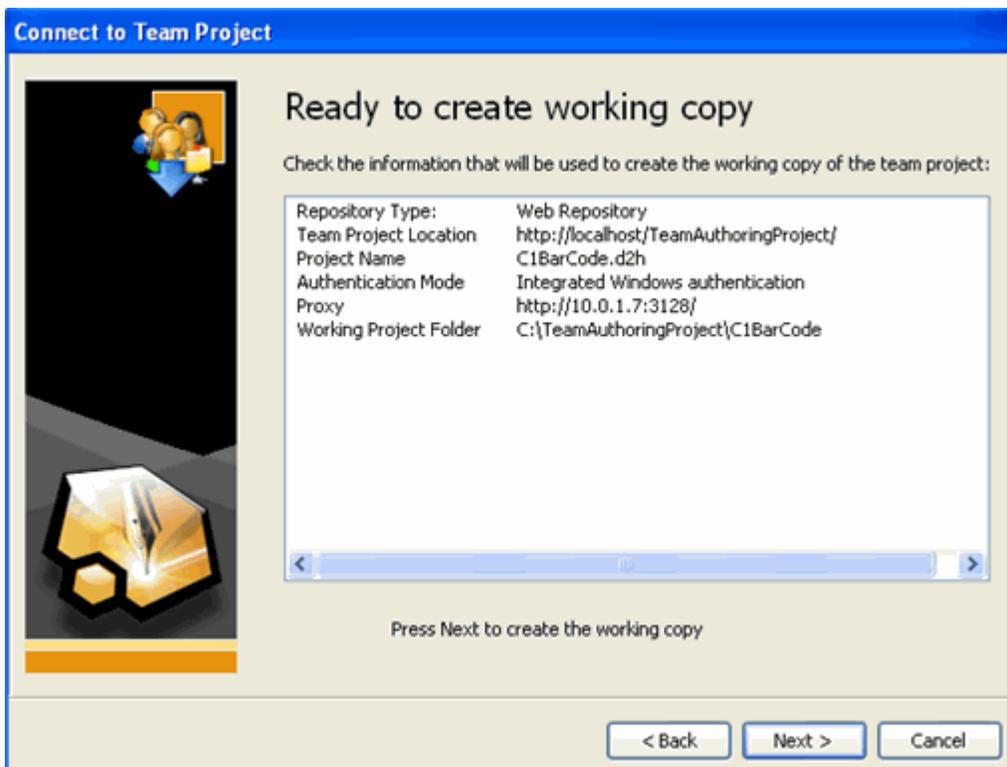


5. If you are not using **Windows Integrated Authentication**, uncheck this check box and enter your login credentials. Note that these credentials must be entered before you try to browse the virtual directory.

- Click **Next** and specify a location for the working project folder. This will be the location of the project in which you will work.



- Click **Next** and confirm the working copy information.



8. Click **Next** to create the working copy. Once the files have been downloaded, you will have a local copy of the project in which to work. Simply open this working copy as you would any other Doc-To-Help project. You do not need to connect to and download the team project every time you open Doc-To-Help.
9. Click **Close** and then click **Finish**. Your working copy of the project opens.

Working with a Team Project

You generally check files out from the repository to edit them and then check them in to upload the changes to the repository. If you are adding new files to a project, they only exist in your working copy until you check them into the repository.

There are three different kinds of files with respect to their role in team authoring:

- **Documents:** These are the source documents, or Word and/or HTML files included in the document tree of the Doc-To-Help project.
- **Auxiliary files:** These are any files, usually graphic files, in the source file folders that are not documents included in the document tree. Source file folders are specified in the **WordSourceFolder** and **HTMLSourceFolder** properties; the default folder names are WordDocuments and HTMLDocuments. These files are only accessible from the **Team Authoring** node of the project editor.
- **Project file (.d2h extension):** This is the project itself, which contains all the information such as the document tree, table of contents, index, and so on.

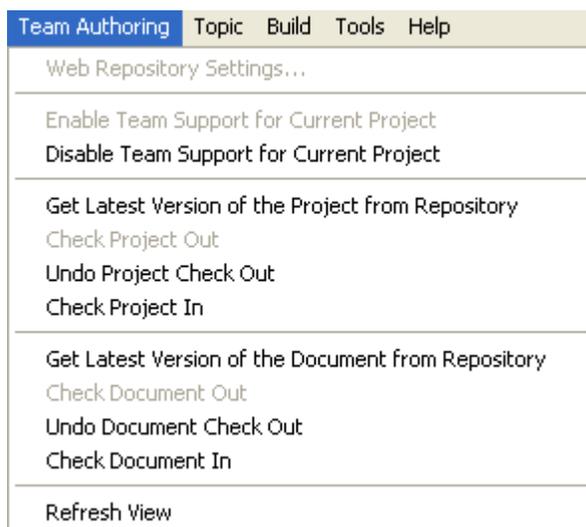
There are several different ways to access team-authoring functions for your documents and projects using the Doc-To-Help interface. Once you have a working copy of a team project, you have the option of using the **Team Authoring** menu, the team-authoring buttons on the **Documents** toolbar, or the **Team Authoring** node.

Accessing Team Authoring Functions

Once you have shared your project or connected to a team project, some changes occur in your working copy of the project. The **Team Authoring** menu is added to the **Menu Bar**, team-authoring buttons appear in the **Documents** toolbar, and the **Team Authoring** node appears under the **Source Files** folder when the **Project** icon is clicked. Each of these three areas of the Doc-To-Help interface provides access to the team-authoring features.

The Team Authoring Menu

The **Team Authoring** menu can be found in Doc-To-Help's main **Menu Bar**. This menu contains all of the important team-authoring features.

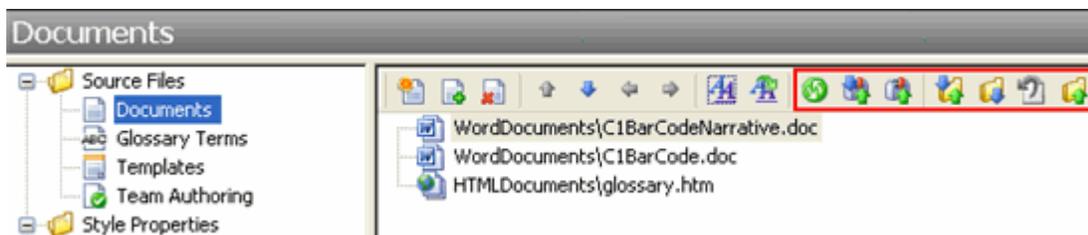


The **Team Authoring** menu is made up of the following options:

Web Repository Settings	Set Web repository location and login.
Enable Team Support for Current Project	Enables team support when it has previously been disabled.
Disable Team Support for Current Project	Disables team support. Typically used only to build Help targets.
Get Latest Version of the Project from Repository	Downloads changes from the team project and updates your working copy.
Check Project Out	Checks out and locks the Doc-To-Help project file (.d2h).
Undo Project Check Out	Reverses the project check-out process.
Check Project In	Checks in and uploads the project file (.d2h) and any changed or added documents to the repository.
Get Latest Version of the Document from Repository	Downloads changes from the team project and updates your working copy.
Check Document Out	Checks out and locks the document.
Undo Document Check Out	Reverses the document check-out process.
Check Document In	Checks in and uploads the document to the repository.
Refresh View	Communicates with the repository and refreshes the status of files under team-authoring control.

The Documents Toolbar

The documents can be viewed in the **Document Tree** of your working copy of the team-authoring project. You will notice that there are several icons in the **Documents** toolbar that are not available in non team-authoring projects.



To view the Document Tree and toolbar icons:

1. On the **Icon** bar, click **Project**.
2. From the left pane, select **Documents**. All of the documents in your project are listed in the right pane, and the **Documents** toolbar is visible above the document list.

Team Authoring Toolbar Buttons

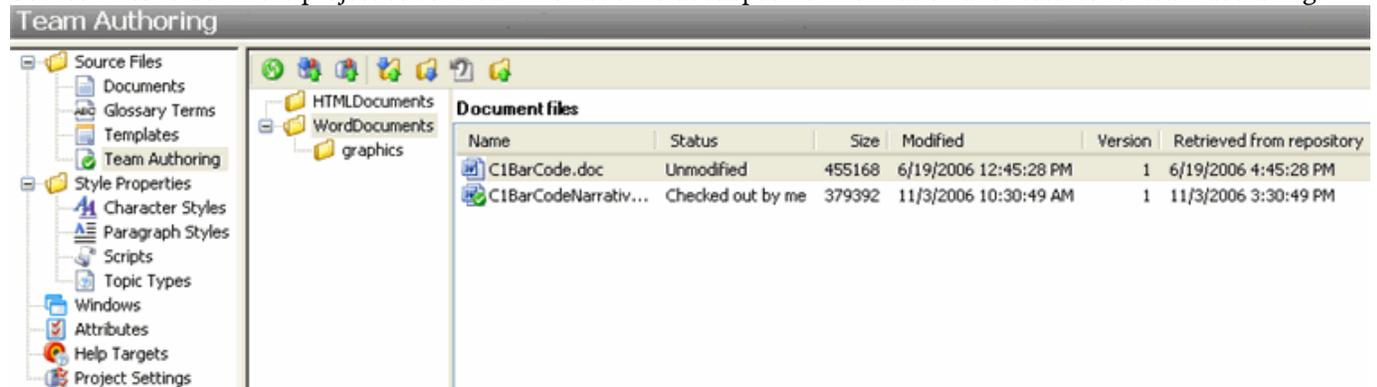


From left to right, the team-authoring buttons in the Documents toolbar include:

Refresh View	Communicates with the repository and refreshes the status of your documents.
Get Latest Version of the Project and Documents	Downloads changes from the team project file (.d2h) and documents and updates your working copy.
Check Project and Documents In	Checks in and uploads the project file (.d2h) and any changed or added documents to the repository.
Get Latest Version of the Document from Repository	Downloads changes from the team project and updates your working copy.
Check Document Out	Checks out and locks the document.
Undo Document Check Out	Reverses the document check out process.
Check Document In	Checks in and uploads the document to the repository.

The Team Authoring Node

To access the **Team Authoring** node, select the **Project** icon in the Icon bar and choose **Team Authoring** from the **Source Files** folder in the project editor tree. This node includes specific information and features for team authoring.

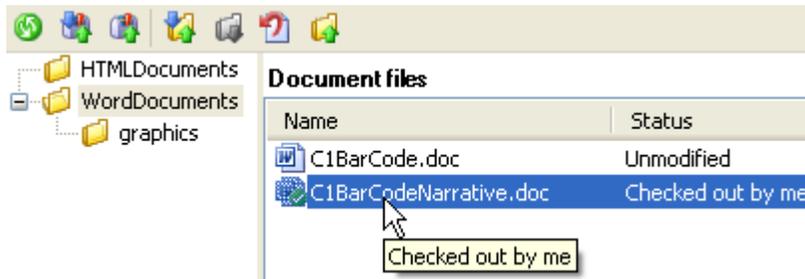


From the **Team Authoring** node, you can see a list of **Document files** and **Auxiliary files** lists for each of your source file directories. The source file directory tree is shown to the left of the file lists. In the **Team Authoring** node, you can obtain additional information, such as the version number or date the document was retrieved from the repository, pertaining to team authoring, and you can perform team-authoring actions on individual files. Using this node of the project editor is not required for source documents unless you need to look at advanced team-authoring information, such as the document status, or sort by that information. This node is also not required for auxiliary files unless you want to perform team-authoring actions on them individually or to check them out for editing.

Team-authoring actions on individual files are available from the context menu and by using the **Team Authoring** toolbar buttons when you select the file. Note that the **Team Authoring** toolbar buttons that appear in this node are identical to the ones on the **Documents** toolbar in the Document Tree.

The information available for each document includes document status, size, date modified, version number, and date the document was retrieved from the repository.

The check-out status of each source file and auxiliary file appears in the *Status* column, but it can also be determined by the file icon and tooltip. For example, a document that is checked out has a checkmark icon and a tooltip stating by whom the document is checked out.



Possible status values include:

Status	Description
Checked out by me	Checked out by this author.
Checked out by "<user name>" on "<computer name>"	Checked out by another author, who is specified in <user name>.
New document, not in repository	File newly added to your working copy of the project; it does not yet exist in the repository.
Modified by me	Modified by current author.
Modified in repository	Modified by another author.
Unmodified	Synchronized with the repository. No changes made by current or other authors since it was last retrieved or sent to the repository by this author.

The version number is the sequential number of the file's version in the repository. When a file is first sent to the repository, it is assigned a version number of 1. Each time it is modified in the repository by any author, its version number is incremented.

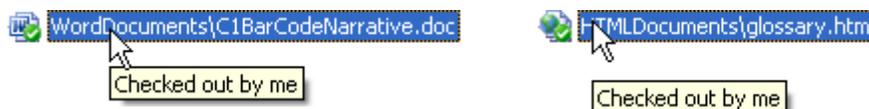
To refresh the file information or retrieve the latest information from the repository, use the **Refresh View** button  in the toolbar.

Working with Documents

Working with documents is the easiest and most common task you will perform when using team authoring. There are three, basic document functions you can perform in your working copy of the team project.

- Check Out

Checking out a document labels it as reserved by the user who checked it out, and it is locked in the repository. Unless you check out a document, the document file is read-only, except for newly added documents. To check out a document, select it, click the **Team Authoring** menu and select **Check Document Out** or use the **Check Document Out** button in the **Documents** toolbar and **Team Authoring** node. Document check-out status is shown by an icon and a tooltip in the document tree:



If you change your mind and do not want to modify a file, you can undo the check out command using the **Undo Document Check Out** option in the **Team Authoring** menu or using the **Undo Document Check Out** button in the **Documents toolbar** and **Team Authoring** node.

- Check In

Checking a document in unlocks the document and sends the changes to the repository; the changes are then made to the team project. You can check in individual files using the **Check Document In** button in the **Documents** toolbar or in **Team Authoring** node, but recommended practice is to check in all of your changes to the project, including all documents and files, using the **Check Project In** option on the **Team Authoring** menu.

- Get Latest Version

You can use the **Get Latest Version of the Document from Repository** menu option to retrieve current versions of documents from the repository. This option is available for individual files in a document's context menu, in the **Documents** toolbar, and in **Team Authoring** node, but recommended practice is to get latest version of the whole project, including all documents and files, using the **Get Latest Version of the Project from Repository** option in the **Team Authoring** menu. If the file was deleted from the team project by another user, getting its latest version deletes that file from your working copy.

You can edit, add, or delete documents in the team project, but certain document functions must be performed to ensure that you have the latest version and that you are providing the latest version for other team members.

- Editing Documents

A document must be checked out of the repository in order to make edits. Only one author is permitted to edit a document at any given time; therefore, if another team member has a document checked out, you cannot check it out and make edits until he/she checks it back into the repository. Doc-To-Help will ask you if you want to check a document out when you open it from the **Document Tree**.

- Adding Documents

When you add a document to the project, it is neither checked in nor checked out because it only exists in your working copy. To add the new document to the team project, simply check it into the repository. Once it is checked in, it will be available to other authors after they get the latest version of the project.

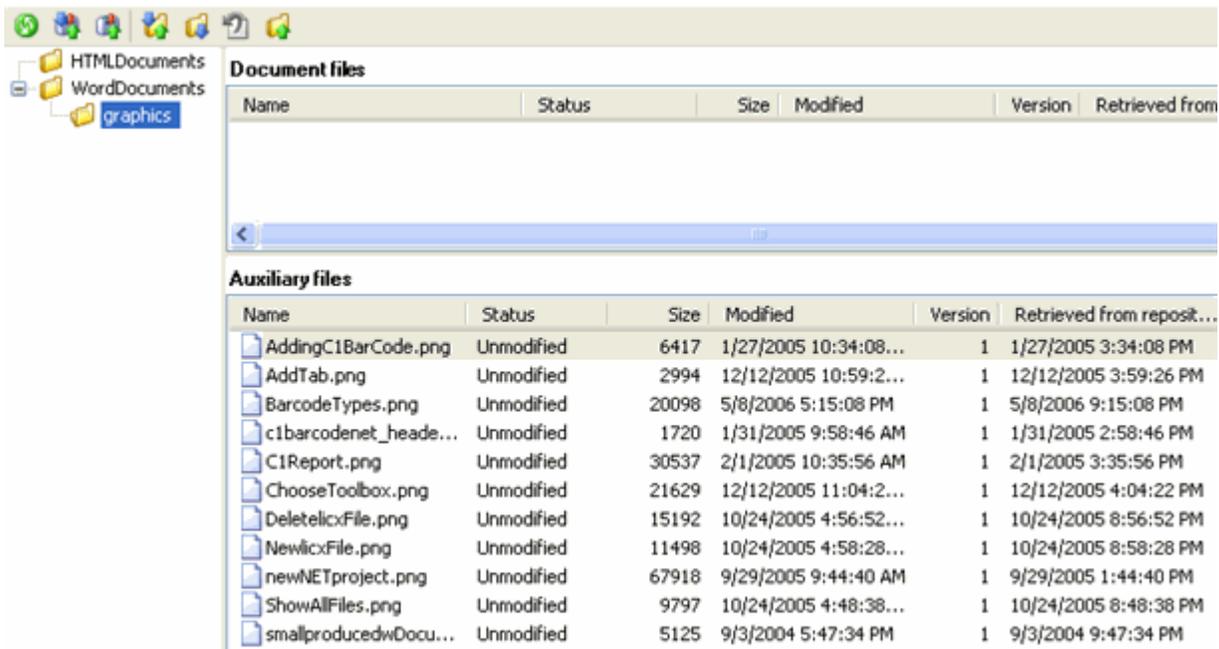
- Deleting Documents

You can delete an existing document from the team project only if you have the project checked out, but you can delete a new document that you have added at any time as long as it has not yet been checked into the repository.

Working with Auxiliary Files

Auxiliary files are any files, usually graphic files, in the source file folders that are not documents included in the **Document Tree**. Source file folders, or the folders where all of the project's source documents are located, are specified in the **WordSourceFolder** and **HTMLSourceFolder** properties; the default folder names are *WordDocuments* and *HTMLDocuments*.

These files are only visible from the **Team Authoring** node of the project editor. For example, here is a list of auxiliary files that are located in a *graphics* folder in the *WordDocuments* source file folder.



Auxiliary files are checked out and checked in the same way as documents. You can select a file and use the **Team Authoring** menu or the **Team Authoring** toolbar buttons in **Team Authoring** node. Note that since these files are not listed in the **Document Tree**, you cannot use the **Documents** toolbar to perform team-authoring functions on these files.

Like documents, auxiliary files are read-only unless checked out, except when they are newly created. Unlike documents, however, you do not need to add them to the project explicitly; they are added automatically and shown in the **Team Authoring** node once they are created in the source file folder.

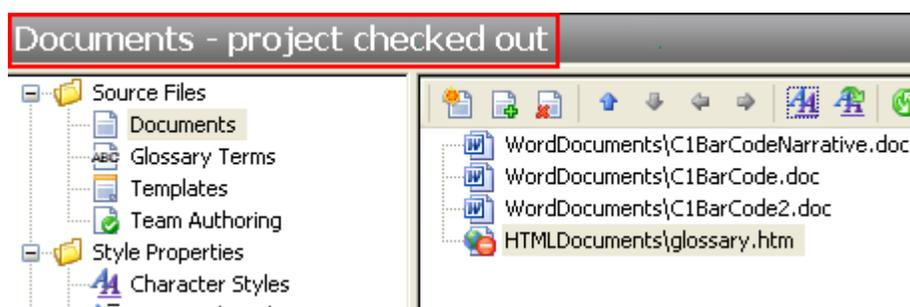
Working with the Project File

The Doc-To-Help project file (.d2h) contains all project information, such as the **Document Tree**, topics, table of contents, index and so on.

The three basic document functions also apply to the Doc-To-Help project file itself.

- Check Out

Checking out a team project allows you to make any changes to the project through the Doc-To-Help project editor in your working copy of the project. Only one author at a time can check out a project. When you have a project checked out, the title area of the window changes to reflect this.



- Check In

Checking a project in uploads any documents that have been changed, along with the project file, into the repository; the changes are made to the team project.

- Get Latest Version

Getting the latest version of the project downloads the team project to your machine and updates your working copy.

Working with project files in team authoring is more complex than working with documents. While some changes can be made to the project file without checking the project out, others cannot. There are generally two different kinds of changes you may need to make in a project:

1. **Global changes:** These are massive changes, such as customizing the table of contents or index, adding Help targets, styles, and adding project properties. To make such changes to the project, you need to check the project out, make your changes, and then check the project back in. This ensures only one author is making such changes at any given time, preventing any conflicts.
2. **Minor changes:** These are minor changes related to the changes you are making in new documents or in documents you have checked out. This includes adding a document to the project and changes in the project that Doc-To-Help makes automatically when you add D2HML hot spots to a document. For example, a keyword is automatically added to the index when you create a new keyword hot spot in the document. These changes can be made simultaneously by multiple authors working on a project; there is no need to check the project out to make them. These changes are made locally, in your working copy. They become available to other authors when you check your project in.

Global Project Changes

If you need to make massive changes to a team project, check the project out, make your changes, and then check the project in. Only one author can do this at a time.

If you check the project out but then decide you do not want to change it, or you want another author to make those changes, you can use the **Undo Project Check Out** option on the **Team Authoring** menu to cancel the check out. However, be aware that all your changes in the project will be discarded and the project will be retrieved from the repository. This does not affect your documents; if you made changes to them, they will not be discarded.

Note that you do not need to check the project file out if you are adding anything new to the project.

Global changes include, but are not limited to:

- Deleting existing documents
- Changing existing document properties
- Changing project properties
- Changing Help target properties
- Changing style properties
- Changing topic properties
- Editing the table of contents

Minor Project Changes

Normally, authors work within their working project copies without checking the team project out. In that case, authors are not allowed to make global changes to the project. More exactly, authors are not allowed to change project data that was created by other authors and already existed in the repository when they got the latest version.

Note: Only one author at a time is allowed to check the project out.

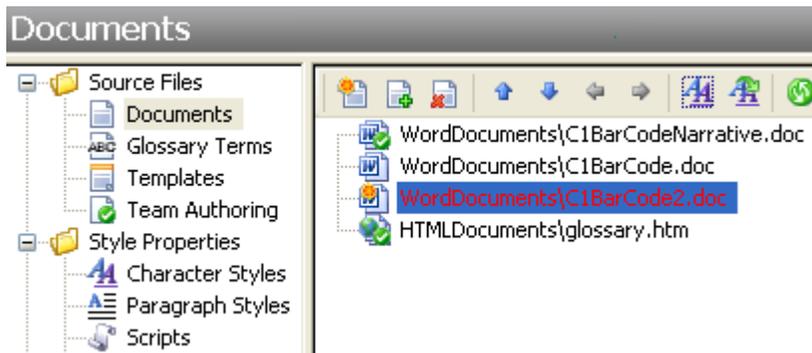
However, authors are allowed to add information, such as documents, keywords, and so on, to the project. For example, you can add a document, delete a document that you added, add a keyword or group, delete a keyword or group that you added, or associate a keyword with a topic. Note that the latter is allowed because it is an addition, not

a modification of pre-existing data. Newly created project items are highlighted by a special color so you can see what can be modified in the project.

To customize the color of newly created project items:

1. In the Doc-To-Help project editor, select the **Tools** menu and choose **Options**.
2. Click the **Team Authoring Control** tab.
3. Click the Text color button and specify a color. The text of new project items will appear in the specified color in the Doc-To-Help project editor.
4. Click the **Highlighted text color** button and specify a color. The text of new project items will appear in the specified color when they are selected, or highlighted, in the Doc-To-Help project editor.

In this example, the highlighted text color was set to red. The new document, C1BarCode2.doc, was added to the project.



Modifying project items that are already shared by the team, in the sense that they already existed in the project when you last got it from the repository, is not allowed. You can modify only the items that you have created and have not yet sent to the repository. If you need to make changes to the project that are not allowed in this mode, you can always check the project out, make your changes, and then check the project in.

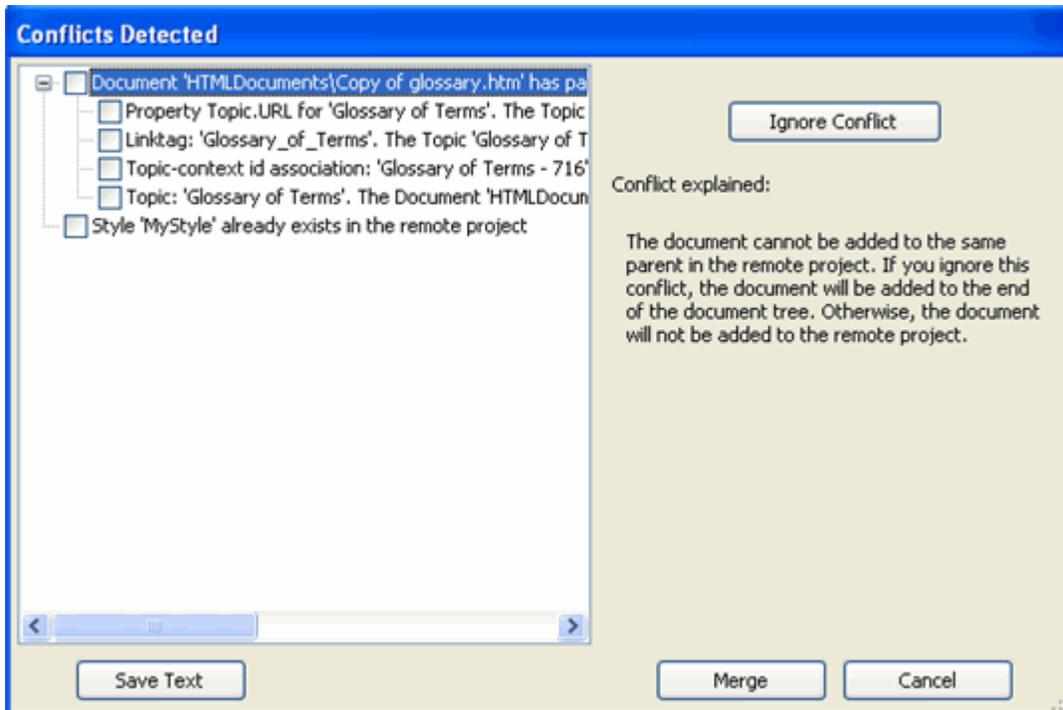
To get changes made to the project and documents by other authors from the repository:

In the Doc-To-Help project editor, click the **Team Authoring** menu and select **Get Latest Version of the Project from Repository**. Note that this will get all changes from the repository, including changes in the documents and auxiliary files.

Your working project will be replaced by the repository project, but any changes you made in your working copy will be preserved. In most cases, your changes will be preserved entirely, but occasionally there may be conflicts between your changes and the current project in the repository. Such conflicts will arise only if another author adds items with names identical to yours to the project, or if an author checks out the project, makes global changes, and then checks it in while you are making minor changes to your working copy.

For example, if you add a document as a child of another document, the parent document, while another author deletes the parent document from the project, your child document will not be added. Some conflicts can be ignored, and a reasonable default action can be taken. For example, if you and another author add a style with the same name, you can rename your style. However, if a conflict cannot be ignored, your item will not be added to the project.

If conflicts exist when you get the latest version of the team project, Doc-To-Help shows a **Conflicts Detected** dialog box.



You can specify which conflicts you want to ignore for the ones that can be ignored. Simply click the **Ignore Conflict** button.

Click the **Save Text** button to save the conflicts in a .tmp file for your reference.

If you click **Merge**, Doc-To-Help will resolve the conflict by merging the changes, which may result in your changes not being added to the project in the repository.

To send the changes you made to the repository:

In the Doc-To-Help project editor, click the **Team Authoring** menu and select **Check Project In**.

The data that you added to the project will be added to the team project in the repository and made available to other users.

Synchronizing Projects and Documents

There are two types of project synchronization:

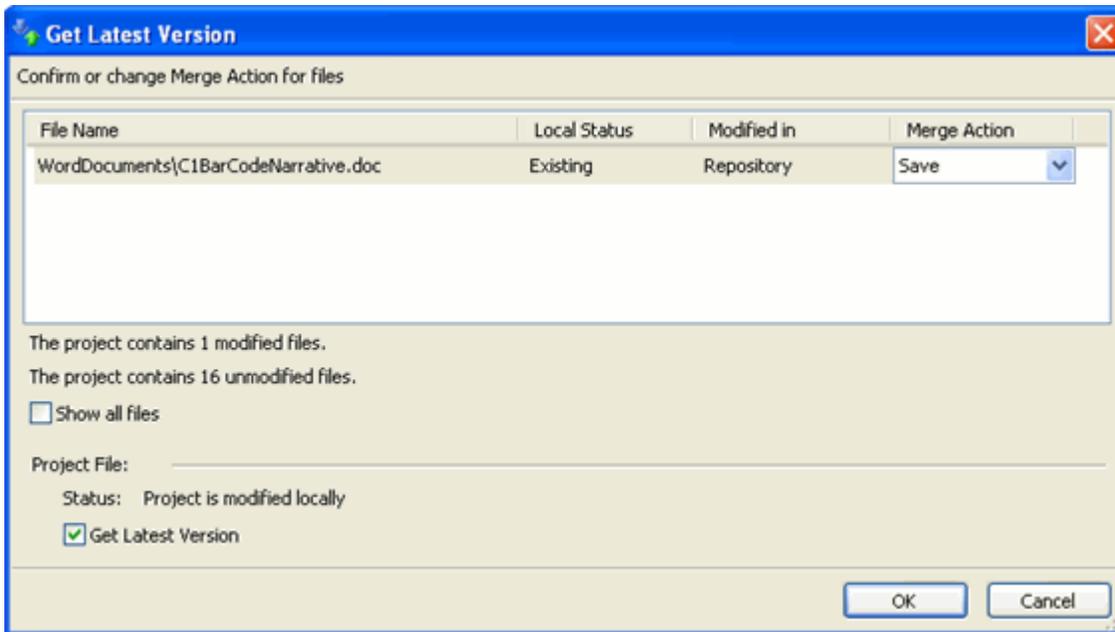
- | | |
|--|---|
| Getting the latest version of the project from the repository | When you do this, you are updating your working copy of the project with the latest team project. This is typically used when other authors have made changes and you need to update your working copy. |
| Uploading your changes to the repository | When you do this, you are updating the team project with changes you have made in your working copy of the project. This is typically used when you need to make your changes available to other authors. |

Getting the Latest Version of a Team Project

Although you can get individual files from the repository separately, we recommend that you retrieve all changes made by other authors from the repository at once.

Open Doc-To-Help, click the **Team Authoring** menu and select **Get Latest Version of the Project from Repository**. The **Get Latest Version** dialog box appears. It shows a list of modified files and states whether the project file in the repository has been modified.

Note: Files that you deleted from your working copy of the project do not appear in this dialog box.



If you would like to see all the files in the project, check the **Show all files** checkbox. You can sort on any of the columns in the list by clicking a column header.

Note: If you would like Doc-To-Help to retrieve the latest version of the team project, check the **Get Latest Version** checkbox. It will not retrieve the project file (.d2h) if that checkbox is unchecked.

The **Get Latest Version** dialog box provides the following information about the project's files:

- Local Status

The *Local Status* column provides the location of a file, whether it is in the repository, in your working copy, or in both.

Added locally The file has recently been added to your working copy of the project. It does not yet exist in the repository. Note that this status can only be shown if the **Show all files** checkbox is checked; otherwise, it will not appear.

Added in repository The file has been added to the repository, but it does not yet exist in your working copy.

Existing The file exists both in the repository and in your working copy.

- Modified in

The *Modified in* column provides information on where a file has been modified, whether it is in the repository, locally, both, not at all or if it has been deleted from the repository.

Repository The file has been modified in the repository.

Repository deleted The file has been deleted from the repository.

Unmodified The file has not been modified in the repository and has not been modified in your working copy. The file in the repository and in your working copy are identical. Note that **Unmodified** will only appear if the **Show all files** checkbox is checked since only files modified in the repository are shown by default.

Local	The file has been modified in your working copy but not in the repository. Local will only appear if the Show all files checkbox is checked since only files modified in the repository are shown by default.
Repository and local	The file has been modified both in the repository and in your working copy. This should never occur under normal circumstances, because only one author can check out a file at a time. However, this can occur if you manually disable the read-only attribute of a file and then modify it, which is strongly discouraged; it can disrupt team-authoring control and cause data loss.

- Merge Action

The *Merge Action* column allows you to determine what action will be performed on the file.

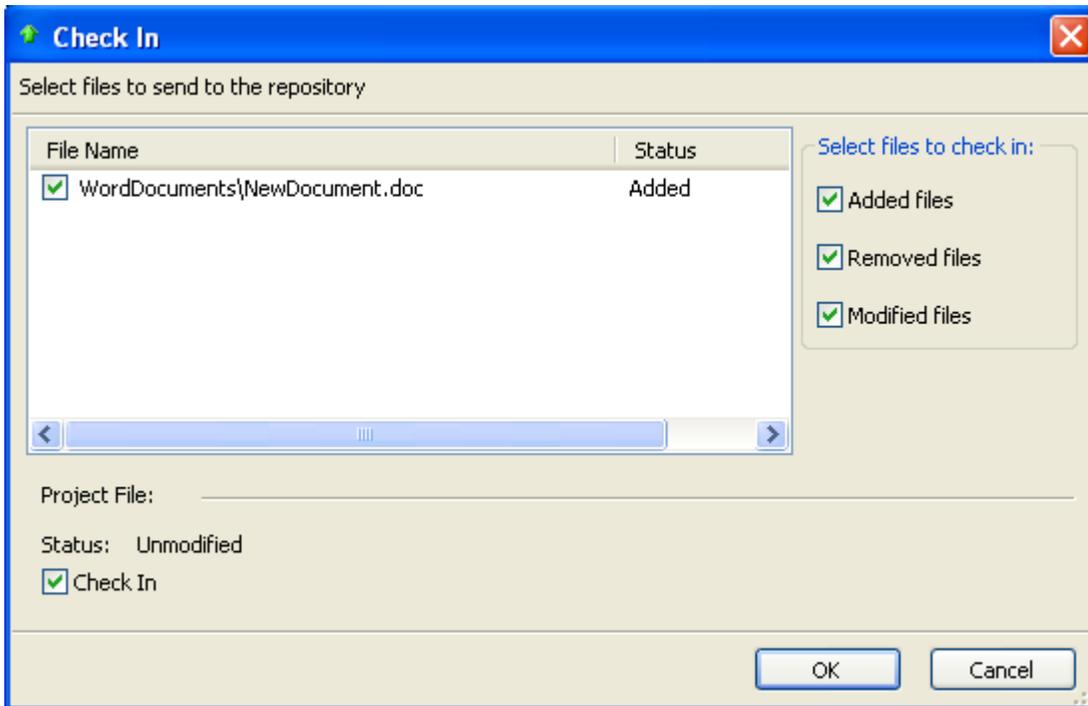
Save	The file will be retrieved from the repository and will replace the file in your working copy. This is the default action for a file modified in or added to the repository.
Delete	The file will be deleted from your working copy. This is the default action for a file deleted from the repository; it is only available for files that have been deleted from the repository.
Restore	The file will be retrieved from the repository. This is the default action in the rare occasion when a file is deleted locally, and the same file is added to the repository by another author; this action is available only in situations such as this.
None	No action is taken; the file is left as it is.
Copy to temp folder	The file will be retrieved from the repository, but it does not replace your local file; instead, the repository file is saved in the <i>D2HTempFolder</i> folder located within your local, working copy folder.

In most cases you will confirm the merge action that Doc-To-Help suggests for the files. If you need specific handling of some files, you can change the merge action to other, non-default options if they are available for a given file in the **Merge Action** drop-down box. If you change the merge action, exercise caution and make sure you fully understand the consequences. Be aware that once you get the latest version of a project, all files are considered synchronized with the repository; therefore, the next time you select **Team Authoring | Get Latest Version of the Project from Repository**, you will not see the files in the list of the repository-changed files unless they were changed after you got the latest version of the project from the repository. If necessary, though, you can retrieve these files from the repository by checking the **Show all files** check box or individually in the **Document Tree** or **Team Authoring** node.

Uplodading Changes to the Repository

Although you can check individual files into the repository separately, we recommend that you check in the entire project with all of your changes at one time.

Open Doc-To-Help, click the **Team Authoring** menu and select **Check Project In**. The **Check In** dialog box appears.



This dialog box shows the list of files changed in your working copy and specifies the kind of change in the *Status* column. The types of changes include **Added**, **Removed** and **Modified**. The **Check In** dialog box also states whether the file in your working copy is modified.

Once you click **OK**, your changes are sent to the repository. If necessary, you can choose to send only the changes in the source files without sending the changes in the project file by unchecking the **Check In** check box under **Project File**. This is an advanced option and is not recommended unless you have a specific reason for doing so. If you only want to send some files to the repository and not others, you can uncheck the desired check boxes in the file list, although this is also an advanced option that is not recommended.

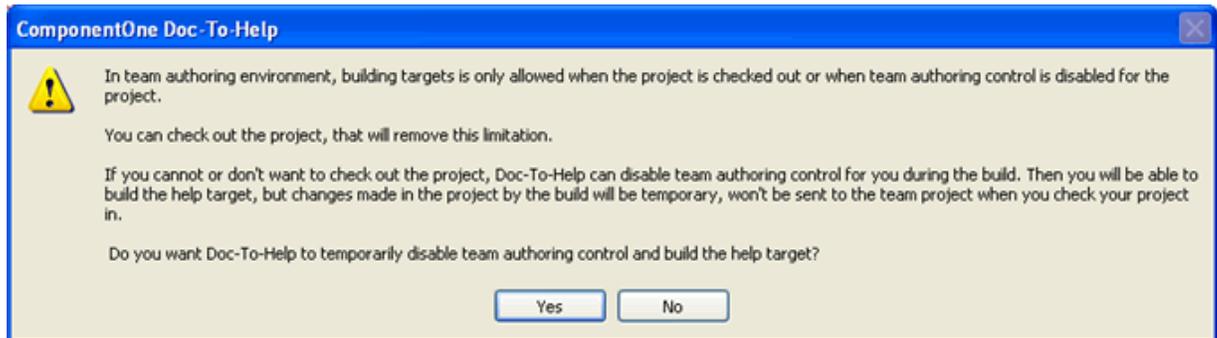
Note: If you do not check out the project, your changes in the project file will be merged into the project in the repository, and that merge process may, on rare occasions, encounter conflicts that must be resolved in the **Conflicts Detected** dialog box.

Building Help Targets

Building a Help target in Doc-To-Help modifies the project. For example, a D2HML keyword hot spot in a multi-topic document creates a keyword in the project when a Help target is built. Since changes in the project must be controlled in a team-authoring environment, Doc-To-Help cannot build a Help target while the project is under team-authoring control. However, you need to build Help targets, so Doc-To-Help allows you to do this in the following ways:

1. You can check out the project by clicking **Check Project Out** in the **Team Authoring** menu and then build your Help targets and modify the project without any restrictions. But bear in mind that after you check out the project and make your changes, which may include building a Help target, you must check your project into the repository so the changes become permanent to all other authors on your team. If you change your mind and do not want to make your changes permanent, you can select **Team Authoring | Undo Project Check Out**, but you will lose any changes you made after you checked the project out, because the project is retrieved from the repository. Note that checking out the project is not always possible, because only one team member can do it at any given time. If someone else checked out the project and has not yet checked it in, you will not be able to check it out, so you will not be able to build Help targets this way.
2. You can temporarily disable team-authoring control for the duration of the Help target build. Disabling team-authoring control means that you can make any changes to the project without restrictions, but those changes will be temporary in the sense that they will only exist while team-authoring control is disabled. They will be discarded when you enable team-authoring control, and the project will return to the state it was when you

first disabled team-authoring control. Disabling team-authoring control is the default option when you build a Help target while your project is not checked out. If you build a Help target without checking out the project, Doc-To-Help shows a dialog box suggesting you disable team-authoring control for the duration of the build.



If you click **Yes**, team-authoring control will be disabled while the build is in progress and will be automatically enabled at the end of the build. This means that you will be able to build the Help target, but any changes in the project, such as added topics, keywords, and so on, made by Doc-To-Help during the build will be discarded once the build is complete.

3. You can disable team-authoring control using the **Disable Team Support for Current Project** option on the **Team Authoring** menu. Then you will be able to build Help targets and make any changes to the project without any restrictions, and you will see the changes made during the build once the build is complete. These changes will be temporary in the sense that they will only exist while team-authoring control is disabled. They will be discarded when you enable team-authoring control. The project will return to the state it was when you disabled team-authoring control. This is similar to option two above, only Doc-To-Help will not automatically enable team-authoring control at the end of the build. You must select **Enable Team Support for Current Project** from the **Team Authoring** menu to do this.

If your intention is only to see the Help target built from your project, you can use option two and build without checking out the project.

If you need to see the project changes made by Doc-To-Help during the build, then you must check out the project. Then you will be able to build Help targets without any restrictions.

Maintaining a Team Project

Doc-To-Help provides two methods of team-project maintenance in order to help you prevent and resolve problems.

- Backup files in the repository

By default, Doc-To-Help keeps all versions of all files, including source files, auxiliary files, and the .d2h project file, sent to the repository by all team members every time they are checked into the repository. These files are kept in the **Backup** folder in the repository. They come in handy if someone accidentally removes or overwrites changes made by other team members or if some other undesirable situation occurs. The backup files can be used to recover lost data. However, this also means that the disk space occupied by the repository grows rapidly when members check in their changes, especially if the .d2h project file is checked in frequently.

To conserve the repository disk space, an administrator should delete unneeded backup files from time to time. This can be done using a special administrative utility included with Doc-To-Help, **C1D2HTeamAdmin.exe**. The administrator can delete all backups older than a certain date there. This can also be done manually simply by deleting all or part of the contents of the **Backup** folder in the repository. Each version number has its own sub-folder inside the **Backup** folder, so you can delete only particular versions, if necessary. It is a good idea to make reserve copies of the folders you are deleting to a backup storage device, such as a DVD or tape.

- Unlocking files in the repository

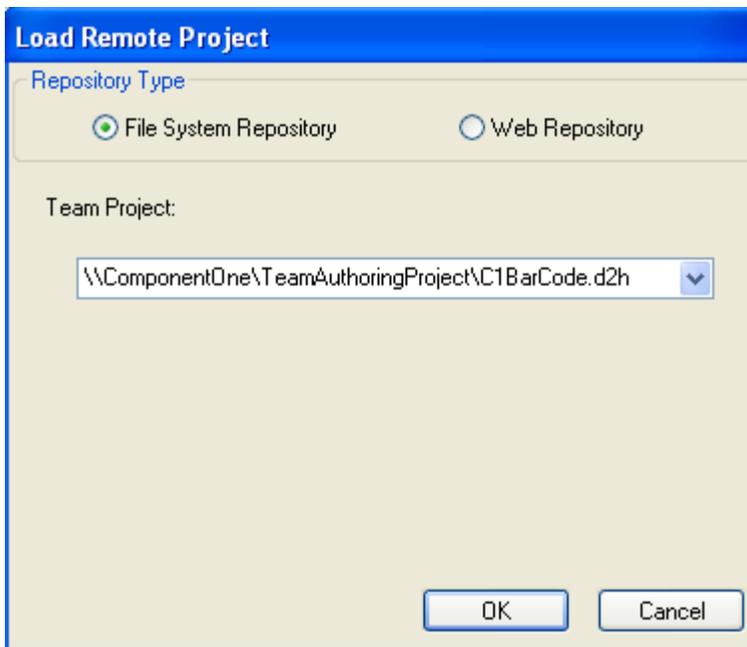
Checking out a file creates a lock in the repository that prevents other team members from checking out the same file. This lock is needed to ensure that a file cannot be checked out by more than one author

simultaneously. This may be problematic if an author checks out a file and forgets, is unable to check it in, or cannot undo the check out for an extended period of time, such as when he/she goes on vacation. Then the file remains permanently locked, preventing the other team members from checking it out and modifying it. If such a situation occurs, it requires an administrator to manually remove the lock from that file. This can be done using a special administrative utility included with Doc-To-Help: **C1D2HTeamAdmin.exe**. The administrator should run the utility from any machine that can connect to the repository.

Removing Backup Files from the Repository

To delete up backup files in the repository:

1. Double-click **C1D2HTeamAdmin.exe**. This file is installed, by default, to C:\Program Files\ComponentOne\DocToHelp when Doc-To-Help is installed.
2. Select **File | Open Team Project** to connect to the repository. The **Load Remote Project** dialog box appears.
3. Select a repository type, either **File System Repository** or **Web Repository**, depending on the type of repository that was created for the team project.



4. Select a team project or choose **(Select Team Project)** from the drop-down box to browse for a project and click **OK**. The utility shows the folder tree of the repository and the files in each folder.
5. Select **Tools | Clear Repository Backup**. The **Clear Repository Backup** dialog box appears.

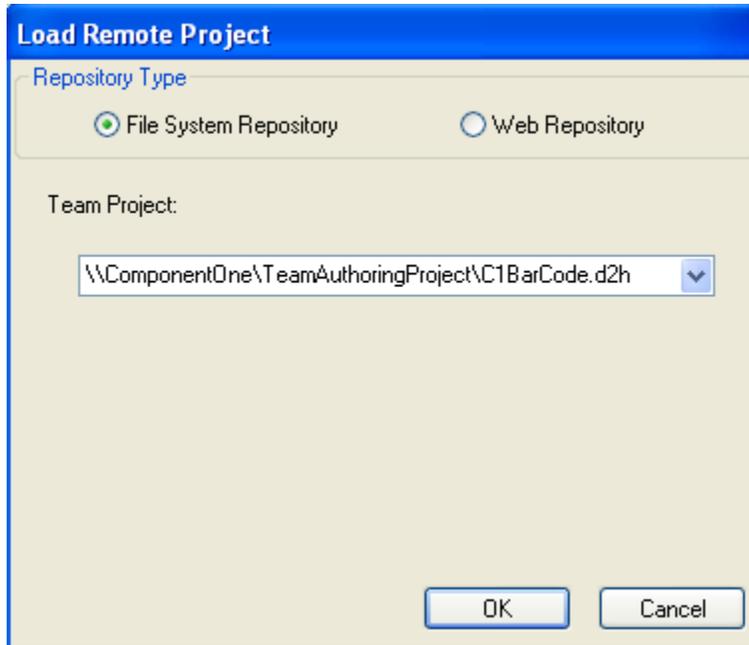


6. Click the drop-down arrow next to the date and select a date from the calendar.
7. Select a time using the up and down arrows in the time box.
8. Click **OK**. Backup files prior to the date and time specified will be removed from the repository.

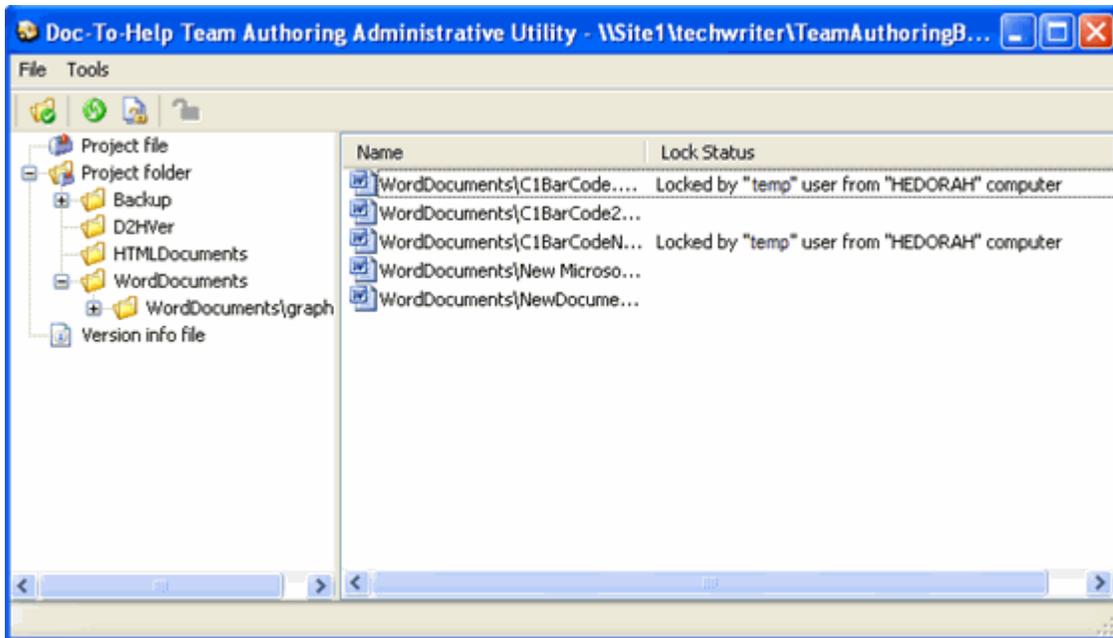
Unlocking Files in the Repository

To unlock files in the repository:

1. Double-click **C1D2HTeamAdmin.exe**. This file is installed, by default, to C:\Program Files\ComponentOne\DocToHelp when Doc-To-Help is installed.
2. Select **File | Open Team Project** to connect to the repository. The **Load Remote Project** dialog box appears.



3. Select a repository type, either **File System Repository** or **Web Repository**, depending on the type of repository that was created for the team project.
4. Select a team project or choose (**Select Team Project**) from the drop-down box to browse for a project and click **OK**. The utility shows the folder tree of the repository and the files in each folder. Files that are currently locked are indicated in the **Lock Status** column, showing the user name and the computer name that owns the lock.



5. Choose one of the nodes in the left pane and select a file from the right pane.
6. Click the **Unlock** button  on the toolbar or right-click the file and choose **Unlock** from the context menu.

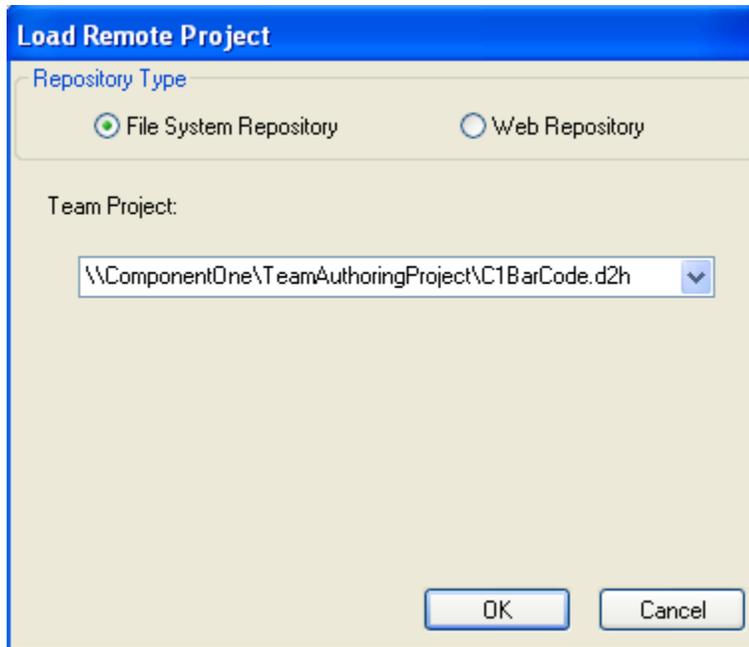
Note: Unlocking files must be done with caution and only by an authorized person, usually the administrator, because doing so resets the check-out state of the file on the team member's machine without getting the latest version of that file from the repository.

Unlocking the .dhv File

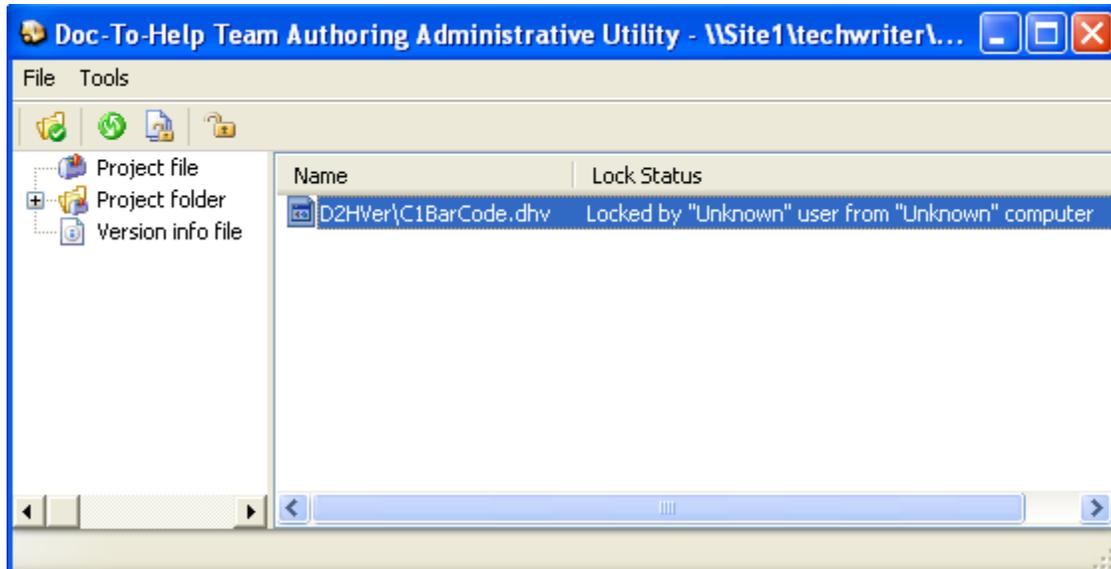
Most team-authoring actions create a temporary lock in the repository for the duration of the action to prevent conflicts between different team members using the repository simultaneously. This temporary lock is created in a special file, **<project_name>.dhv**, located in the **D2HTeamInfo** folder created by Doc-To-Help when the project is set up for team authoring. Normally, these locks are only temporary and are automatically handled by Doc-To-Help without user interference. However, if a team-authoring action is not normally completed or if it is cancelled on a team member's machine, for example, due to an unexpected crash or power failure, it may leave the lock permanently in the repository. If this happens, no team members will be able to use the repository, and Doc-To-Help will show errors saying the repository is busy doing other tasks, although no author is actually doing anything with the repository. This is a very rare occurrence, but it may happen. If it does, the administrator must use the administrative utility, **C1D2HTeamAdmin.exe**, to remove the lock from the **<project_name>.dhv** file.

To unlock the **<project_name>.dhv** file in the repository:

1. Double-click **C1D2HTeamAdmin.exe**. This file is installed, by default, to C:\Program Files\ComponentOne\DocToHelp when Doc-To-Help is installed.
2. Select **File | Open Team Project** to connect to the repository. The **Load Remote Project** dialog box appears.



3. Select a repository type, either **File System Repository** or **Web Repository**, depending on the type of repository that was created for the team project.
4. Select a team project or choose (**Select Team Project**) from the drop-down box to browse for a project and click **OK**. The utility shows the folder tree of the repository and the files in each folder.
5. Click the **Version info file** node and select **D2HTeamInfo\<project name>.dhv**.



6. Click the **Unlock** button  on the toolbar or right-click the file and choose **Unlock** from the context menu.

Note: You can use the **Unlock All** button  on the toolbar to release all locks from all files, but this should be done with extreme caution.

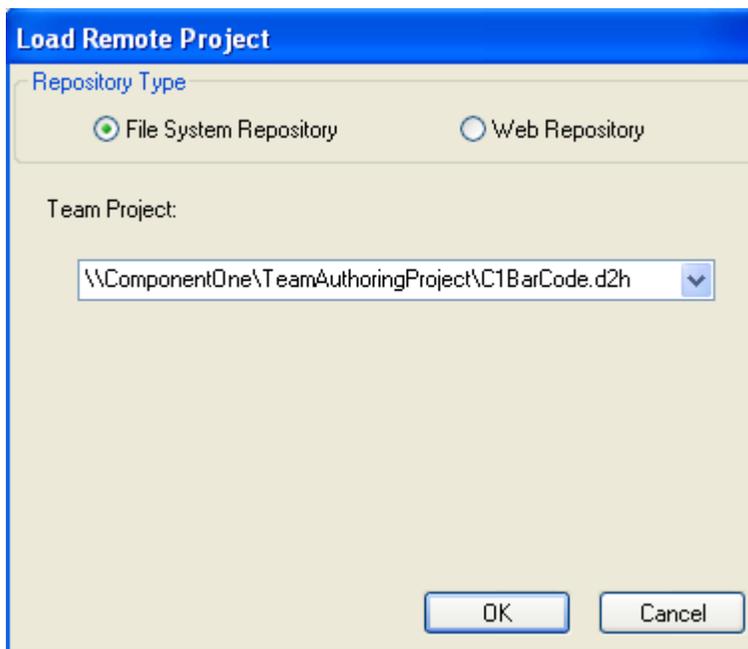
Upgrading a Team Authoring Project

All users of a team project must have exactly the same version, or same build number, of Doc-To-Help installed on their machines. Version incompatibility will be detected as an error. If you want to upgrade the project to a newer version of Doc-To-Help, the administrator must upgrade the team project in the repository and then each user can upgrade his/her working copy.

To upgrade the project to a new version of Doc-To-Help:

Note: The following steps should be performed by the administrator.

1. Open your working copy of the project in a new version of Doc-To-Help.
2. Double-click **C1D2HTeamAdmin.exe**, the administrative utility included with Doc-To-Help. This file is installed, by default, to C:\Program Files\ComponentOne\DocToHelp when Doc-To-Help is installed.
3. Select **File | Open Team Project** to connect to the repository. The **Load Remote Project** dialog box appears.



4. Select a repository type, either **File System Repository** or **Web Repository**, depending on the type of repository that was created for the team project.
5. Select a team project or choose **(Select Team Project)** from the drop-down box to browse for a project and click **OK**.
6. Select **Tools | Upgrade Team Project to the Current Doc-To-Help Version** to change the Doc-To-Help build number stored in the repository to the new build number.

Once the project is upgraded to the current version of Doc-To-Help, all users can upgrade their working copies of the project by opening them in the newer, or current, version of Doc-To-Help.

Removing Team Authoring Support from a Working Copy

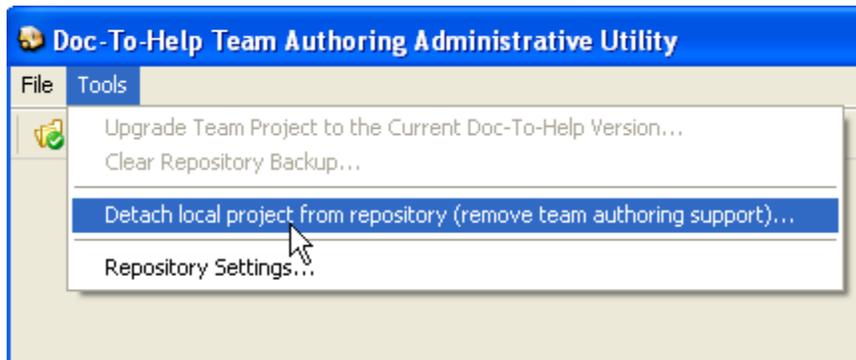
You can remove team-authoring support in a local, working copy of a project from the repository to which it is connected.

Removing team-authoring support is different from disabling team-authoring support. The **Disable Team Support for Current Project** option in the **Team Authoring** menu is usually just used when you want to temporarily disable team-authoring support to build a Help target. However, if you want to detach your working copy of the project from the repository and remove all team-authoring functions, as well as the **Team Authoring** menu and node and the team-

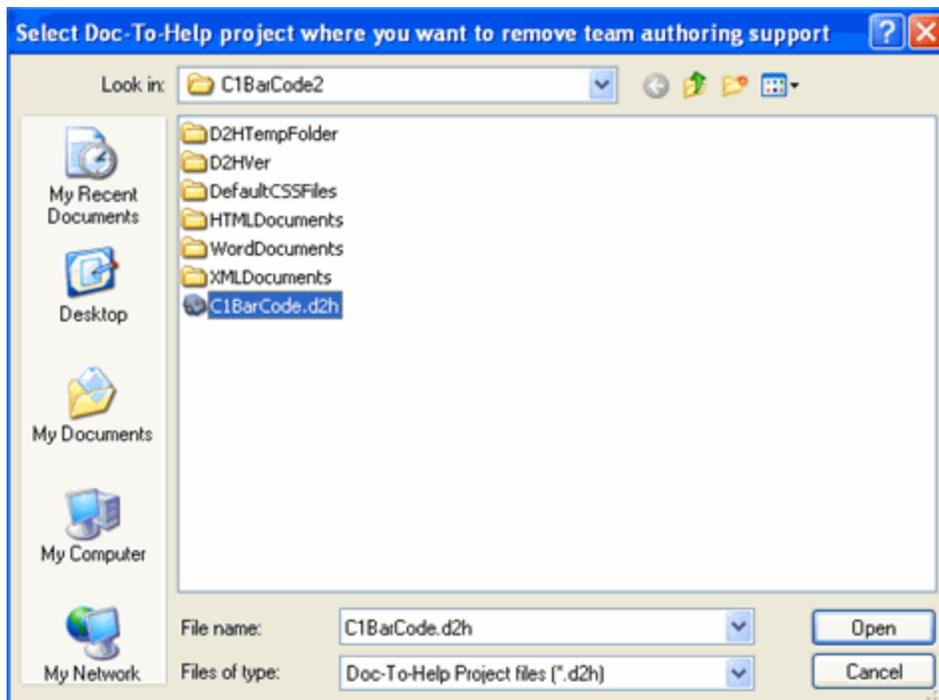
authoring buttons in the **Documents** toolbar, you can use the special administrative utility included with Doc-To-Help, **C1D2HTeamAdmin.exe**.

To remove team-authoring support from your working copy of the project:

1. Double-click **C1D2HTeamAdmin.exe**. This file is installed, by default, to C:\Program Files\ComponentOne\DocToHelp when Doc-To-Help is installed.
2. Select **Tools | Detach local project from repository (remove team authoring support)**.



3. Locate and select the local project you want to detach from the repository.



4. Click **Open**. A dialog box appears, confirming team-authoring support has been removed from your project.



5. Click **OK**. Your local project becomes a regular, single-author project without team-authoring support. This action only changes your local project; it does not change the repository team project or other authors' projects.

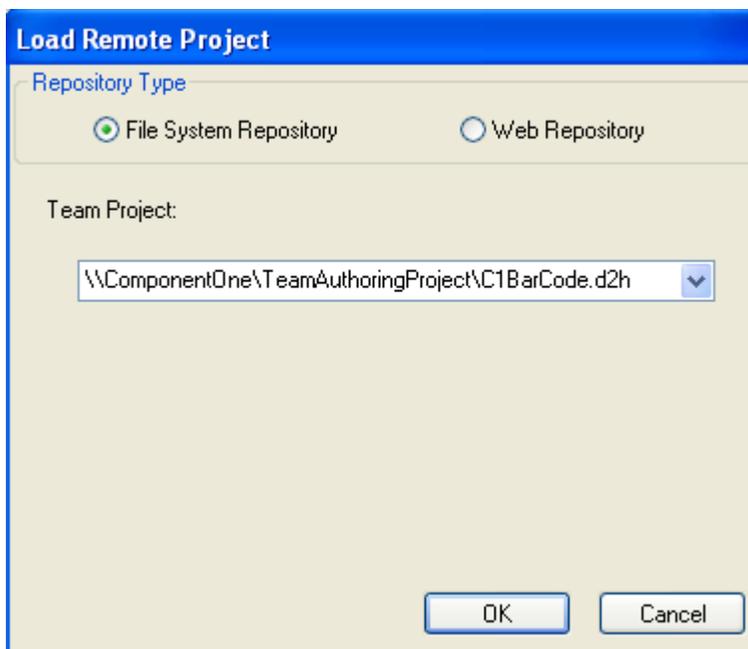
Changing Repository Settings

When a project is first set up for team authoring using the **Share Project** wizard, the author sharing the project determines if the number of backup files kept in the repository for each file is limited, and, if so, to how many. The author also determines whether other team members can delete old backup files when they check a newer version of the file into the repository and the limit of backups for that file has been exceeded.

These settings can be changed by the administrator at any time using the Doc-To-Help team authoring administrative utility.

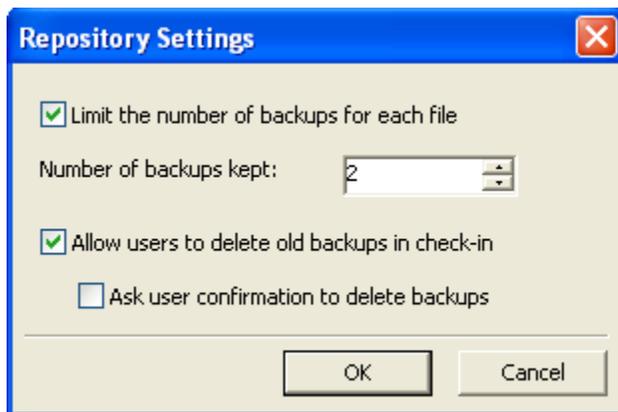
To change the repository settings:

1. Double-click **C1D2HTeamAdmin.exe**.
2. Select **File | Open Team Project** to connect to the repository. The **Load Remote Project** dialog box appears.



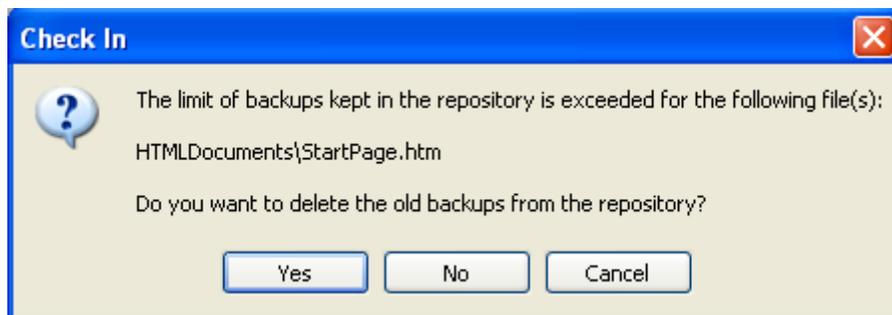
3. Select a repository type, either **File System Repository** or **Web Repository**, depending on the type of repository that was created for the team project.
4. Select a team project or choose **(Select Team Project)** from the drop-down box to browse for a project and click **OK**.
5. Select **Tools | Repository Settings**.

The **Repository Settings** dialog box appears.



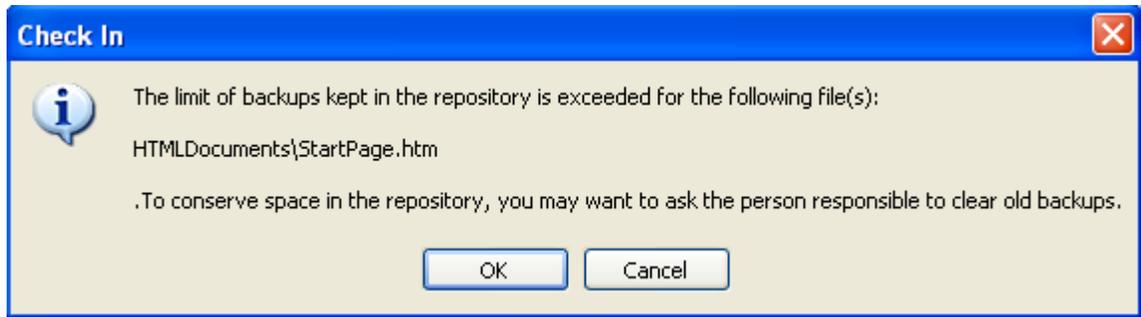
- **Limit the number of backups for each file:** Uncheck this box if you want to allow unlimited backups for each file. Every time a file is checked-in, a new backup file is created in the repository, and there is no automatic cleanup of old backup files. To limit the number of backups for each file, check this checkbox and specify the **Number of backups kept** in the repository.
- **Allow users to delete old backups in check-in:** Check this box if you want other team members to be able to delete old backup files when they check a newer version of the file into the repository when the limit of backups for that file has been exceeded.
- **Ask user confirmation to delete backups:** Check this box if you want a confirmation dialog box to appear when the limit of backups for a file has been exceeded when a team member checks a newer version of the file into the repository. This option is only available when the **Allow users to delete old backups in check-in** checkbox has been checked.

If the **Allow users to delete old backups in check-in** and **Ask user confirmation to delete backups** dialog boxes are both checked, team members will receive the following dialog box when checking a file into the repository once the backups limit has been exceeded.



The **Ask user confirmation to delete backups** checkbox is unchecked by default, meaning the team member will not be asked to confirm the backup cleanup; it is performed automatically without any notification. Therefore, the above dialog box will not appear when **Allow users to delete old backups in check-in** is checked and the **Ask user confirmation to delete backups** is not checked.

If the **Allow users to delete old backups in check-in** checkbox is unchecked, when a team member checks a file into the repository and the limit of backups for the file has been exceeded, the following dialog box appears. The member is not permitted to delete the old backup files and must contact the administrator to do so.



6. Click **OK** to close the **Repository Settings** dialog box.

Using Natural Search for Doc-To-Help



ComponentOne Natural Search provides a natural language search capability for WinHelp (.hlp), compiled HTML Help (.chm) and NETHelp Help (.htm) files. This feature is available in Doc-To-Help Enterprise 2006 only.

When Doc-To-Help 2006 Enterprise builds a help file, Natural Search automatically generates an index file (.smi). When the end-user types a question and initiates a search, the Natural Search runtime checks this index to generate a set of answers that match the question asked.

If Natural Search has been installed, Doc-To-Help 2006 Enterprise displays additional properties for the WinHelp, HTML Help, and NetHelp targets that enable and control the search features. By setting the [SearchEnabled](#) property to **True**, Natural Search automatically generates index and configuration files for the project in the same directory as the finished Help file.

Note: Although the NetHelp target exposes the [SearchEnabled](#) property, it does not incorporate Natural Search into the user interface of the generated Help system itself. This option is provided for importing HTML pages into the **ComponentOne Response** product.

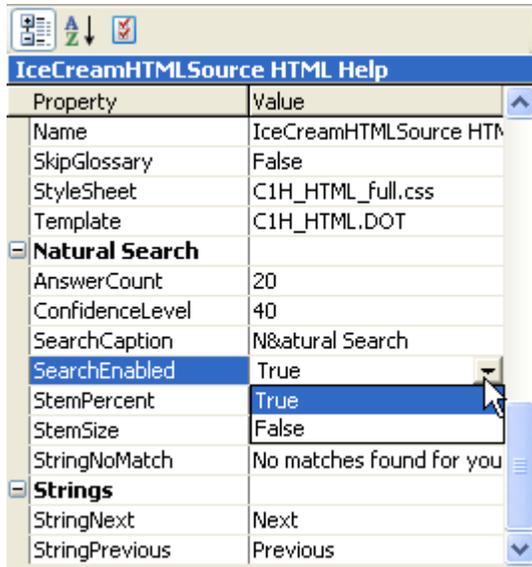
Natural Search Properties

AnswerCount	Determines the maximum number of answers to be returned when a natural language search is performed.
ConfidenceLevel	Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed.
SearchCaption	Specifies the text added to the tab caption (HTML Help) or button caption (WinHelp) when ComponentOne Natural Search is enabled.
SearchEnabled	Controls whether ComponentOne Natural Search is enabled for a Help target, document, or topic.
StemPercent	Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.
StemSize	Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.
StringNoMatch	Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Enabling Natural Search

To enable Natural Search for HMTL Help 1.x (.chm), NetHelp (.htm) and WinHelp 4.0 (.hlp) targets:

1. Select the desired Help target from the Project Editor.
2. In the property pane, set the [SearchEnabled](#) property to **True**.



Note: Make sure the [GenerateProjectFile](#) property is also set to **True** (HTML Help and WinHelp only). If you have the [GenerateProjectFile](#) property set to **False**, you will need to manually edit the .hlp or .hlp file to enable Natural Search. For more information, see [Manually Adding Natural Search Support](#) (page 557).

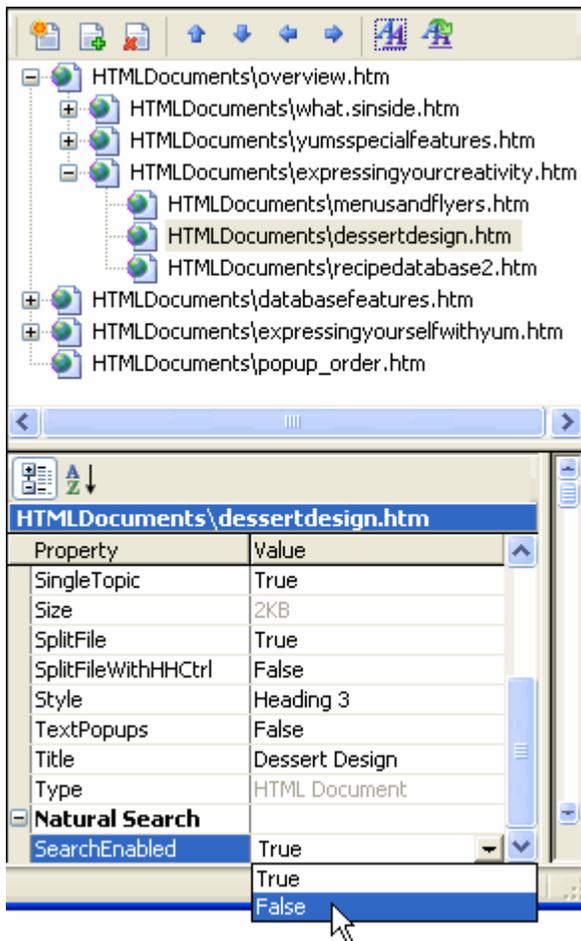
Defining Your Natural Search Content

In addition to enabling the Natural Search functions, the [SearchEnabled](#) property allows you to define which topics and documents you want included in the search index. By default, all documents and topics are included in the Natural Search index.

To exclude a document from the Natural Search index:

1. Select the desired Help target from the Project Editor.
2. In the Icon Bar, click **Project**.
3. From the left pane, select **Documents**.
4. Select the document you want to exclude from the right pane.

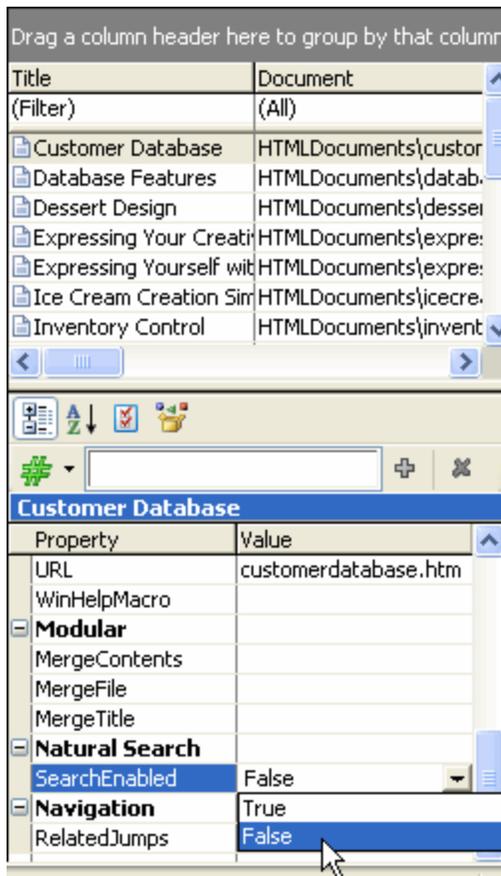
5. In the property pane, set the [SearchEnabled](#) property to **False**.



To exclude a topic from the Natural Search index:

1. Select the desired Help target from the Project Editor.
2. In the Icon Bar, click **Topics**.
3. From the right pane, select the topic you wish to exclude.

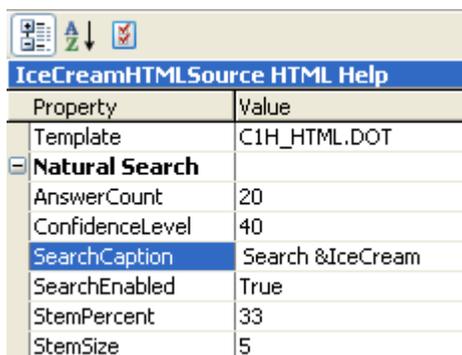
- In the property pane, set the [SearchEnabled](#) property to **False**.



Modifying the Natural Search Tab Caption

By default, the caption for the Natural Search tab is "N&atural Search". You can modify the caption by changing the [SearchCaption](#) property.

- Select the desired Help target from the Project Editor.
- In the property pane, select the [SearchCaption](#) property and insert the desired text into the right cell.



- Before building your Help target, make sure that the [SearchEnabled](#) property is set to **True** (HTML Help and Winhelp).

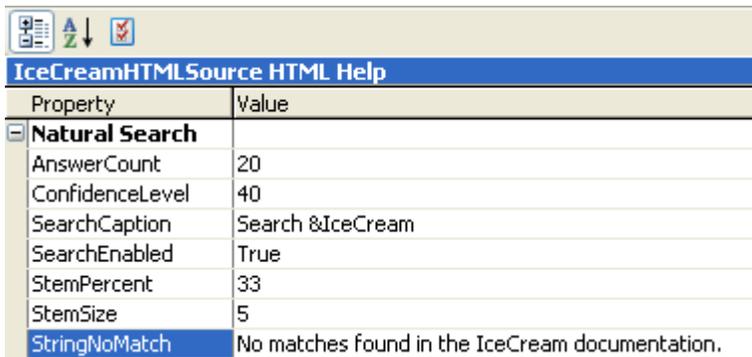
After building the Help target, the Natural Search tab displays your [SearchCaption](#) property text.

Note: Placing an ampersand in front of any letter in the caption defines that letter as an shortcut key. Your reader can then access the Natural Search tab by simply pressing Alt key plus the defined letter. (In the example above, it would be **Alt+I**.)

Defining the "No Match" Message

Whenever a user performs a search without a successful match, Natural Search delivers a No Match message. By default, the No Match is "**No matches found for your request.**". You can modify this message by changing the [StringNoMatch](#) property.

1. Select the desired Help target from the Project Editor.
2. In the property pane, select the [StringNoMatch](#) property and insert the desired text into the right cell.



Property	Value
Natural Search	
AnswerCount	20
ConfidenceLevel	40
SearchCaption	Search &IceCream
SearchEnabled	True
StemPercent	33
StemSize	5
StringNoMatch	No matches found in the IceCream documentation.

After building the Help target, the Natural Search tab displays your [StringNoMatch](#) property text when a search is performed with no matches.



Defining the Number of Natural Search Returns

Doc-To-Help Natural Search provides two properties to allow you to define the number of search returns that your user receives. The [AnswerCount](#) property defines the maximum number of answers that Natural Search is returned to the user. The [ConfidenceLevel](#) property defines lowest score (in terms of percentage) that is returned to the user.

Defining the Maximum Number of Returns

1. Select the desired Help target from the Project Editor.
2. In the property pane, set the [AnswerCount](#) property to whatever number you wish to define.

By default, this property is set to 20 returns.

Defining the Minimum Score for Returns

1. Select the desired Help target from the Project Editor.
2. In the property pane, set the [ConfidenceLevel](#) property to whatever percentage score you wish to define.

By default this property is set to 40%.

Note: Matching answers that have a lower score than the value of the [ConfidenceLevel](#) property will be discarded.

Modifying the Search Root Word

Doc-To-Help Natural Search provides two properties to allow you to modify the search root words in the index.

The [Stemsize](#) property defines the maximum number of characters used as a root word in the natural search index. The default value of this property is 5. Highly technical documentation, where longer search terms are common, may benefit from a longer stem size.

The [ConfidenceLevel](#) property defines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes. The default value for this property is 33. That is, 1 character is added to the root word for every 3 additional characters in the word. Some international languages, where lengthy suffixes are common, may benefit from a higher percentage.

To modify either of these properties:

1. Select the desired Help target from the Project Editor.
2. In the property pane, select the appropriate property and set it to the desired number.

Manually Adding Natural Search Support

The HHP file for HTML Help projects and the HPJ file for WinHelp projects can be manually edited to include Natural Search support if the **GenerateProjectFile** option is set to **False**.

Adding Support to a HTML Help 1.x Target

To manually modify a .HPP file to include Natural Search, you need to make the following modifications:

1. In the [OPTIONS] section, add the following line:

```
Custom tab="N&atural Search", SmarTEng.SmartSearchPane
```
2. In the [WINDOWS] section, modify the line for the main window (or \$global_main) by adding 0x8000 (hexidecimal) to the 10th item in the list and by appending "0,,0,[N&atural Search,SmarTEng.SmartSearchPane]" to the end of the line.

For example, if the line is

```
$global_main="VSA", "VSA. hhc", "VSA. hhk", , , , , , 0x2120, , 0x3006, [200, 100, 1050, 700], 0x0, 0x0, , , 0, ,
```

it should be changed to

```
$global_main="VSA", "VSA. hhc", "VSA. hhk", , , , , , 0xa120, , 0x3006, [200, 100, 1050, 700], 0x0, 0x0, , , 0, , 0, [N&atural Search, SmarTEng.SmartSearchPane]
```

Adding Support to a WinHelp Target

To manually modify a .HPJ file to include Natural Search, you need to make the following modifications:

Add the WinHelp macro for the **Natural Search** button using the following code:

```
[CONFIG]
RegisterRoutine(`SmarTEng.dl I', `SmarTLookupEx', `SSSU'): CreateButton(`smartBtn', `N&atural Search', `SmarTLookupEx(qchPath, `', `Natural Search', hwndApp)')
```

Altering the Natural Search Noise Word List

The noise word list is a simple text file containing all of the words you want to be ignored when a Help file is searched. Each individual noise word appears on a separate line in the file. The initial list is in alphabetical order but there is no requirement that they be added that way.

The default versions of the configuration and noise word list files are installed in the Natural Search default directory (C:\Program Files\ComponentOne\Natural Search\Default\smdefault.cfg and smdefault.snz). Open the smdefault.snz text file to add or remove specific noise words. If adding words, remember that each word must appear on a separate line.

Distributing Natural Search with your Help Files

There are four elements that need to be installed to enable Natural Search to work correctly with your Help file.

For each help target, Natural Search will generate three files: The **Index** (.smi), the **Noise Word List** (.snz), and the **Configuration** file (.cfg). These files will have the same name as the Help file, and should be installed in the same directory as the .chm or .hlp file.

In addition, the Natural Search Engine needs to be installed. Natural Search provides a setup.exe program, located in the "Redist" folder of the Natural Search installation directory (C:\Program Files\ComponentOne\Natural Search\Redist\). This installs the Natural Search runtime (SmaRTEng.dll) and configures the necessary system information.

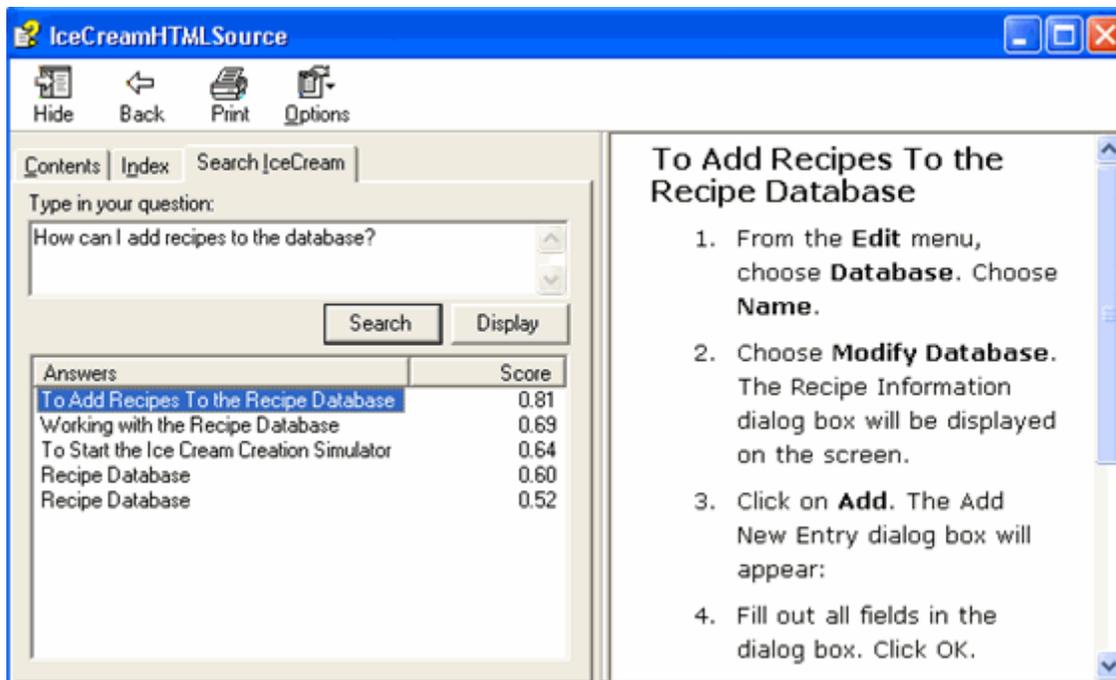
Note: The **Index** (.smi) files for WinHelp and HTML Help 1.x files are not interchangeable and should be installed with their appropriate help target.

Searching a Help File Using Natural Search

If the Natural Search [SearchEnabled](#) property was enabled for the Help target, the resulting WinHelp or HTML Help file will include the Natural Search interface for natural language search.

Searching an HTML Help File

Natural Search adds a tab to the navigation pane of the HTML Help window. Selecting this tab displays the search screen where the user can enter a question.



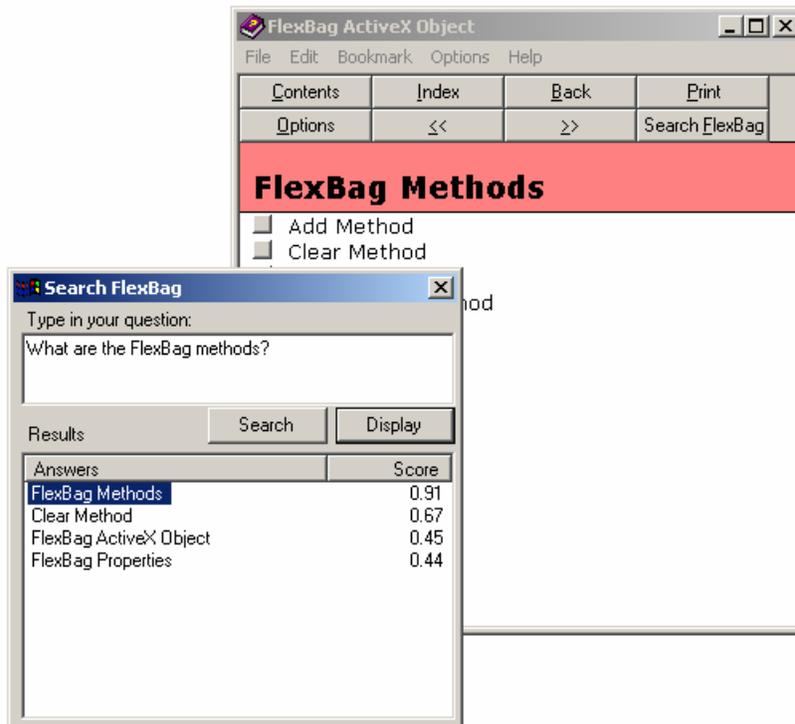
To Search an HTML Help file with Natural Search

1. Click the Natural Search tab in the navigation pane.
2. Enter a question in the text field at the top of the screen and press **Search**. All matching topics are displayed in the **Results** list.

3. Double-click a topic and click **Display** to navigate to that topic in the main Help window.

Searching a WinHelp Help File

Unlike the HTML Help file, the WinHelp interface does not provide additional tabs. Instead, **Natural Search** is added as a navigation button to the top of the help window. Clicking the **Natural Search** button opens a dialog box where the user can enter a question.

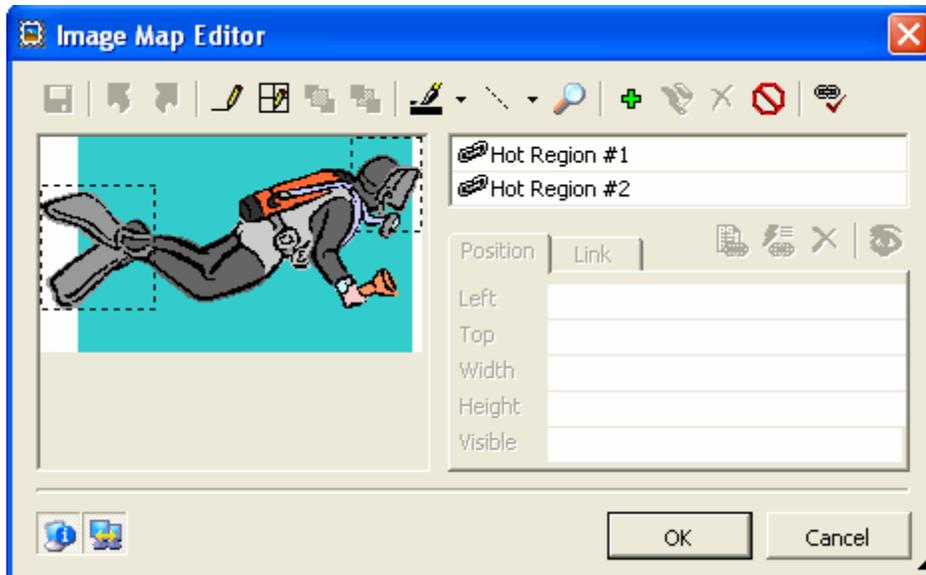


To Search a WinHelp file with Natural Search

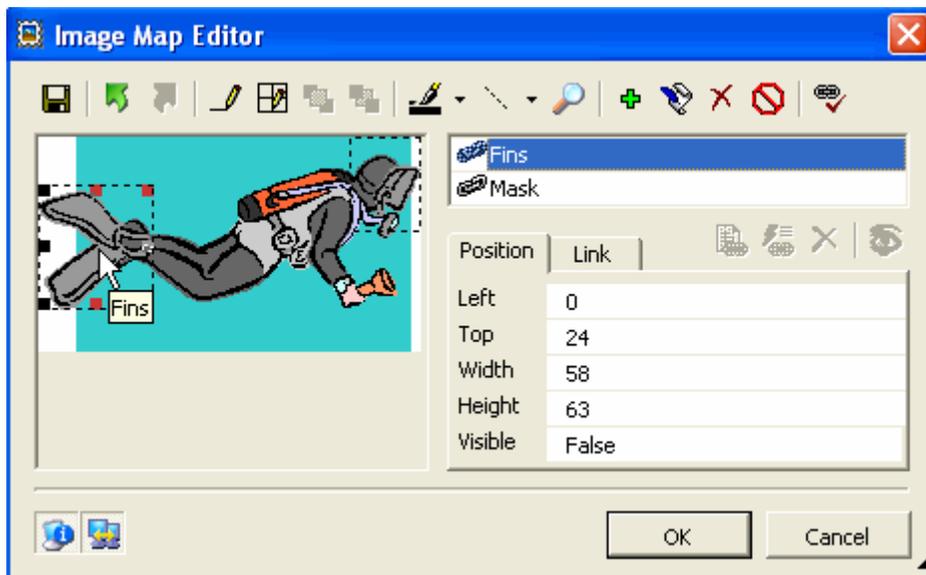
1. Click the **Natural Search** button in the WinHelp window.
A dialog box is displayed.
2. Enter the question in the text field at the top of the dialog box and press **Search**.
All matching topics are displayed in the **Results** list.
3. Double-click a topic and click **Display** to navigate to that topic in the main WinHelp window.

Using the Image Map Editor in Microsoft Word

The ComponentOne **Image Map Editor** allows you to place hot spot topic links within a graphic. By simply selecting a graphic and clicking the Image Map Editor button on the ComponentOne **Doc-To-Help** toolbar, you can define any number of hot spots to topics within your Help project.



In the online Help, the image above is defined as an image map. If you cursor over the diver's fins or mask, an author defined label opens, giving you information on the link.



If you click on the link, the defined topic replaces the current topic in the Help pane. You can also define the link so that it opens in a separate window.

Note: You can also create image maps within your HTML source documents. See [Using Image Maps](#) (page 329) for additional information.

Exploring the Image Map Toolbar Controls

	Save	Save defined topic links.
	Undo	Undo your last action.
	Redo	Redo your last action.
	Draw (Insert)	Defines the hot spot area.
	Split	Splits the graphic into a defined number of hot spot areas.
	Bring To Front	Brings a hot spot to the front.
	Send To Back	Sends a hot spot to the back.
	Line Color	Defines the line color of the hot spot area.
	Line Style	Defines the line style of the hot spot area.
	Zoom Mode	Zooms in on the image.
	New	Creates a new hot spot.
	Rename	Renames a hot spot.
	Delete	Deletes a hot spot.
	Clear	Clears all hot spots from an image.
	Verify Links	Verifies that the hot spot links are correct.

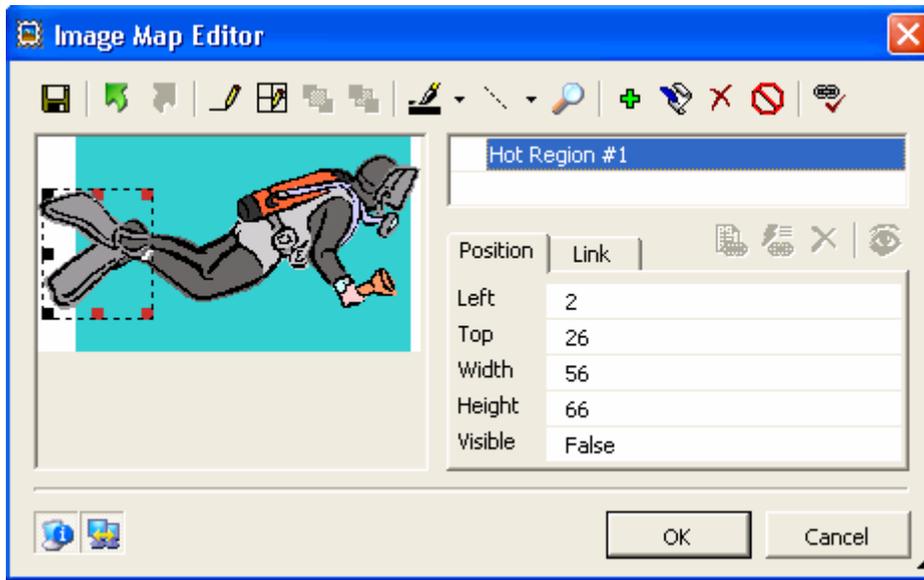
Exploring the Topic Link Toolbar Controls

	Topic Link	Assigns a topic link to a hot spot.
	Dynamic Link	Assigns a dynamic link to a hot spot.
	Delete Link	Deletes a link from a hot spot.
	View Topic	Views the topic that is linked to the hot spot.

Creating Image Hot Spots

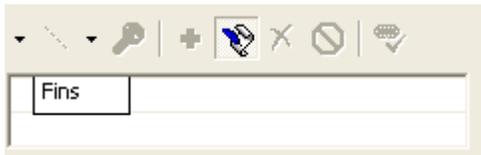
In many cases, it is more effective to create topic links using image hot spots rather than text links. In this way, the user can get information on a specific aspect of an interface or jump to a description of a specific tab by simply clicking on that area of the image in the Help file.

1. Select an image in your source document.
2. Click the **Image Map Editor** icon .
3. Click the **Draw (Insert)** icon .
4. Define the size of your hot spot by dragging the box over the image in the editor.

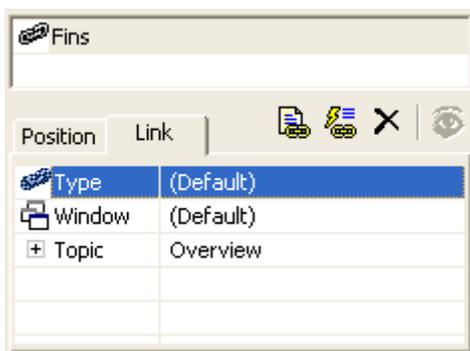


Note that the size properties (**Left**, **Top**, **Width** and **Height**) have been defined by the size of the box you created.

- Rename your link by clicking the **Rename** icon  and typing a new name. This name appears as a label when you cursor over the link area.



- Select the **Link** tab.
- Define your topic or dynamic topic by selecting the appropriate icon.



- Continue to insert additional hot spots as needed.
- Click **OK** to save your changes and close the **Image Map Editor**.
- Build your Help target.

Modifying the Size of an Image Hot Spot

There are two methods for modifying the size of an Image Hot Spot.

Dragging the Hot Spot Box

1. Select the image you want to modify and click the **Image Map Editor** icon.
2. Select the Hot Spot region box you wish to modify.
3. Stretch the box to the size you want.
4. Click **OK** to save the changes.

Using the Size and Position Properties

1. Select the image you want to modify and click the **Image Map Editor** icon.
2. Select the Hot Spot region box you wish to modify.
3. On the **Position** tab, modify the **Left**, **Top**, **Width** and **Height** properties.
4. Click **OK** to save the changes.

The Modular TOC Utility

One of the major problems with modular HTML Help is that the "child" modules do not contain the entire table of contents. Generally, this limitation is not an obstacle because the "parent" or hub Help file has the full table of contents and, in most cases, this is the Help file that is exposed to the end-user. Where the incomplete TOC can become an issue is during the implementation of "F1" or context sensitive Help. In this case, the child Help file may be called, leaving the end-user with no way to navigate through the entire Help contents.

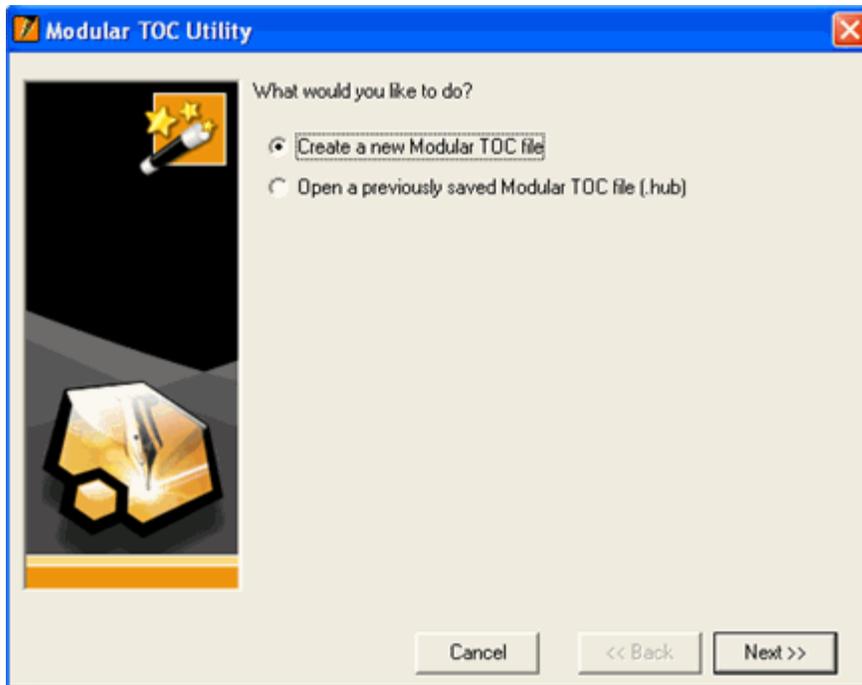
The ComponentOne **Modular TOC Utility** removes the table of contents limitation by automatically adding the full table of contents to each Help file in a modular project. The Modular TOC Utility is a stand-alone tool that can be used on any set of modular Help files. By simply pointing the utility at each of the component .HHP files a fully functional table of contents is incorporated into each modular Help file.

For more information on building a modular Help system, see [Using Modular Help](#) (page 427).

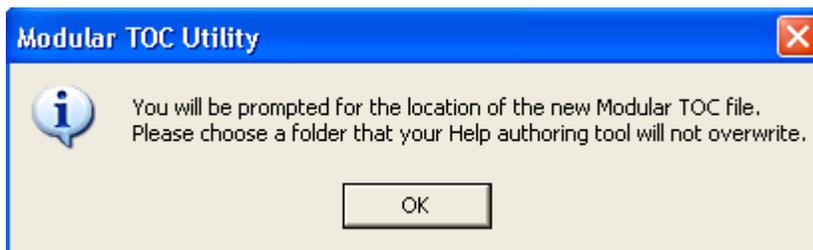
Using the Modular TOC Utility

Make sure that you have a fully defined modular Help system before continuing with the following steps.

1. Start the Modular TOC Utility by selecting Start | All Programs | ComponentOne Doc-To-Help | Modular TOC Utility. The Modular TOC Utility opens.



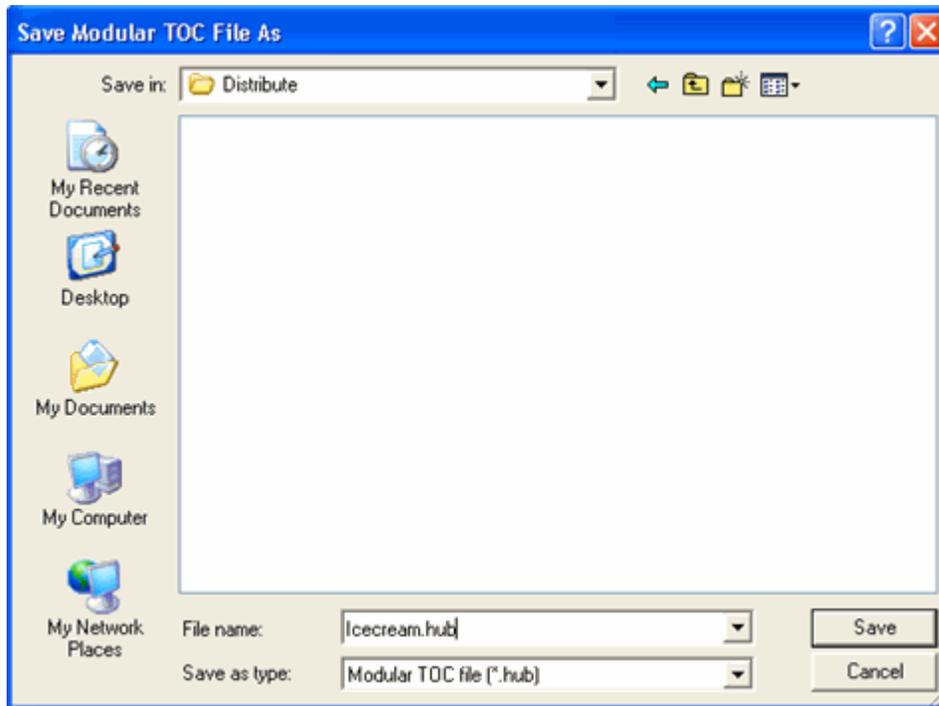
2. Select the **Create a new Modular TOC file** option and click **Next**.
3. After reading the informational dialog, click **OK**.



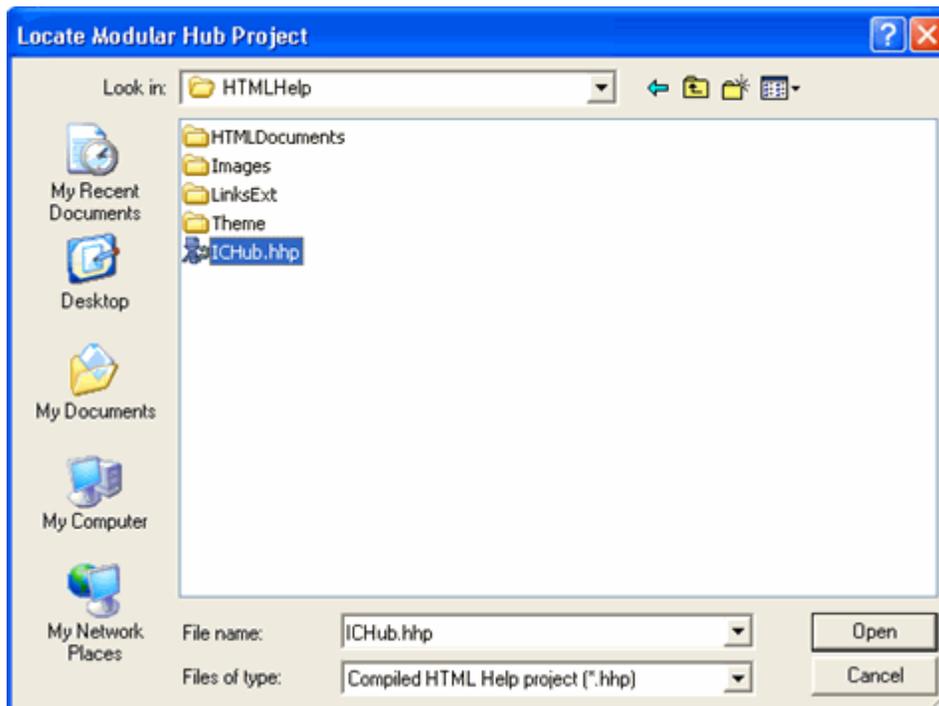
4. Browse to the directory where you want to store your .HUB file.

Note: The Modular TOC Utility will overwrite duplicate .HUB files.

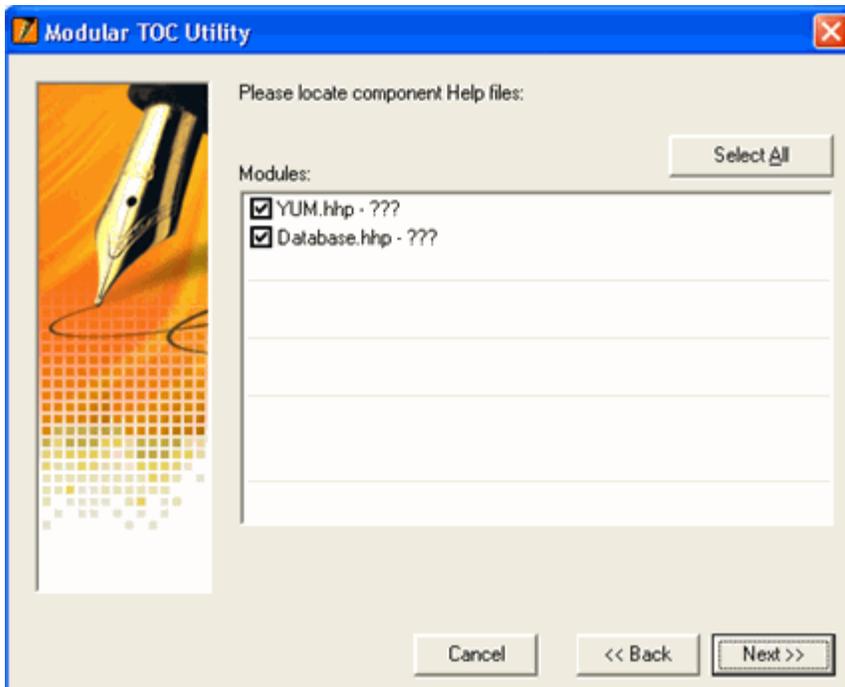
5. Provide the Modular TOC Utility with a .HUB file name and click **Save**.



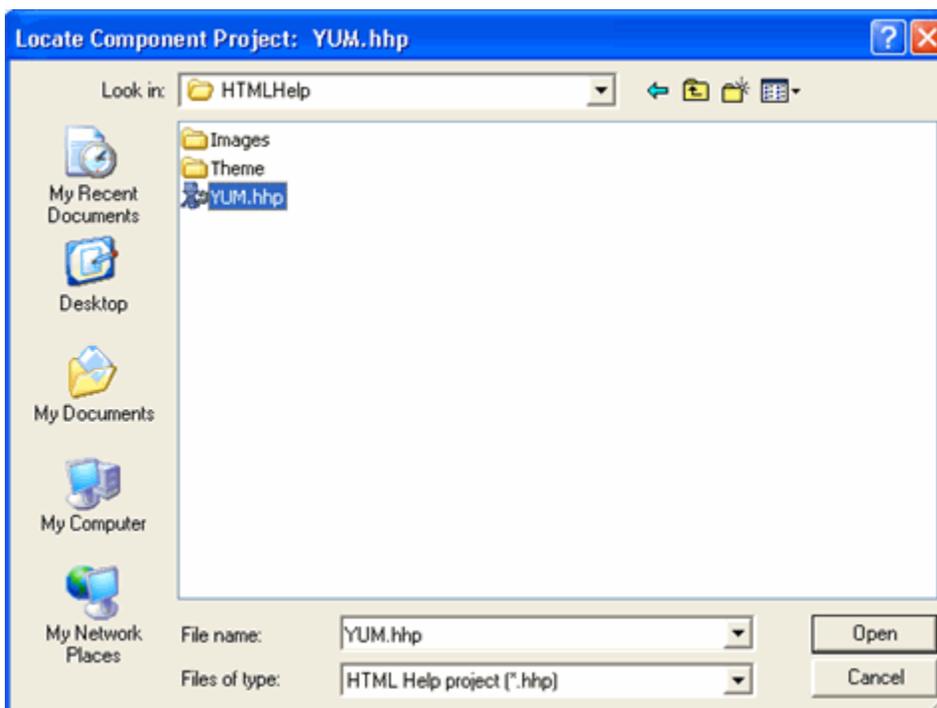
6. The Modular TOC Utility prompts you for the "hub" or parent .HHP file. Browse and select the file, then click **Open**. (This file is normally stored in the HTMLHelp sub-directory of your hub Help file.)



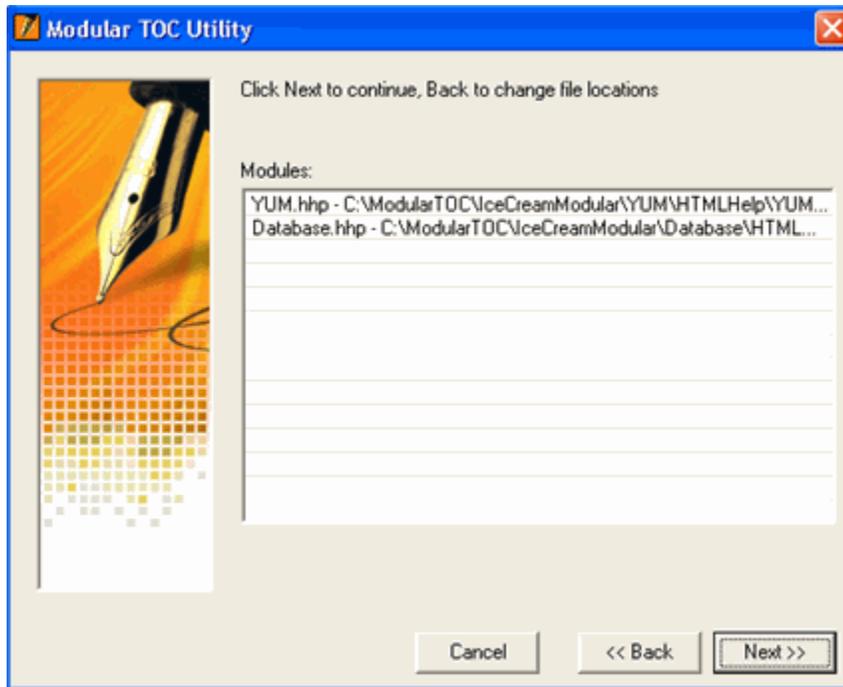
The Modular TOC Utility, lists all "child" or module .HHP files in your modular Help system.



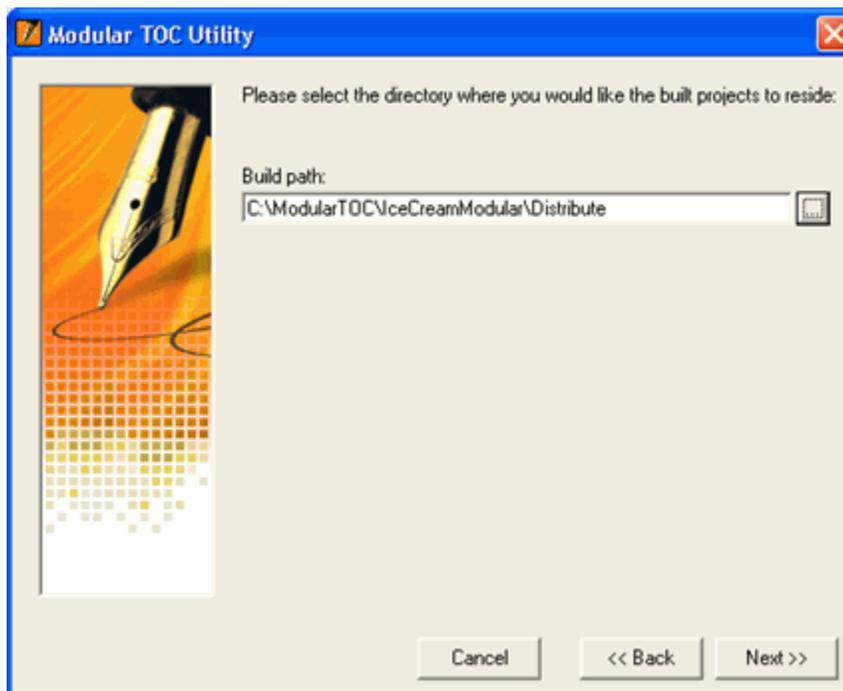
7. Click **Next**.
8. The Modular TOC Utility prompts you for the first "child" or module .HHP file. Browse and select the file, then click **Open**. (This file is normally stored in the HTMLHelp sub-directory of your module Help file.)



9. Repeat step 8 until each of the module .HHP files has been located.
The Modular TOC Utility, prompts you to verify that the paths to each .HHP file are correct.

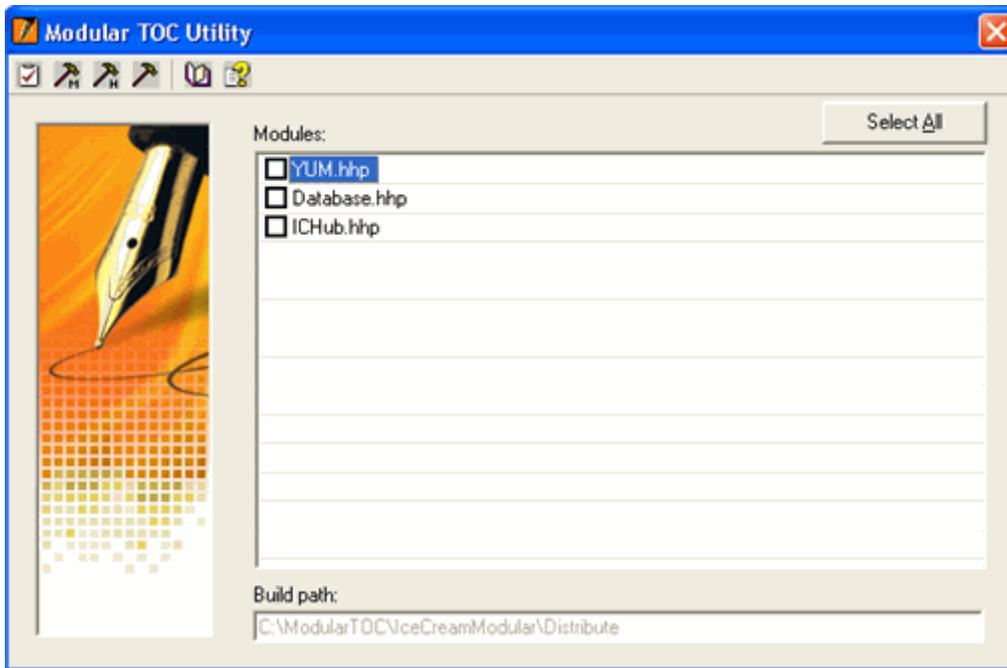


10. Click **Next**.
11. Browse to select a directory to store your new modular Help files.



12. Click **Next**.
13. Verify the .HUB file and build paths.
14. Click **Finish** to build the new modular Help files.

A number of black text boxes appear as the **Modular TOC Utility** builds the Help files. After the files are built, the **Modular TOC Utility Editor** opens.

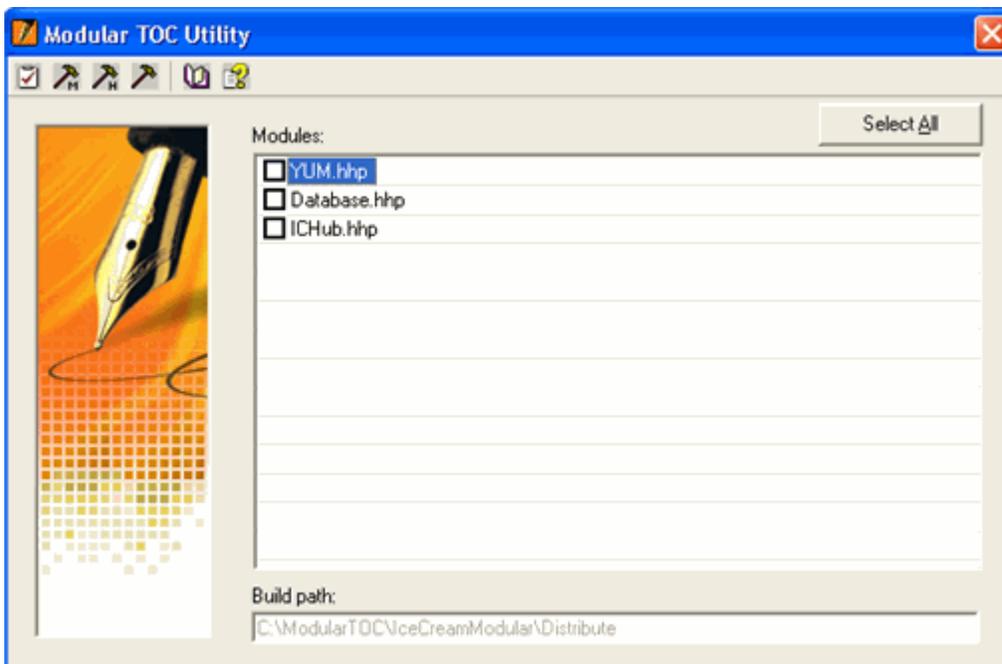


15. For now, close the **Modular TOC Utility Editor** and examine the Help files in your build directory. Note that all Help files, hub and module, include the entire table of contents.

For more information on using the **Modular TOC Utility Editor**, see [Exploring the Modular TOC Utility Editor](#) (page 568).

Exploring the Modular TOC Utility Editor

You can access the **Modular TOC Editor** immediately after building new modular Help files or by locating a previously built .HUB file. The editor allows you to compile or view any of the Help files in your modular Help system (by selecting the appropriate checkbox) or view the modular hub file.



Using the Context String Editor

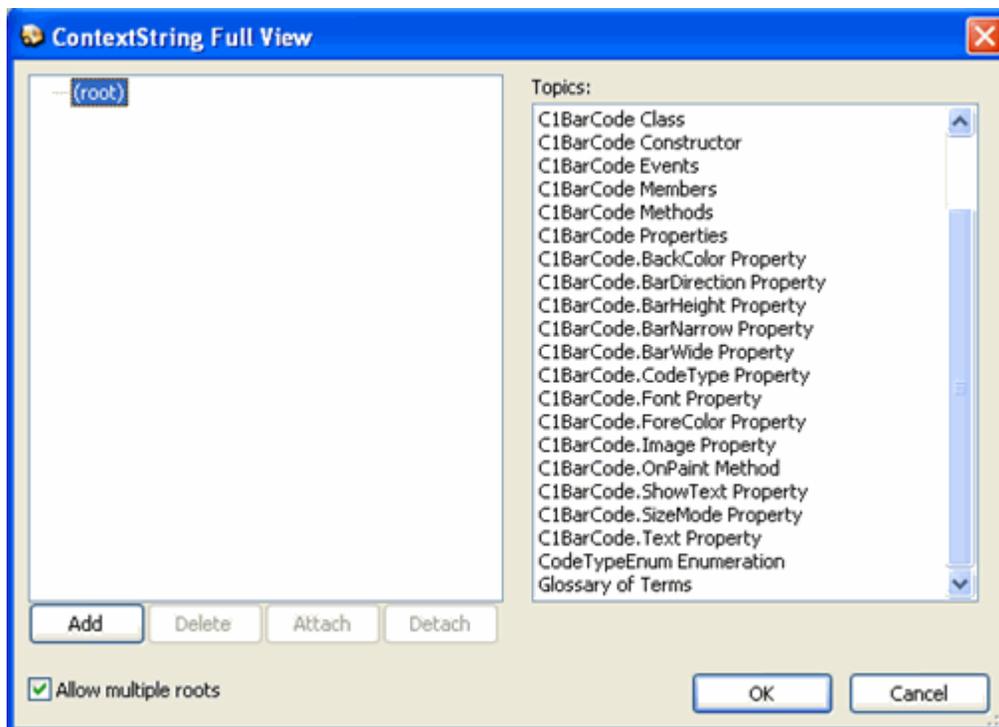
The purpose of the ComponentOne **Context String Editor** is to facilitate defining Microsoft Help 2.0 context strings and linking them to topics. For more information, see [MS Help 2.0 Target](#) (page 39).

In Microsoft Help 2.0, unique context strings are used to construct F1 Help. These hierarchical strings replace context ID numbers. An example of a string might be, **C1.C1Zip. C1ZipEntry.FileName**. This string is used to identify the topic that relates to specific .NET property.

Adding Context Strings

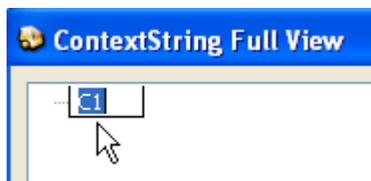
The **Context String Editor** allows you add context strings by defining each section of the hierarchical structure.

1. To open the **Context String Editor**, select **Topics** from the Icon Bar.
2. Select any topic from the Right Pane.
3. From the Property Pane, select the cell next to the [ContextString](#) property and click the arrow.

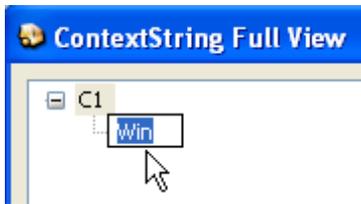


The **Context String Editor** opens. Note that the context strings are entered in the left pane while the topics in your Help project are displayed in the right pane.

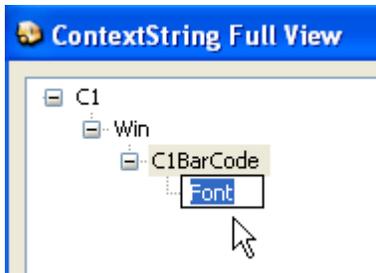
4. In the **(root)**, type in the first part of your context string and press **Enter**.



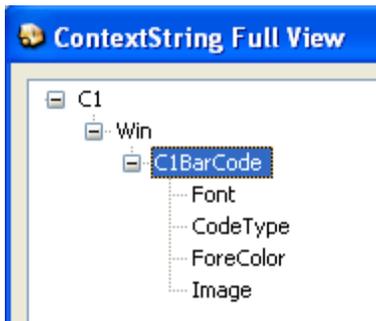
5. Click **Add**.
6. Enter the next section of your context string and press **Enter**.



7. Click **Add**.
8. Enter the third section of your context string and press **Enter**.
9. Continue to do this until you have added the last section of the context string.



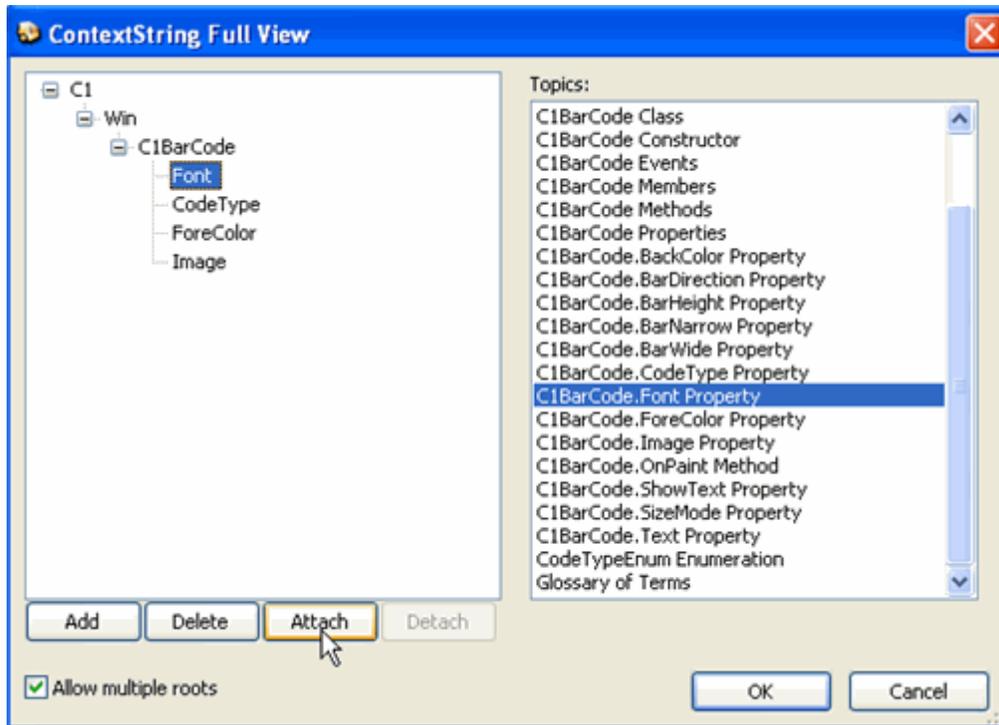
10. To add another context string, select the preceding section and click **Add**.
11. Enter the last section of the next context string and press **Enter**.
12. Continue this until you have entered all the context strings.



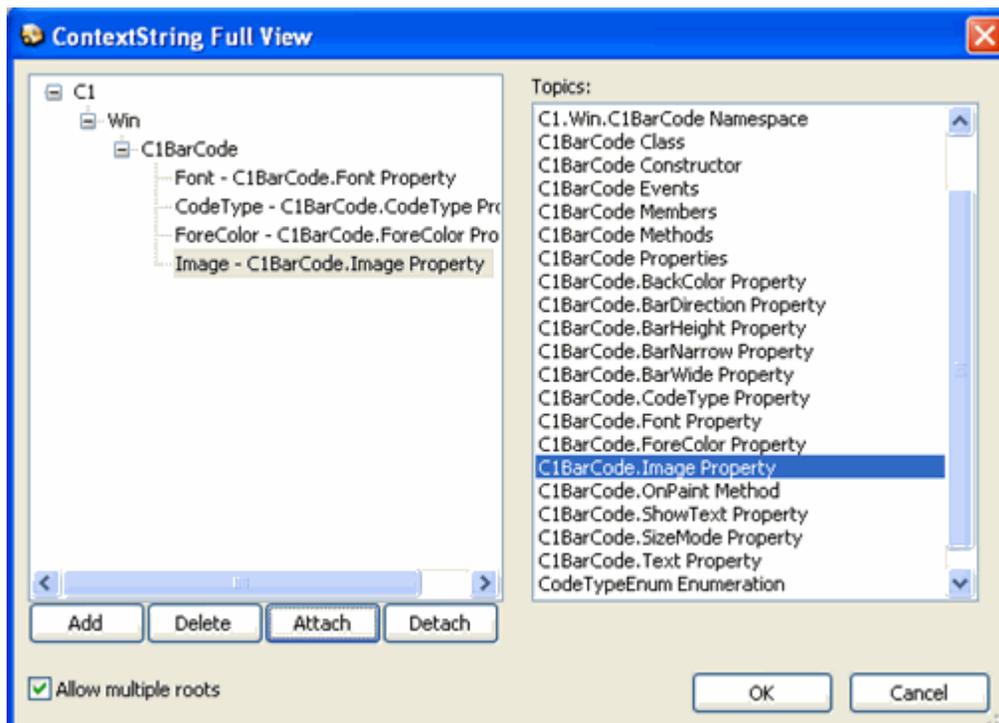
Linking Topics to Context Strings

To link a topic to a context string:

1. Select one of your context strings.
2. Select the corresponding topic from the right pane.
3. Click **Attach**.



- Continue to do this until you have finished attaching all topics to context strings.

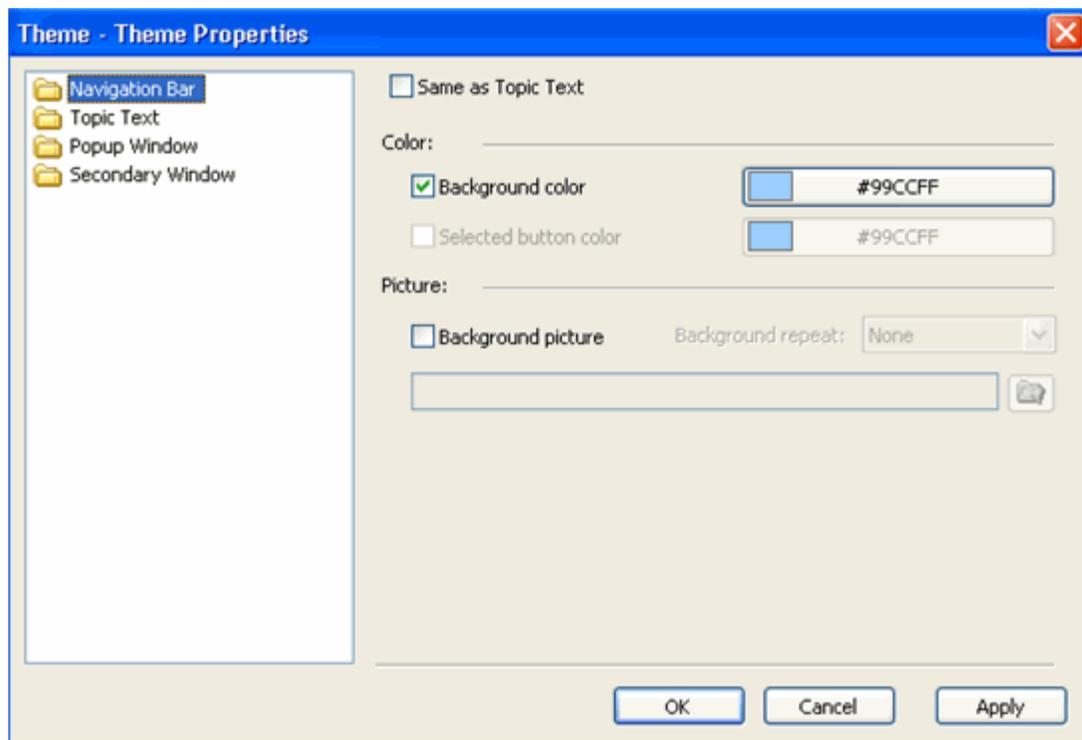


- Click **OK** when you are finished.

Using the Theme Designer

Since the original implementation of Doc-To-Help themes, authors have needed a method by which they could modify themes to fit their needs. The ComponentOne **Theme Designer** provides you with the means to develop and save customized themes. The theme designer gives you control over navigation, background styles and image insertion for the any defined window, secondary window or pop-up. These themes can be used for HTML Help, NetHelp, JavaHelp and Help 2.0 targets.

The **Theme Designer** allows you to modify the Navigation Bar, Topic Text, Pop-up windows, Secondary windows and more by simply clicking on the item in the left pane.

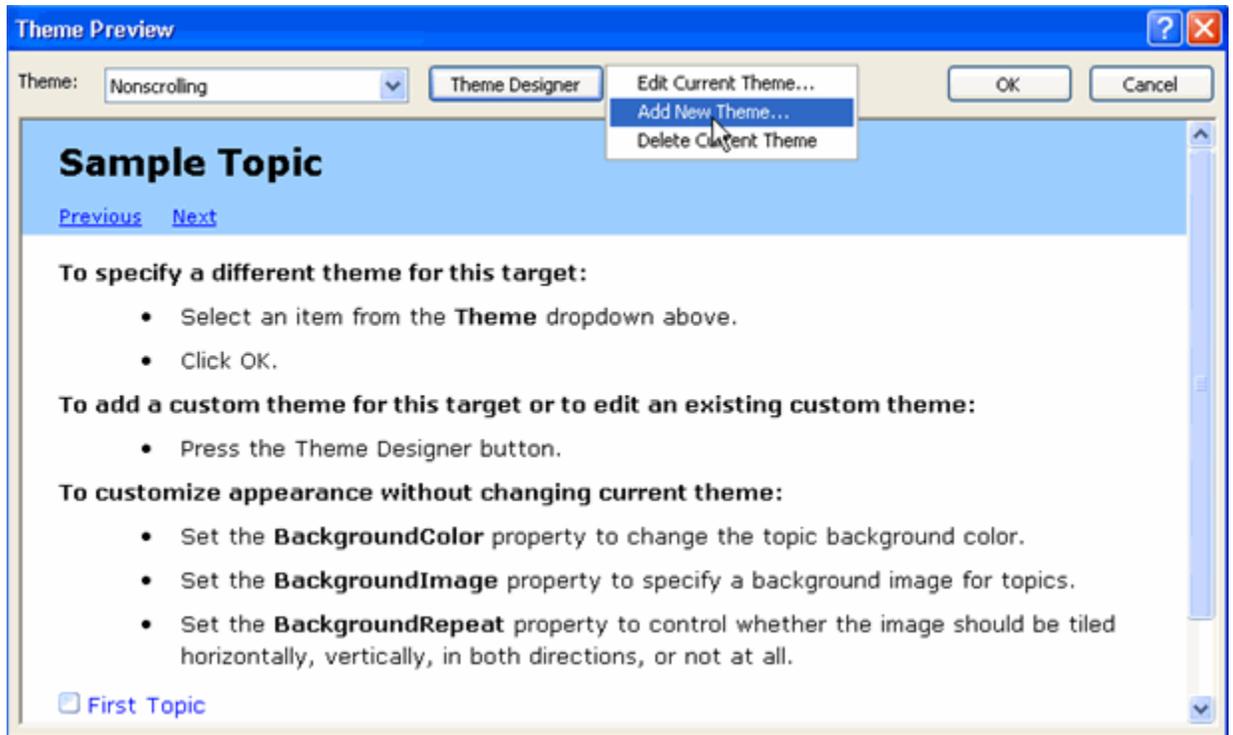


For more information on using themes, see [Using Themes to Modify the Help Window](#) (page 357).

Creating a New Theme

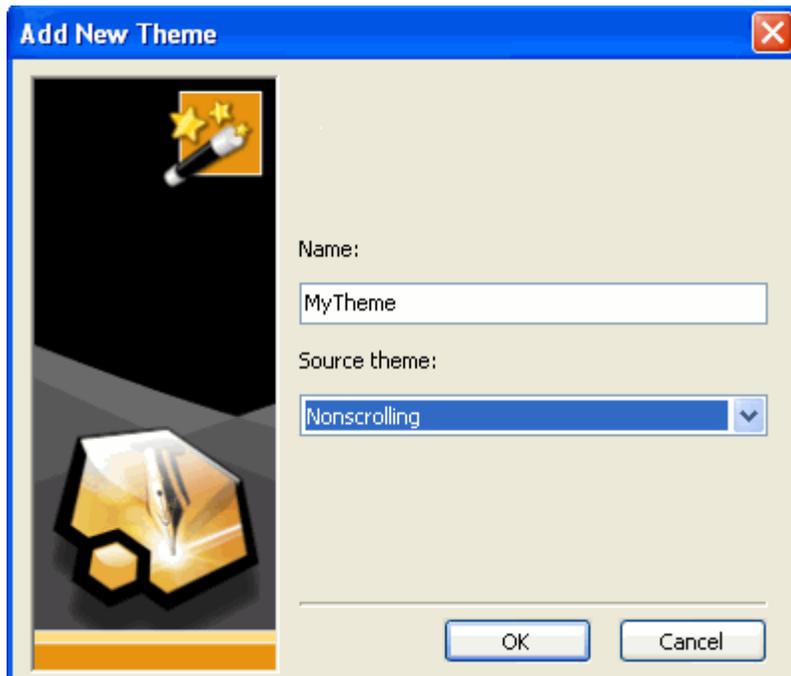
Doc-To-Help comes with a number of pre-formatted themes for HTML Help and HTML. These pre-formatted themes cannot be modified. You can develop your own themes based upon the pre-formatted themes by following the steps below.

1. Select the Help Target that you wish to modify.
2. From the property pane, select the [Theme](#) property and click the button in the box to the right of the property name.
3. Click the **Theme Designer** button and select **Add New Theme** from the menu.



The **Add New Theme** dialog opens.

4. Enter the name of the new theme in the **Name** textbox.
5. Select the pre-formatted theme you want to use as your source from the **Source Theme** drop-down.
6. Click **OK**.



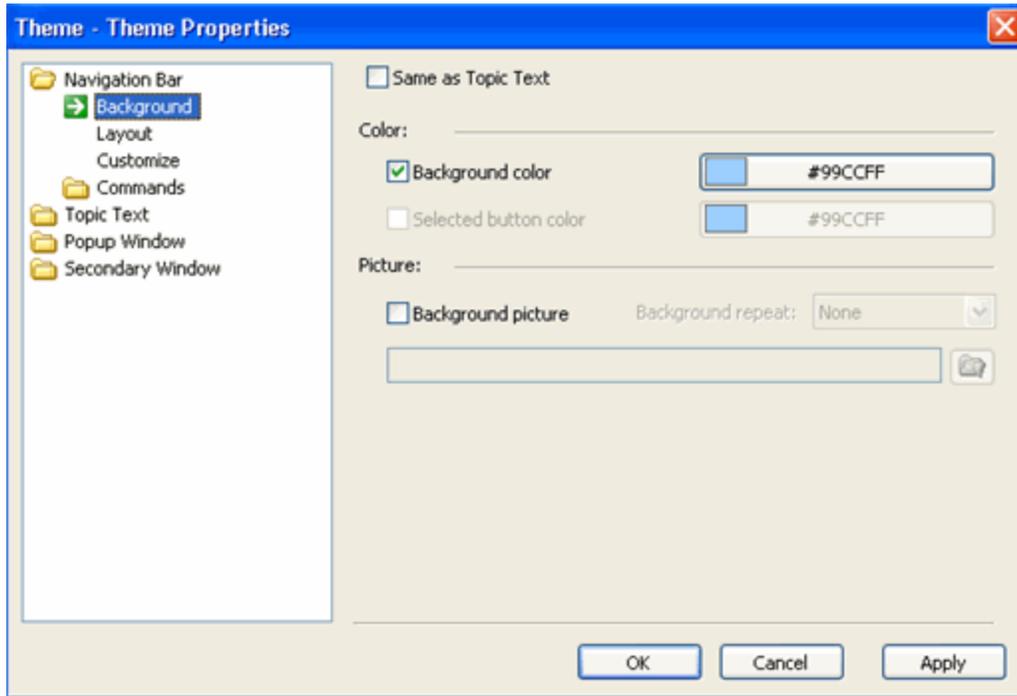
Doc-To-Help creates your new theme, and opens the theme editor.

Exploring the Navigation Bar Node

The Navigation bar node includes sections that allow you to define the background, layout and next and previous commands.

Note: The Navigation bar node is not available for NetHelp themes.

Background Controls



Same as Topic Text

Sets the theme properties so they are inherited from the project settings.

Background color

Defines the color in the navigation bar area. Click the button to the right to access color pallets.

Selected button color

Defines the color of a button once it has been clicked.

Background picture

Defines the background image in the navigation bar area. Click the browse button



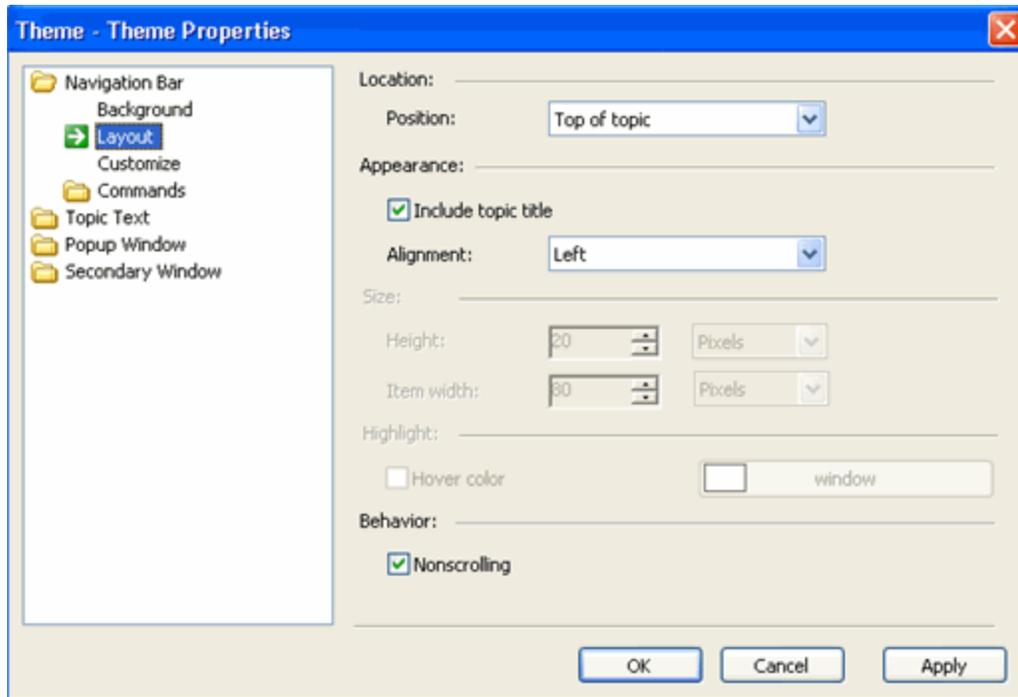
to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:

ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat

Defines where the image will appear in the background.

The Layout Controls



Position	Defines the location of the navigation area.
Include topic title	Determines if the topic title is included in the navigation area.
Alignment	Defines the position of the navigation elements.
Size	Defines the height and width of the navigation area. Note: The size attributes are only available in NetHelp targets.
Hover color	Defines the hover color for the navigation links. Click the button to the right to access color palettes.
Nonscrolling	Determines if the navigational area is nonscrolling.

The Commands Controls

The **Previous** and **Next** controls are available for HTML Help, JavaHelp and Help 2.0 targets. The **Previous** and **Next** controls are also available for the NetHelp target, in addition to a variety of other controls. See [Commands Node \(NetHelp Only\)](#) (page 584) for a full description. All of these controls have the following properties:

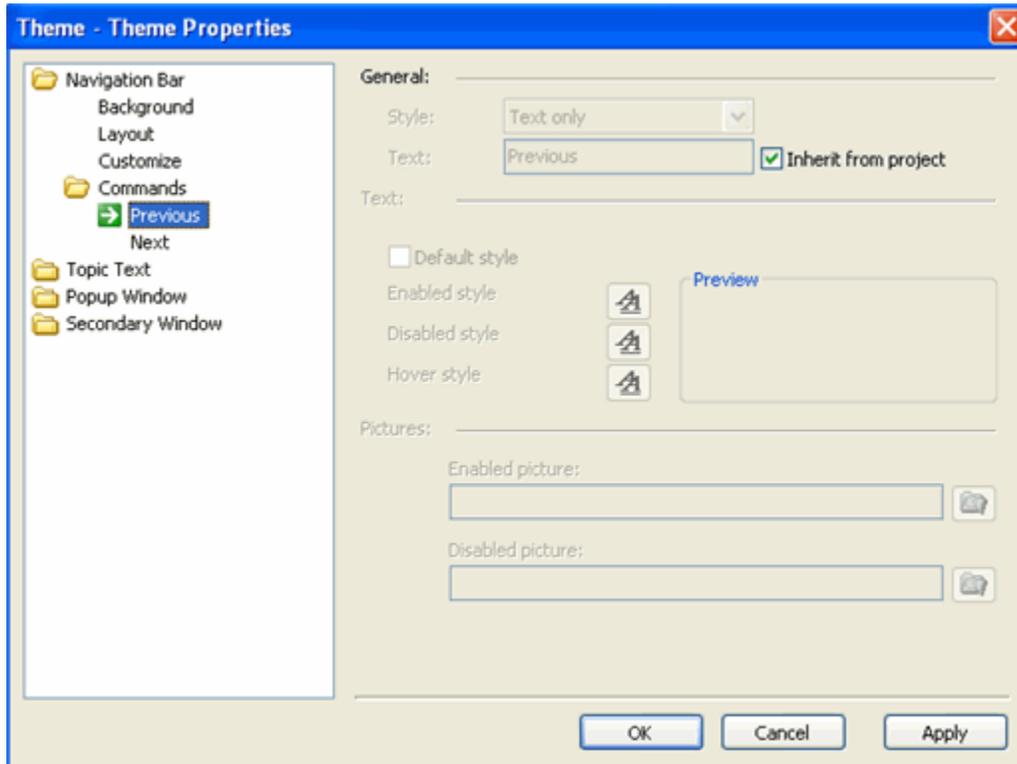
Style	Defines the Previous navigation as text, image or text and image.
Text	Defines text for the Previous link.
Inherit from project	Determines if the Previous link text is inherited from the project.
Default style	Determines if the Previous link style is inherited from the project.
Enabled style	Defines the enabled text style for the link.
Disabled style	Defines the disabled text style for the link.
Hover style	Defines the hover text style for the link.

Enabled picture

Defines the enabled image for the button. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Disabled picture

Defines the disabled image for the button. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

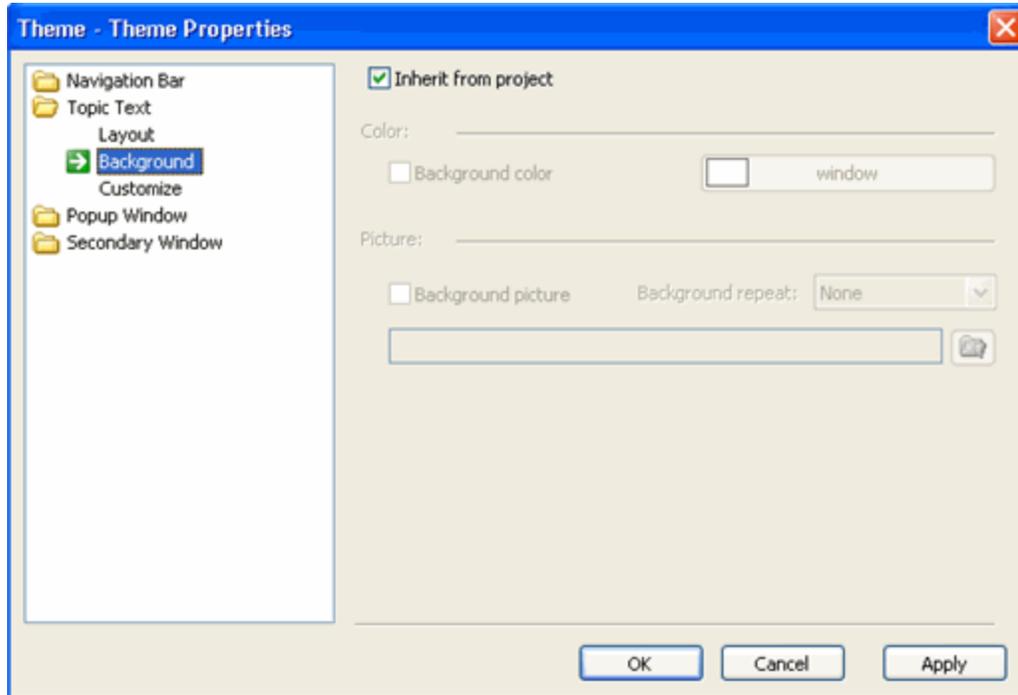
**Exploring the Topic Text Node**

The Topic Text node allows you to modify the formatting of the contents pane of your Help target.

Layout Controls**Add margins to the HTML source**

Adds a small padding, or margins, to the theme by default. If you have defined margins in your HTML source document and do not want Doc-To-Help to change them, uncheck this box. This check box applies to HTML source documents only.

Background Controls



Inherit from project

Sets the theme properties so they are inherited from the project settings.

Background color

Defines the color in the contents area. Click the button to the right to access color palettes.

Background picture

Defines the background image in the contents area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

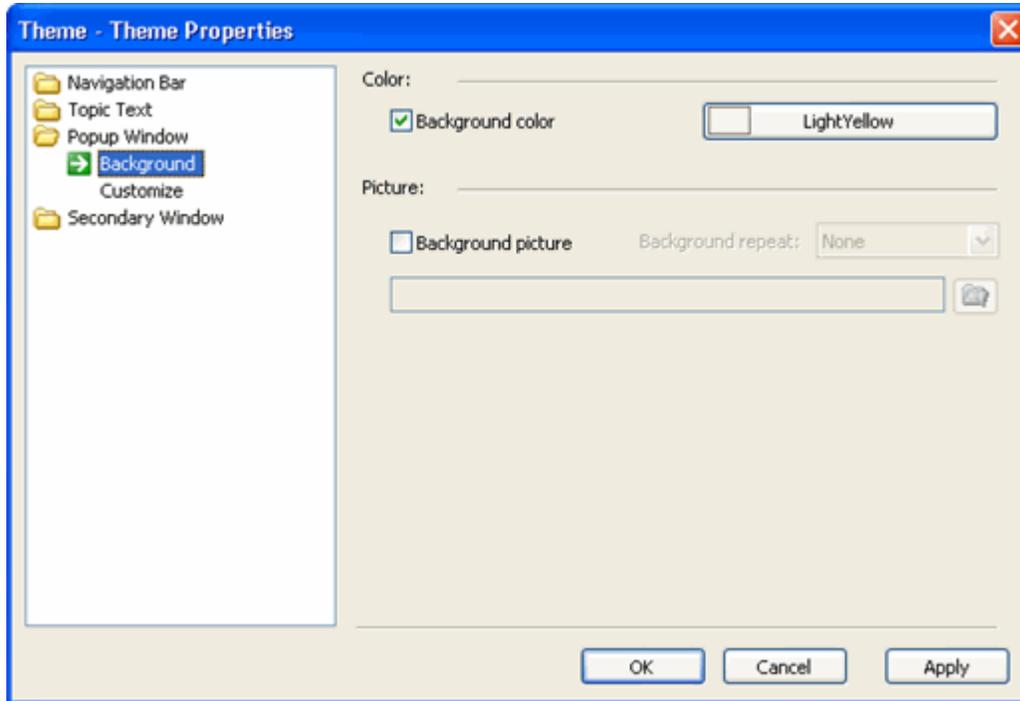
Background repeat

Defines where the image will appear in the background.

Exploring the Popup Window Node

The Popup Window node allows you to modify the formatting of the pop-up windows of your Help target.

Background Controls



Background color

Defines the color in the pop-up. Click the button to the right to access color palettes.

Background picture

Defines the background image in the pop-up. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

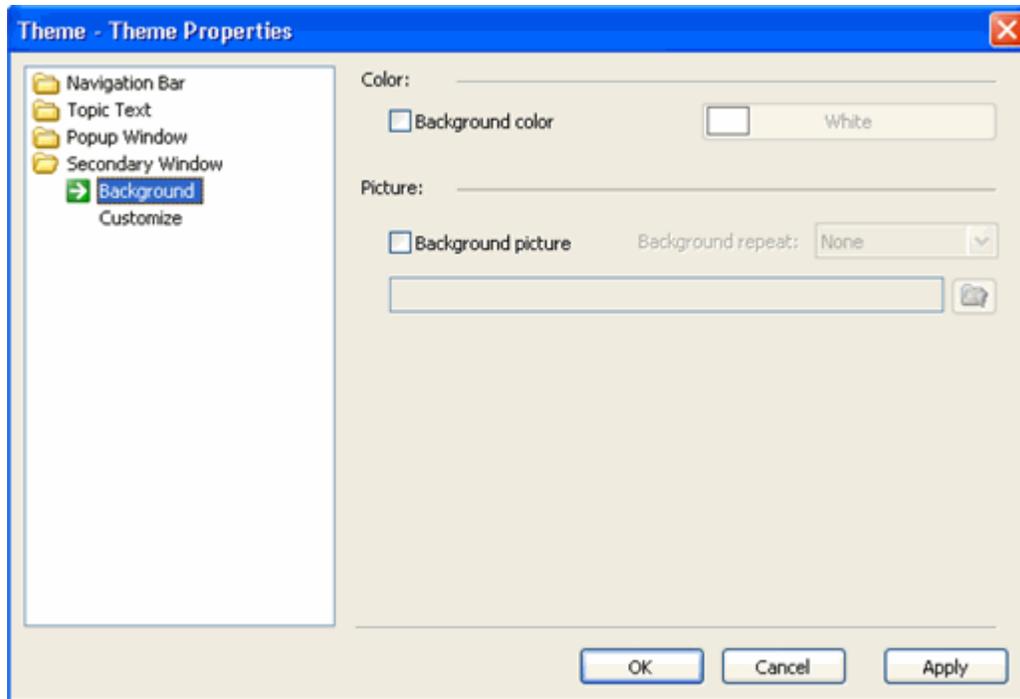
Background repeat

Defines where the image will appear in the background.

Exploring the Secondary Window Node

The Secondary Window node allows you to modify the formatting of the secondary windows of your Help target.

Background Controls



Background color

Defines the color in the secondary windows. Click the button to the right to access color palettes.

Background picture

Defines the background image in the secondary windows. Click the browse button



to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

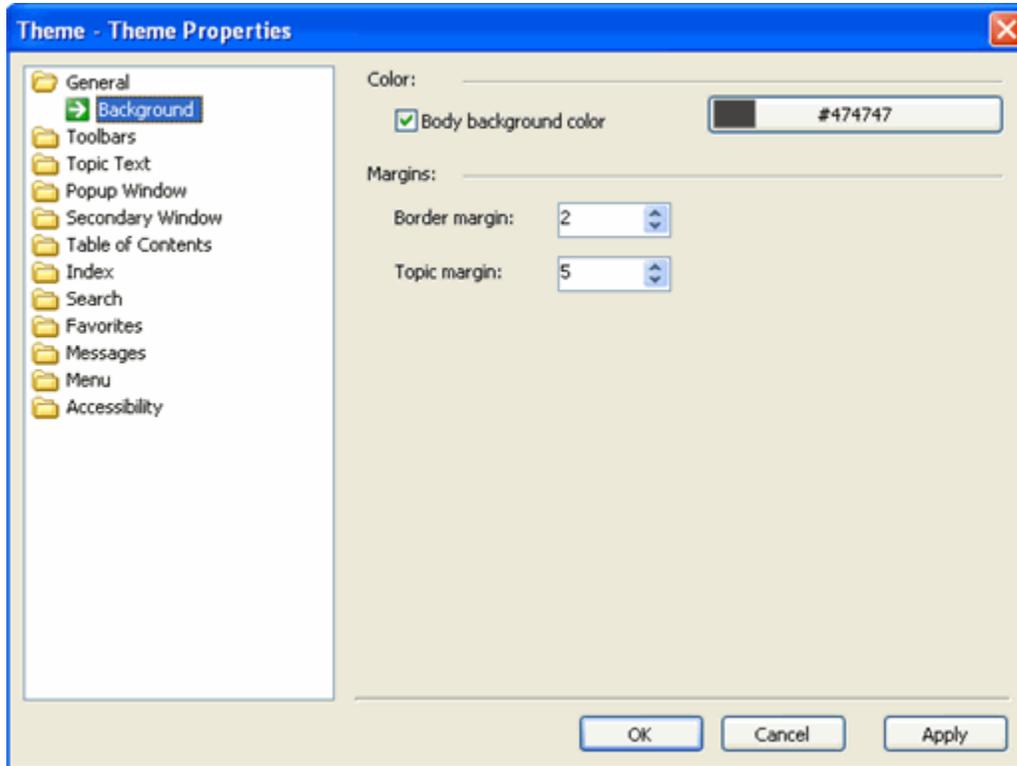
Background repeat

Defines where the image will appear in the background.

Exploring the General Node (NetHelp Only)

The General node includes sections that allow you to define the background of the NetHelp window.

Background Controls



Body background color

Defines the color of the <BODY> tag. The specified color appears in places not covered by the NetHelp panes, which include toolbars and the navigation and topic panes, such as margins. Click the button to the right to access color pallets.

Border margin

If non-zero, specifies the margins, or borders, of every pane, including toolbars and the navigation and topic panes, that can be colored by the **Body background color**.

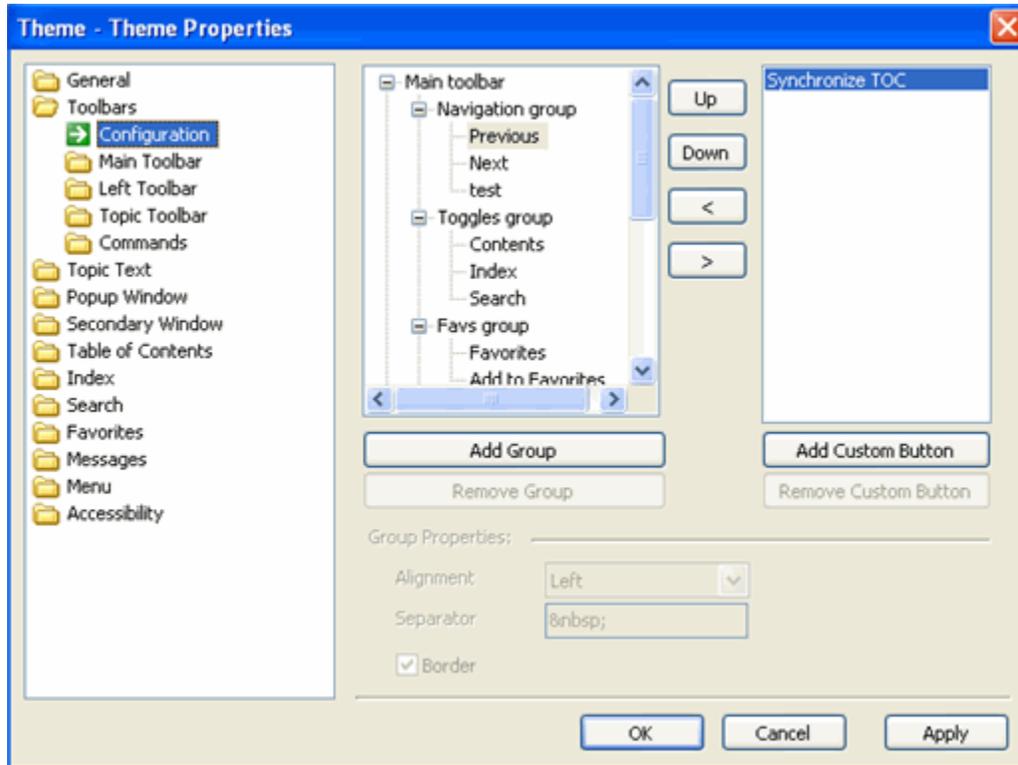
Topic margin

If non-zero, specifies an additional margin added to the topic pane, colored with **Body background color**. The margin of the topic pane is the sum of the **Border margin** and the **Topic margin**.

Exploring the Toolbars Node (NetHelp Only)

The Toolbars node includes sections that allow you to configure and customize the toolbars, define button groups in them, assign buttons to toolbars and button groups, add new custom commands, and set button group properties.

Exploring the Configuration Node (NetHelp Only)



Add Group

Click this button to add a button group, or a list of buttons that will appear together, under the selected toolbar. There may be multiple buttons or only one button in a button group. A group can also be empty, but this would be done on rare occasions to define a separator preceding the first group in the toolbar. If there are no groups and you want to add a button, create a group by clicking this button and then you can add a button.

Remove Group

Click this button to remove the selected button group.

Up/Down

Click these buttons to move a button group up or down in the list of button groups under the toolbar, or use these buttons to move a button up or down in the list of buttons under the group. Note that you can also use the drag-and-drop method to move button groups within or between toolbars or to move buttons within or between button groups.

Add Custom Button

Click this button to add a new button to the button list.

Remove Custom Button

Click this button to remove a button from the button list.



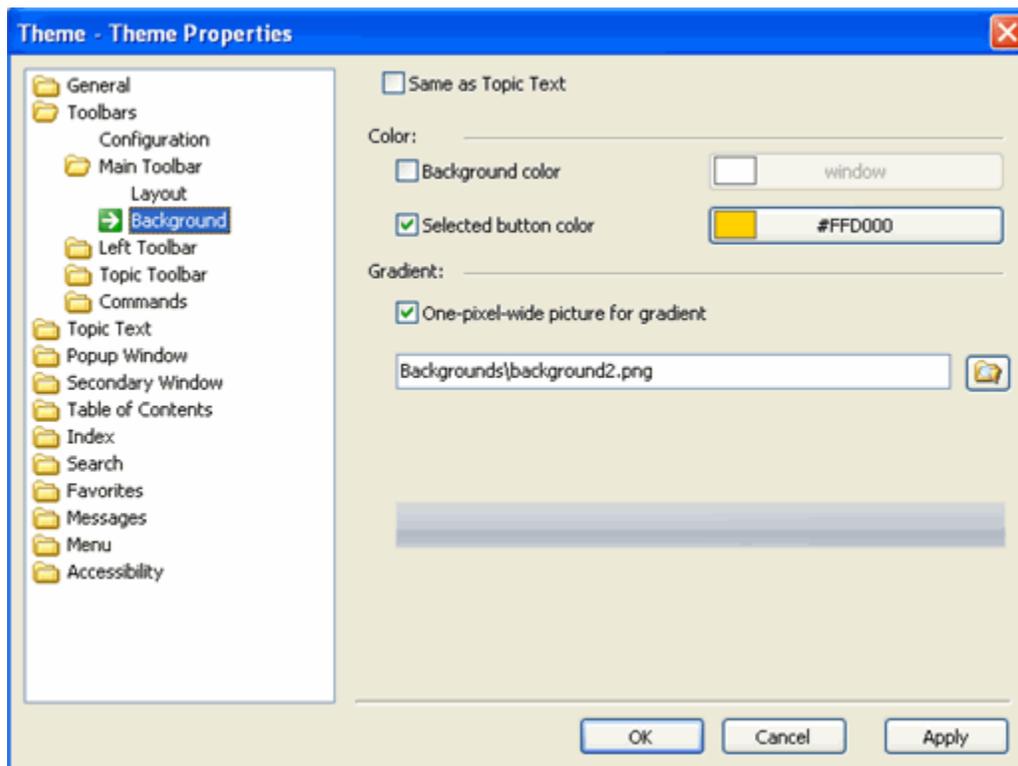
Click these buttons to move a button to the selected group or remove a button from the selected group and place it back in the button list. Note that you can also use the drag-and-drop method to move a button to and from the button list.

Alignment

Aligns the button group to the **Left**, **Right**, or **Center** of the toolbar where it is located.

Button width	If the Fixed button width checkbox is unchecked, the width of a button is determined by its picture or text; otherwise, the width is fixed and can be specified using the arrows.
Button height	Specifies the height of a button inside the toolbar. The difference between the toolbar height and the button height determines the vertical distance from toolbar border to button border.
Hover color	Defines the hover color for the buttons. Click the button to the right to access color palettes.

Background Controls (NetHelp Only)



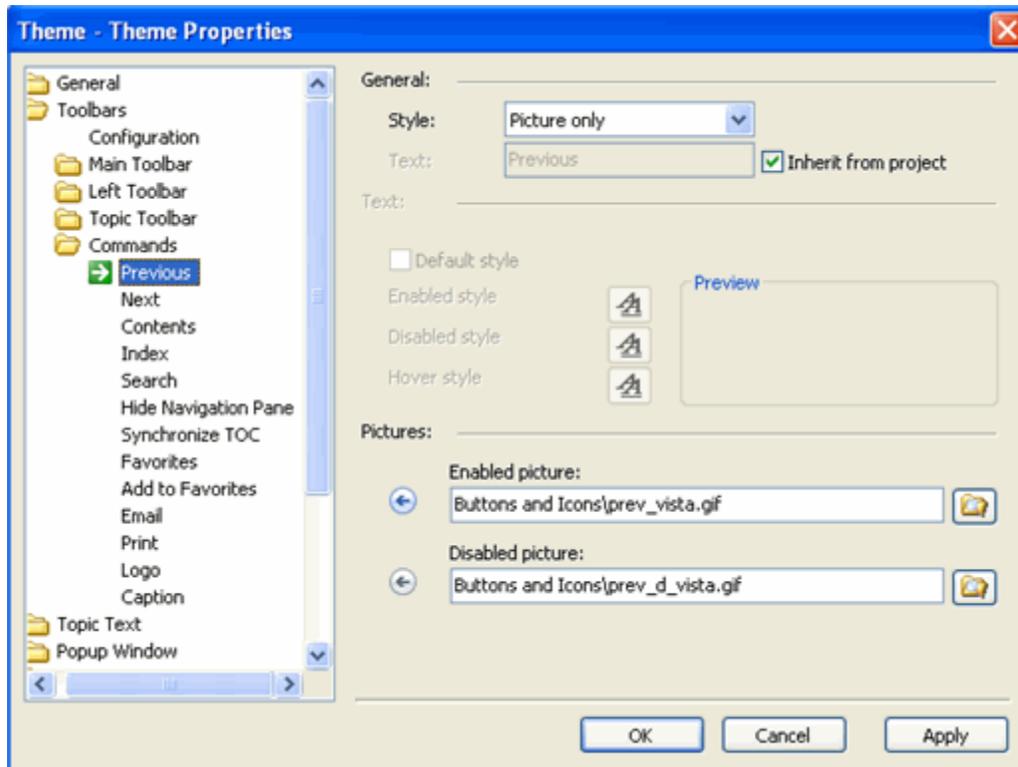
Same as Topic Text	Sets the toolbar's properties so they are inherited from the project settings.
Background color	Defines the color in the toolbar area. Click the button to the right to access color palettes.
Selected button color	Defines the color of a button once it has been clicked.
One-pixel-wide picture for gradient	When selected, the background picture for the gradient can be specified, and it must be one pixel wide. To specify a picture for the gradient, click the browse button  and choose an image file. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Backgrounds.

Commands Node (NetHelp Only)

The Commands node allows you to modify the formatting of the controls, or buttons and links, in the toolbars of the NetHelp target.

The following controls are available for NetHelp targets. Note that the toolbar location of each command can be changed under the Configuration node of the Toolbars node.

<i>Control</i>	<i>Description</i>
Previous	The Previous navigation button that appears in the Main toolbar, by default, and jumps to the previous topic, when clicked.
Next	The Next navigation button that appears in the Main toolbar, by default, and jumps to the next topic, when clicked.
Contents	The Contents button that appears in the Main toolbar, by default, and jumps to the table of contents, when clicked.
Index	The Index button that appears in the Main toolbar, by default, and jumps to the table of contents, when clicked.
Search	The Search button that appears in the Main toolbar, by default, and jumps to the search box, when clicked.
Hide Navigation Pane	The X button that appears in the Left toolbar, by default, and hides the navigation pane when clicked. The navigation pane can be reopened by clicking one of the navigational panel toggle buttons: Contents, Index, Search or Favorites. Set the HelpTarget. HideNavigationPane property in the project editor to True to hide the pane initially, when the user first opens the NetHelp target.
Synchronize TOC	The default Sync TOC button that appears when the Synchronize TOC button is added under the Configuration controls in the Toolbars node; when this button is added, automatic TOC synchronization is disabled.
Favorites	The Favorites pane appears if the HelpTarget. FavoritesTab is set to True in the project editor and the Favorites button or link is clicked in the NetHelp target.
Add to Favorites	The Add to Favorites button or link that appears when the Add to Favorites button is added under the Configuration controls in the Toolbars node. When this button or link is clicked in the NetHelp target, the current topic is added to the list of Favorites.
Email	The Email button that appears in the Main toolbar, by default. When clicked, the button opens your default email program, creates an email with a link to the current topic, and allows you to finalize the text and send the email. The initial email subject line and address are customizable. See The Message Files Commands (NetHelp Only) (page 597) for more information.
Print	The Print button that appears in the Main toolbar, by default, and prints the topic when clicked.
Logo	The Logo text and/or picture that appears in the Main toolbar, by default. The Action property described below determines what happens when the Logo is clicked.
Caption	The Caption text and/or picture that appears in the navigation pane. The Action property shows the name of the navigation tab currently selected by the user.



The following properties can be set for each control:

Style	Defines the Commands controls as text, image or text and image.
Text	Defines text for the Commands controls' links. You may choose to use the variable <i>%LeftPaneCaption%</i> in the text to show the caption of the currently selected navigation pane, either Contents , Index , Search or Favorites . For example, if a button's text is "This is the <i>%LeftPaneCaption%</i> ", and the Contents tab is selected, "This is Contents" will be displayed. When the Index tab is selected, "This is Index" will be displayed. The <i>%LeftPaneCaption%</i> variable is used in the default theme's Caption command control.
Inherit from project	Determines if the button or link text is inherited from the project.
Default style	Determines if the button or link style is inherited from the project.
Enabled style	Defines the enabled text style for the button or link.
Disabled style	Defines the disabled text style for the button or link.
Hover style	Defines the hover text style for the button or link.
Enabled picture	Defines the enabled image for the button or link. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Buttons and Icons
Disabled picture	Defines the disabled image for the button or link. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

If a custom button was added in the Configuration node of the Toolbars node, it will appear under the list of Commands controls. For example, the built-in NetHelp themes have two custom commands: **Logo** and **Caption**. You

will notice these in the list of Commands. These commands will have all of the properties listed above, just as other clickable commands like **Email** have, but they also have one additional property: **Action**.

Action Defines the string value of the 'href' attribute of the <A> tag which reacts when the custom button is clicked. If you want to open a topic or an external Web page, enter a URL here, <http://www.doctohelp.com>, for example. If you want to execute a script, enter a JavaScript here, as it is done in the 'href' attribute of an <A> tag in regular HTML code.

Exploring the Topic Text Node

The Topic Text node allows you to modify the formatting of the topic pane of your Help target.

Layout Controls

Add margins to the HTML source Adds a small padding, or margins, to the theme by default. If you have defined margins in your HTML source document and do not want Doc-To-Help to change them, uncheck this box. This check box applies to HTML source documents only.

Combine topic title with toolbar (for NetHelp only) Determines if the topic title is included in the navigation area.

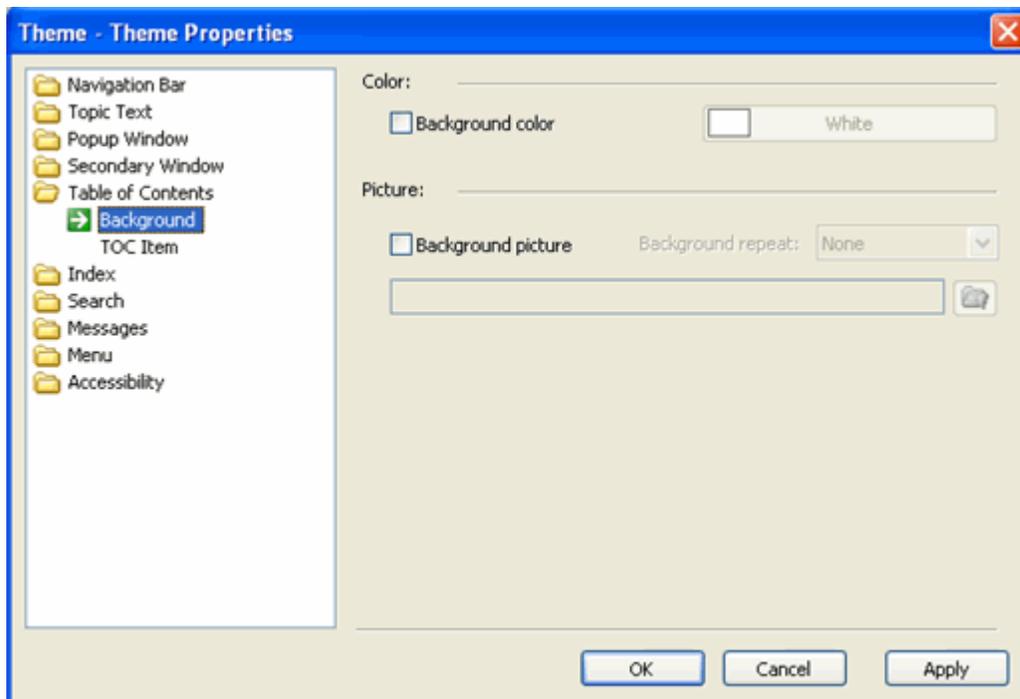
Topic title position (for NetHelp only) If **Combine topic title with toolbar** is set to **True**, this property specifies the relative positions of the toolbar and topic, whether the topic title is above or below the toolbar.

Toolbar in Nonscrolling area (for NetHelp only) Determines if the navigational area is nonscrolling.

Exploring the Table of Contents Node (NetHelp Only)

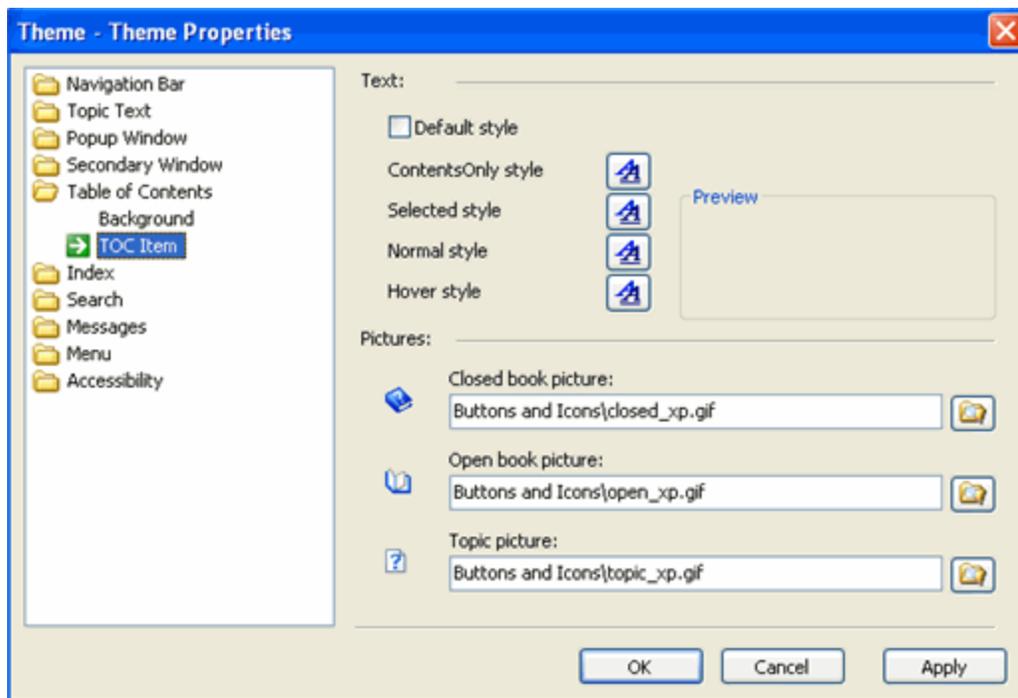
The Table of Contents node allows you to modify the background and text formatting of the table of contents area of your NetHelp target.

Background Controls (NetHelp Only)



Background color	Defines the color in the table of contents area. Click the button to the right to access color palettes.
Background picture	Defines the background image in the table of contents area. Click the browse button  to access the image directory. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Backgrounds
Background repeat	Defines where the image will appear in the background.

The TOC Item Commands (NetHelp Only)

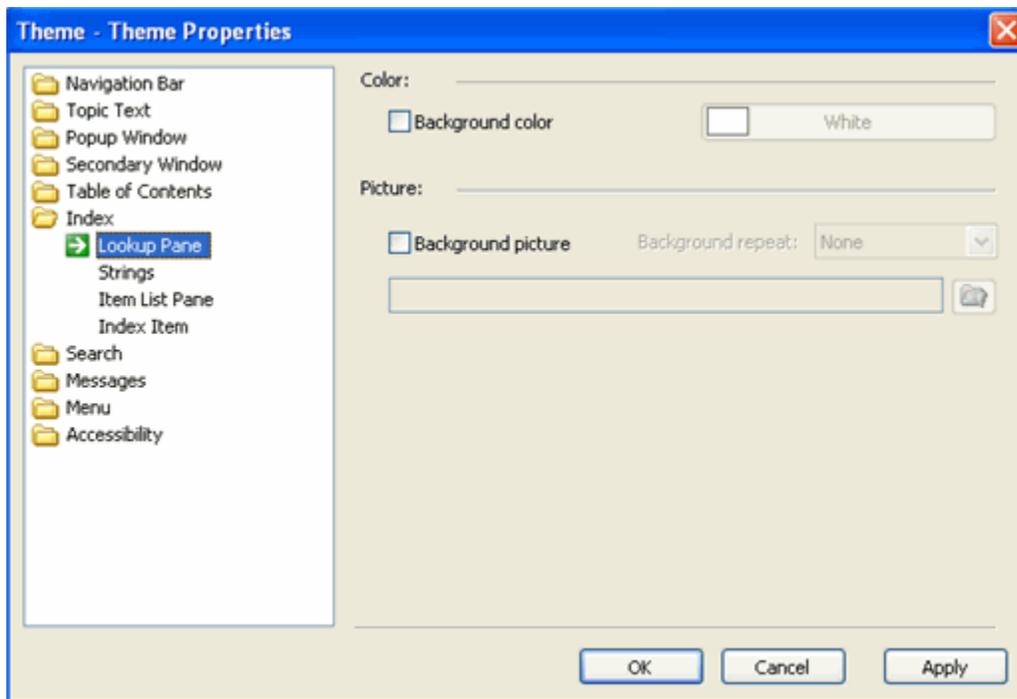


Default style	Determines if the TOC style is inherited from the project.
ContentsOnly style	Defines the text style for the TOC items defined as Contents Only.
Selected style	Defines the text style for TOC items that have been visited.
Normal style	Defines the text style for the TOC items that have not been visited.
Hover style	Defines the hover text style for the TOC items.
Closed book picture	Defines the closed image for the TOC. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Buttons and Icons
Open book picture	Defines the opened image for the TOC. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Buttons and Icons
Topic picture	Defines the topic image for the TOC. Note: ComponentOne suggests that any images you want to use should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Exploring the Index Node (NetHelp Only)

The Index node allows you to modify the background and text formatting of the index area of your NetHelp target.

Lookup Pane Controls (NetHelp Only)



Background color

Defines the color in the index area. Click the button to the right to access color palettes.

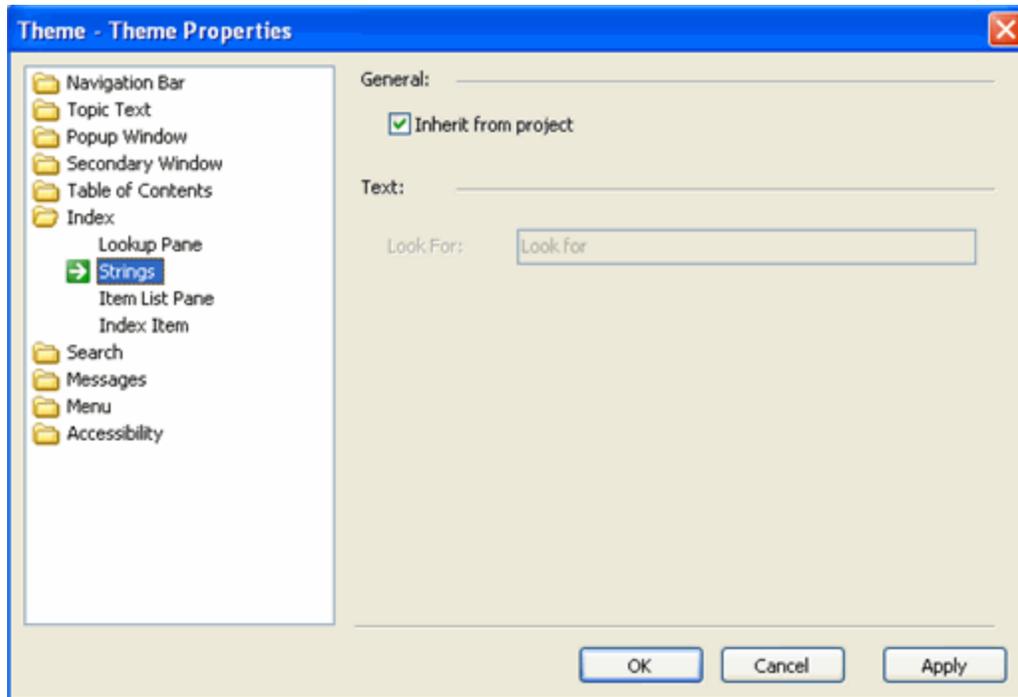
Background picture

Defines the background image in the index area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat

Defines where the image will appear in the background.

The Strings Commands (NetHelp Only)



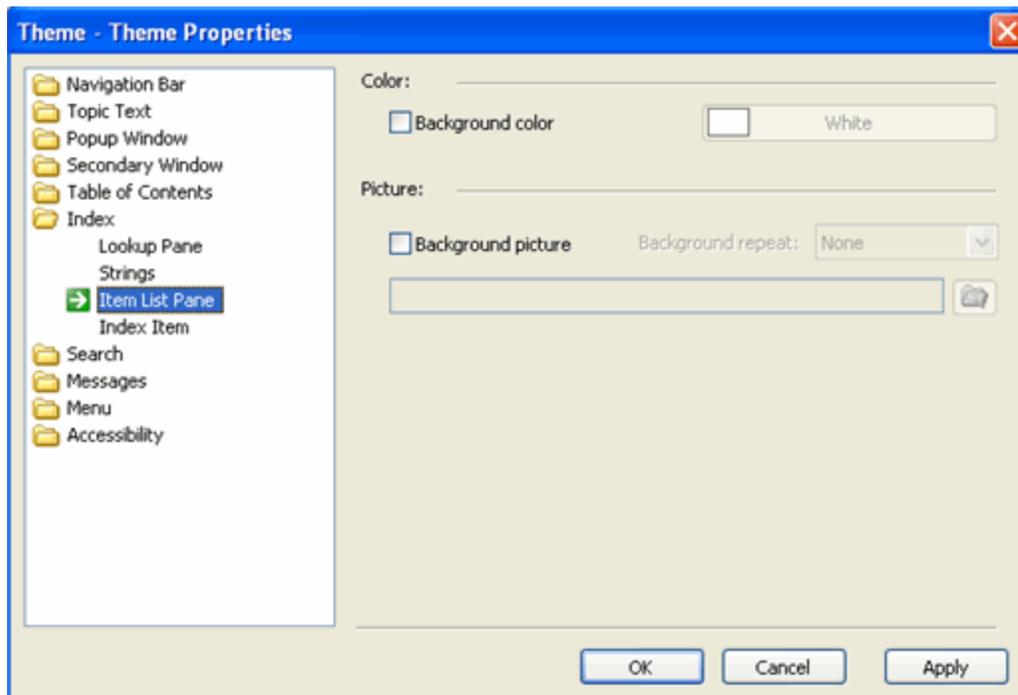
Inherit from project

Sets the theme properties so they are inherited from the project settings.

Look for

Defines the text to be inserted above the search text box.

The Item List Pane Commands (NetHelp Only)

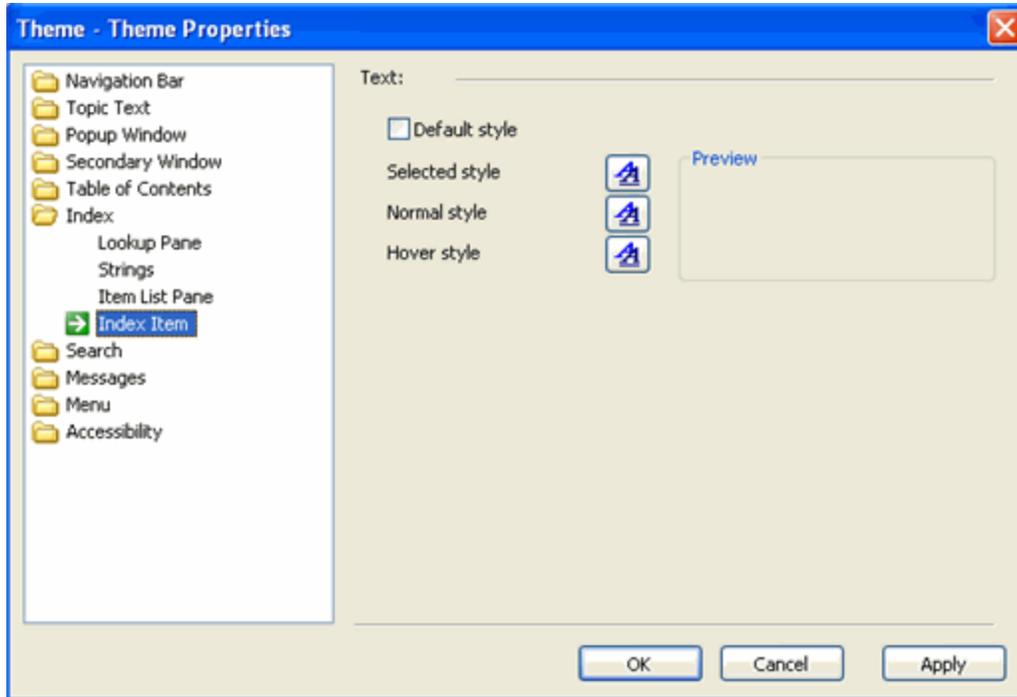


Background color Defines the color in the index area. Click the button to the right to access color palettes.

Background picture Defines the background image in the index area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat Defines where the image will appear in the background.

The Index Item Commands (NetHelp Only)



Default style Determines if the Index area style is inherited from the project.

Selected style Defines the text style for Index items that have been visited.

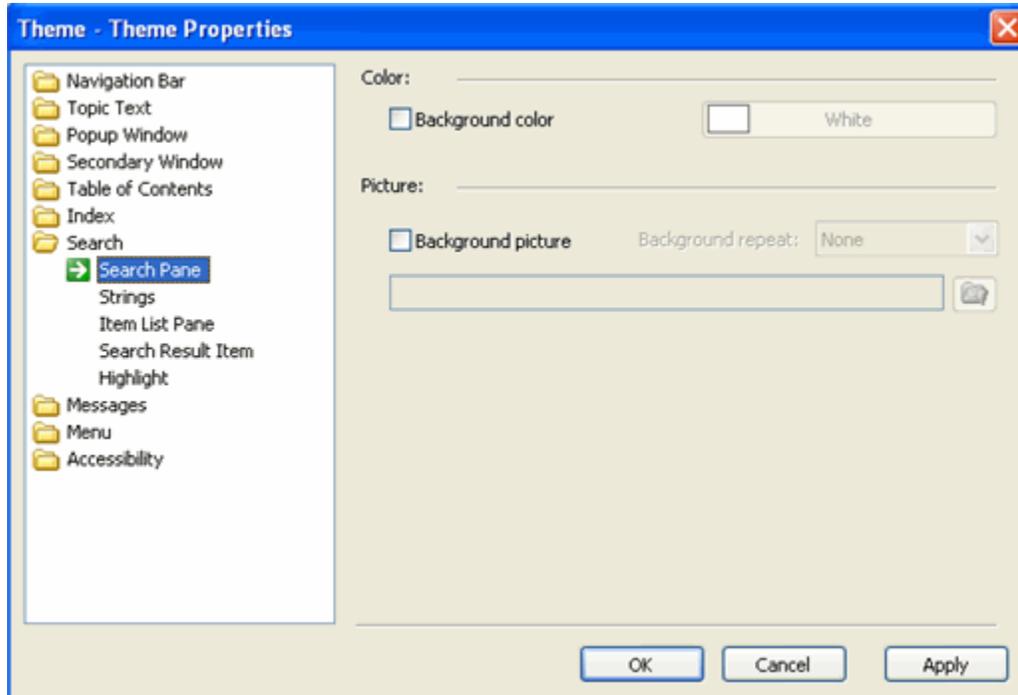
Normal style Defines the text style for the Index items that have not been visited.

Hover style Defines the hover text style for the Index items.

Exploring the Search Node (NetHelp Only)

The Search node allows you to modify the background and text formatting of the Search area of your NetHelp target.

Search Pane Controls (NetHelp Only)



Background color

Defines the color in the Search area. Click the button to the right to access color palettes.

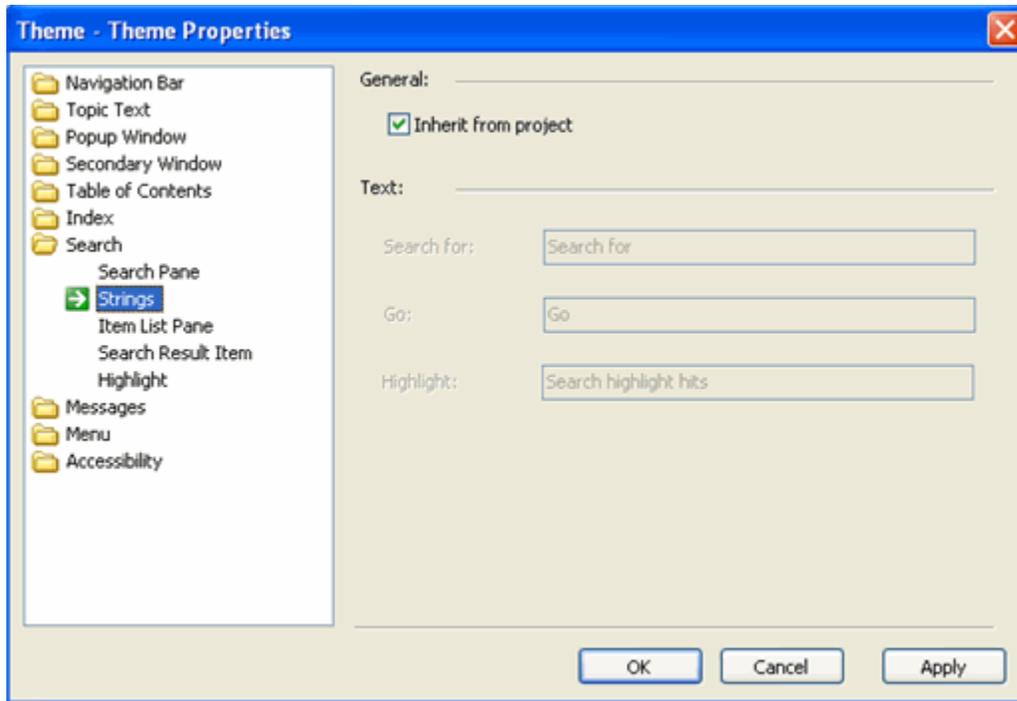
Background picture

Defines the background image in the Search area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat

Defines where the image will appear in the background.

The Strings Commands (NetHelp Only)



Inherit from project

Sets the theme properties so they are inherited from the project settings.

Search for

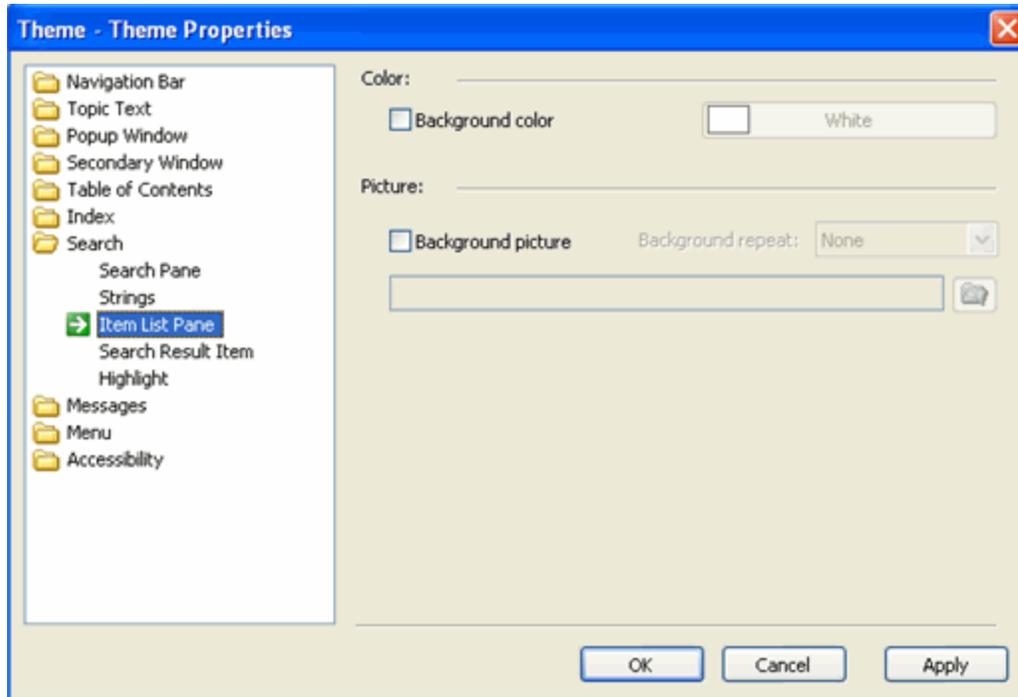
Defines the text to be inserted above the search text box.

Go

Defines the text to be used for the search button.

Highlight

Defines the text to be used for the checkbox controlling highlighting search phrases.

Item List Pane (NetHelp Only)**Background color**

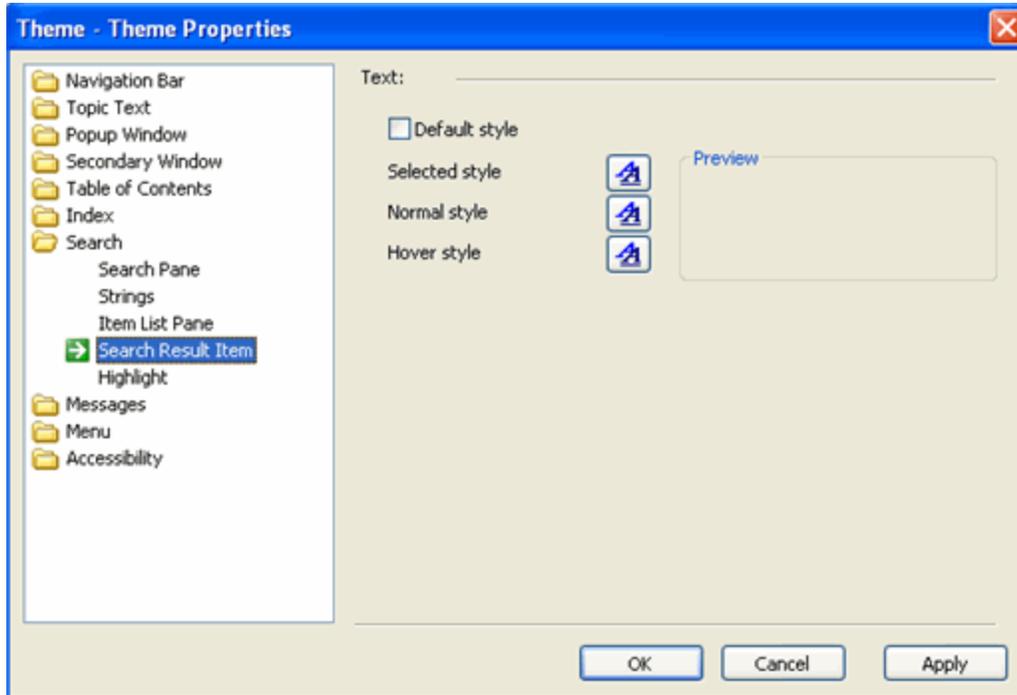
Defines the color in the Search area. Click the button to the right to access color palettes.

Background picture

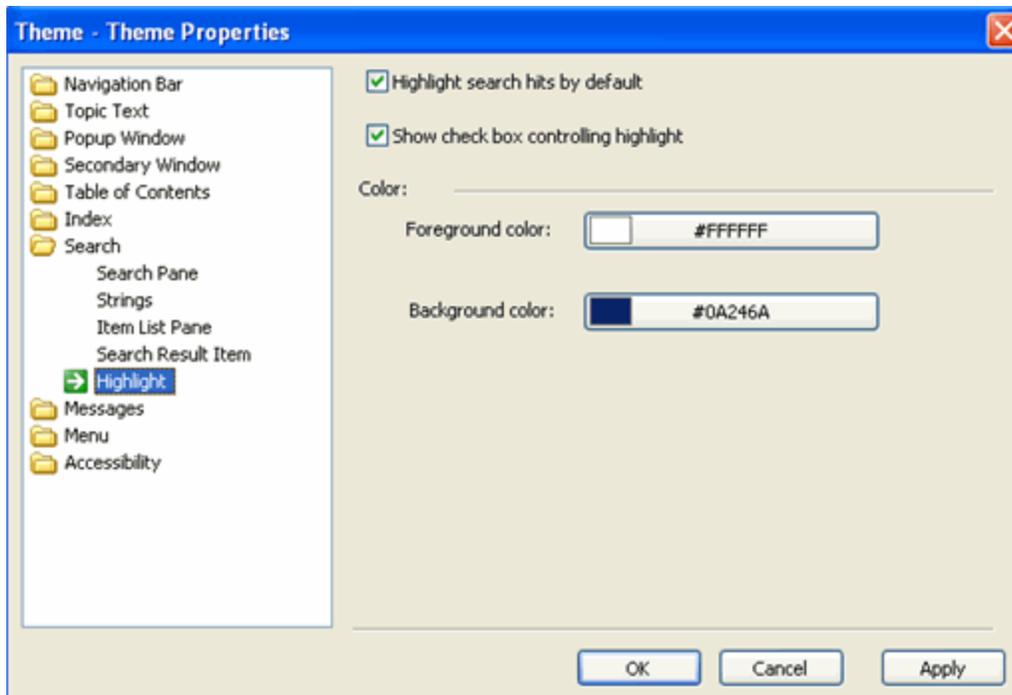
Defines the background image in the Search area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat

Defines where the image will appear in the background.

The Search Result Item Commands (NetHelp Only)

Default style	Determines if the Search area style is inherited from the project.
Selected style	Defines the text style for Search items that have been selected from the listed.
Normal style	Defines the text style for the Search items that have not been selected.
Hover style	Defines the hover text style for the Search items.

The Highlight Commands (NetHelp Only)

Highlight search hits by default

Determines if search phrases are, by default, highlighted in the topic text during a search. If checked, the **Highlight search hits** checkbox is checked in the NetHelp target's Search tab.

Show check box controlling highlight

Determines if the **Highlight search hits** checkbox is visible in the NetHelp target's Search tab.

Foreground color

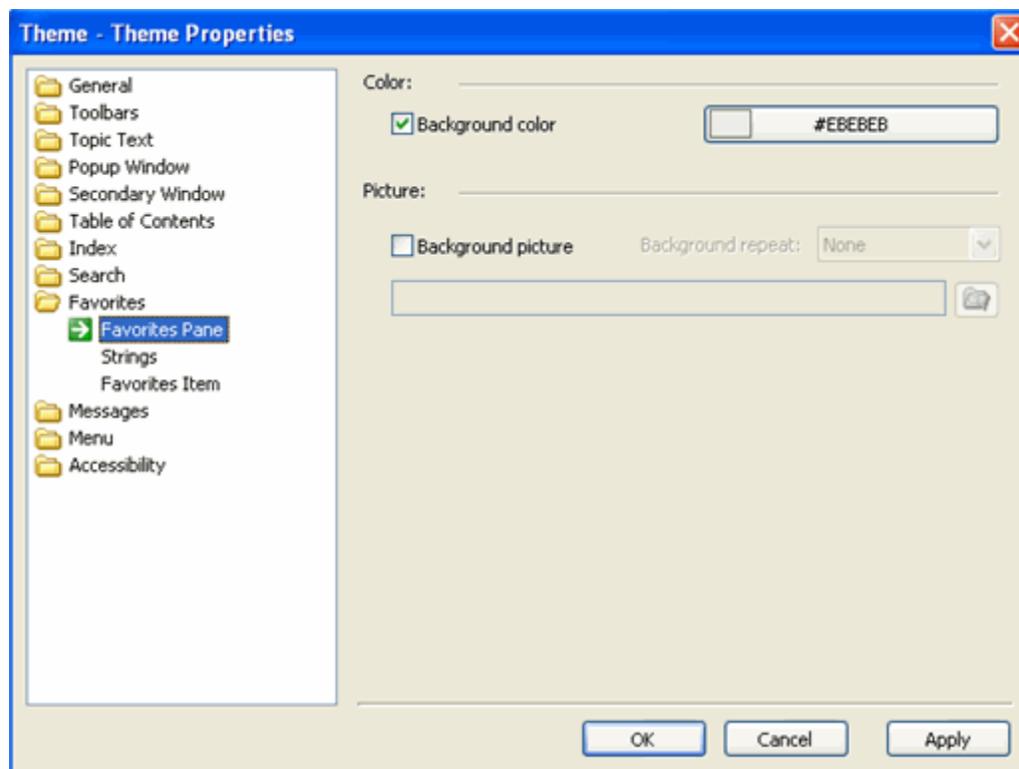
Determines the foreground color of the highlighted phrase.

Background color

Determines the background color of the highlighted phrase.

Exploring the Favorites Node (NetHelp Only)

The Favorites node allows you to specify the appearance of the Favorites pane of your NetHelp target.

Favorites Pane Controls (NetHelp Only)**Background color**

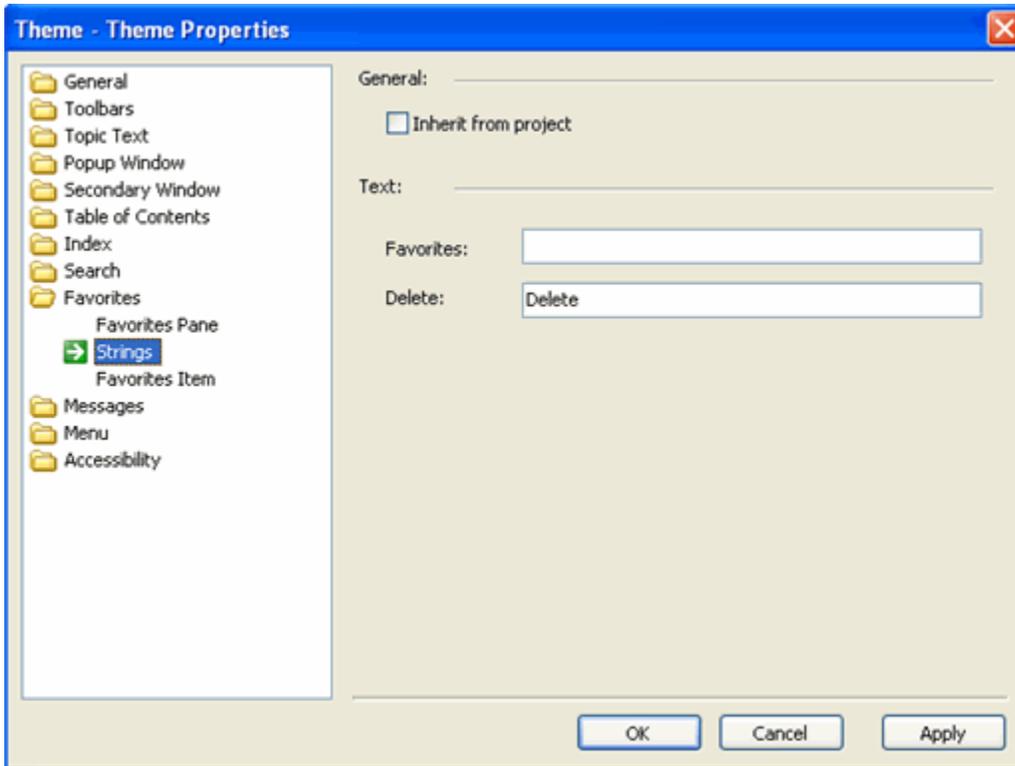
Defines the color in the Favorites area. Click the button to the right to access color palettes.

Background picture

Defines the background image in the Favorites area. Click the browse button  to access the image directory. **Note:** ComponentOne suggests that any images you want to use should be stored in the following folder:
ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat

Defines where the image will appear in the background.

The Strings Commands (NetHelp Only)**Inherit from project**

Sets the theme properties so they are inherited from the project settings.

Favorites

Defines the text in the **Favorites** pane.

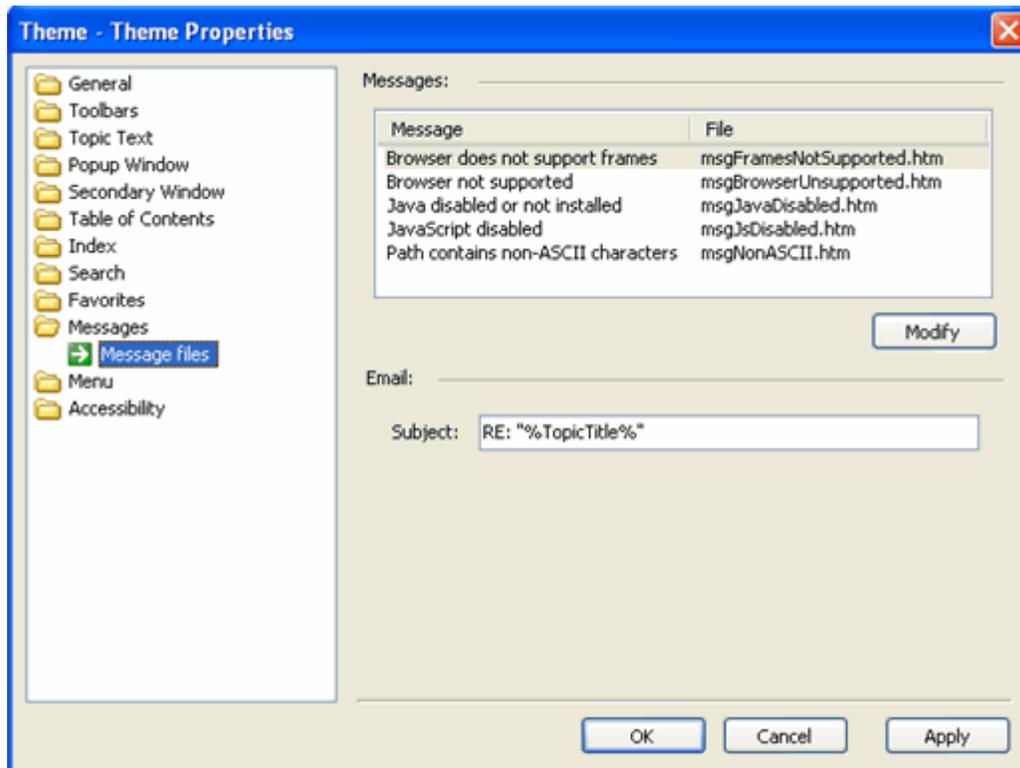
Delete

Defines the text for the link used to remove a favorites item in the **Favorites** pane.

Exploring the Messages Node (NetHelp Only)

The Messages node allows you to customize the error messages that appear in the NetHelp target. If an error occurs while using the help file, this message is displayed in an HTML page. You may use the default message files or create your own.

The Message Files Commands (NetHelp Only)



Javascript disabled

Displays the msgJsDisabled.htm file (default), stating there is an Error: Javascript is disabled, or you can specify a file to display your own message.

Java disabled or not installed

Displays the msgJavaDisabled.htm file (default), stating there is an Error: Java is disabled or not installed, or you can specify a file to display your own message.

Browser not supported

Displays the msgBrowserUnsupported.htm file (default), stating the Search engine does not support this browser, or you can specify a file to display your own message.

Browser does not support frames

Displays the msgFramesNotSupported.htm file (default), stating Your browser does not support frames, or you can specify a file to display your own message.

Path contains non-ASCII characters

Displays the msgNonASCII.htm file (default), stating the Search engine failed to initialize because of illegal characters in the path name, or you can specify a file to display your own message.

Modify

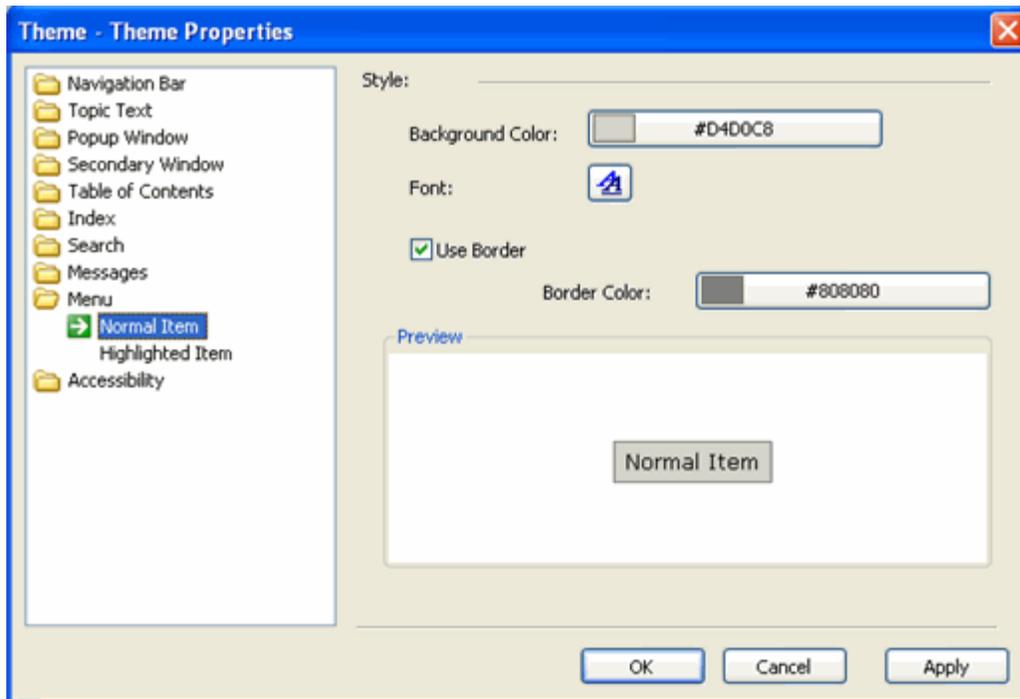
Allows you to add your own message file to the theme. A message can be any HTML that is stored in a file. Create a new HTML file with your message, and click **Modify** to replace the default file with your own file.

Email

Defines the string that appears in the subject line of the email created when a user clicks the **Email** button. The default string is **RE: "%TopicTitle%"**, where **%TopicTitle%** is a variable substituted by the topic title when the email is created. Another variable that can be used here is **%TopicURL%**, which is substituted by the URL of the topic. The email address can be set in the project editor using the HelpTarget.[EmailAddress](#) property.

Exploring the Menu Node (NetHelp Only)

The Menu node allows you to customize the menu that is displayed when the user clicks a link that has multiple destinations.

The Normal Item Commands (NetHelp Only)**Background Color**

Defines the color of the menu. Click the button to the right to access color palettes.

Font

Defines the font for menu items.

Use Border

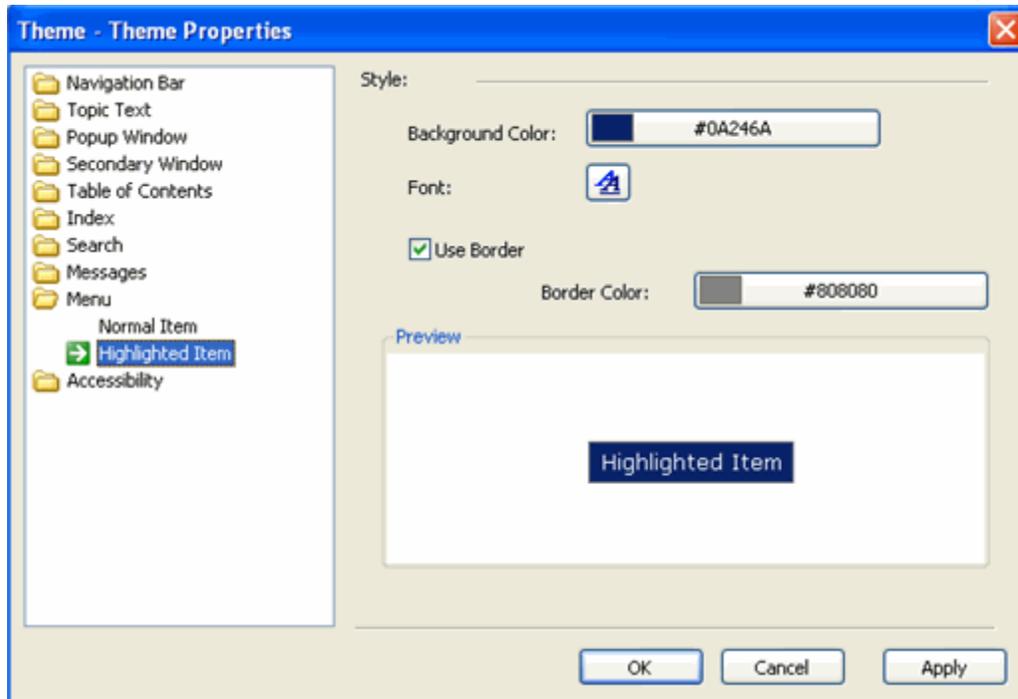
Determines whether a border appears around the menu.

Border Color

Defines the color of the border. Click the button to the right to access color palettes.

Preview

Shows a preview of how the menu will appear in the help file.

The Highlighted Item Commands (NetHelp Only)**Background Color**

Defines the color of a highlighted item in the menu. Click the button to the right to access color palettes.

Font

Defines the font used for highlighted menu items.

Use Border

Determines whether a border appears around a highlighted item in the menu.

Border Color

Defines the color of the border. Click the button to the right to access color palettes.

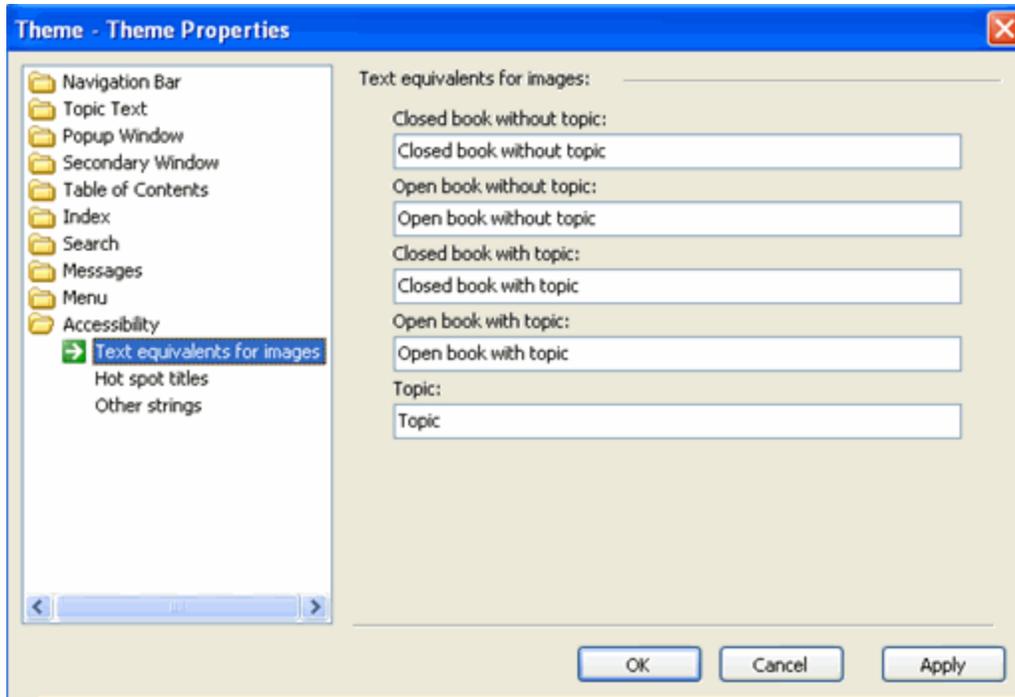
Preview

Shows a preview of how the menu will appear in the help file.

Exploring the Accessibility Node (NetHelp Only)

The Accessibility node allows you to specify the strings that are shown as text equivalents for table of contents images and to specify titles for hot spots. This is the text that is read by accessibility devices.

The Text Equivalents for Images Commands (NetHelp Only)



Closed book without topic

This text is read by the accessibility device when you mouse over a closed book that has no associated topic and, therefore, does not display a topic when clicked. The default is "Closed book without topic".

Open book without topic

This text is read by the accessibility device when you mouse over an open book that has no associated topic and, therefore, does not display a topic when clicked. The default is "Open book without topic".

Closed book with topic

This text is read by the accessibility device when you mouse over a closed book that has an associated topic, which is displayed when the book is clicked. The default is "Closed book with topic".

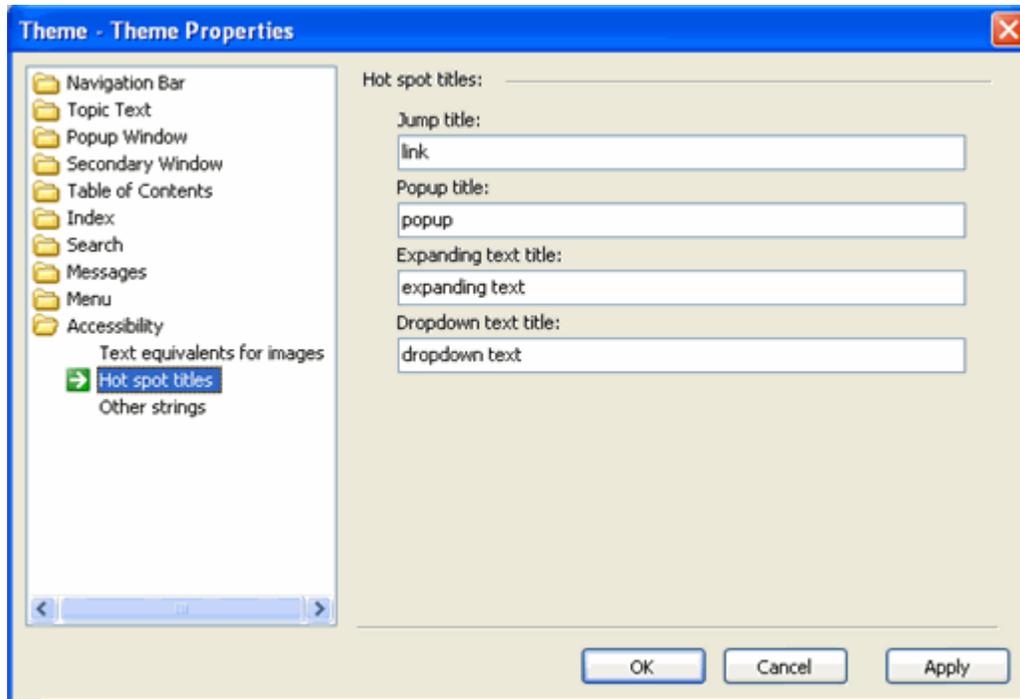
Open book with topic

This text is read by the accessibility device when you mouse over an open book that has an associated topic, which is displayed when the book is clicked. The default is "Open book with topic".

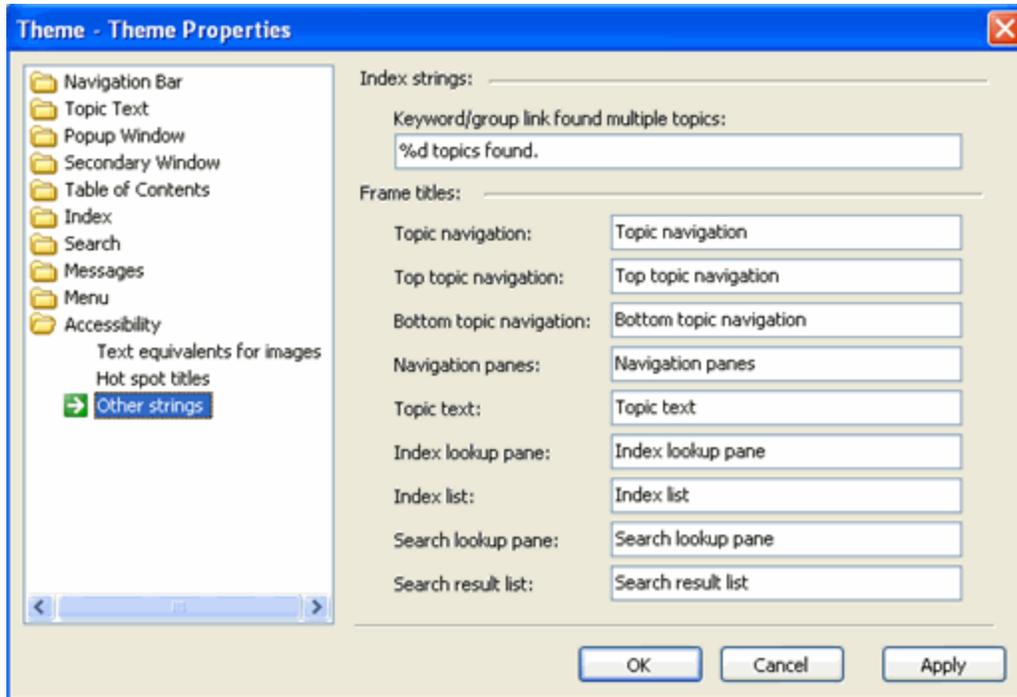
Topic

This text is read by the accessibility device when you mouse over a topic. The default is "Topic".

The Hot Spot Titles Commands (NetHelp Only)



- Jump title** This text is read by the accessibility device when you mouse over a hot spot formatted with the C1H Jump style. The default is **link**.
- Popup title** This text is read by the accessibility device when you mouse over a hot spot formatted with the C1H Popup style. The default is **popup**. Note that pop-up links appear as jump links for easier accessibility.
- Expanding text title** This text is read by the accessibility device when you mouse over a hot spot formatted with the C1H Inline Expand and C1H Expand Text styles. The default is **expanding text**.
- Dropdown text title** This text is read by the accessibility device when you mouse over a hot spot formatted with the C1H inline Dropdown and C1H Dropdown Text styles. The default is **dropdown text**.

The Other Strings Commands (NetHelp Only)**Keyword/group link found multiple topics**

The text specified here is read by the accessibility device when the Topics Found HTML page appears. This is the page that appears when a user clicks a group or keyword link or a keyword in the index that has multiple destinations, or target topics. The default text for this command is **%d topics found**, where **%d** is the number of topics found.

Topic navigation

This text is read by the accessibility device for the frame containing the navigation bar. The default is **Topic navigation**.

Top topic navigation

This text is read by the accessibility device for the frame containing the navigation buttons when it appears at the top of a topic. The default is **Top topic navigation**.

Bottom topic navigation

This text is read by the accessibility device for the frame containing the navigation buttons when it appears at the bottom of a topic. The default is **Bottom topic navigation**.

Navigation panes

This text is read by the accessibility device for the frame containing navigation buttons or text. The default is **navigation panes**.

Topic text

This text is read by the accessibility device for the frame containing topic text. The default is **Topic text**.

Index lookup pane

This text is read by the accessibility device for the frame containing the index search text box. The default is **Index lookup pane**.

Index list

This text is read by the accessibility device for the frame containing the list of indexed topics returned when an index keyword search is performed. The default is **Index list**.

Search lookup pane

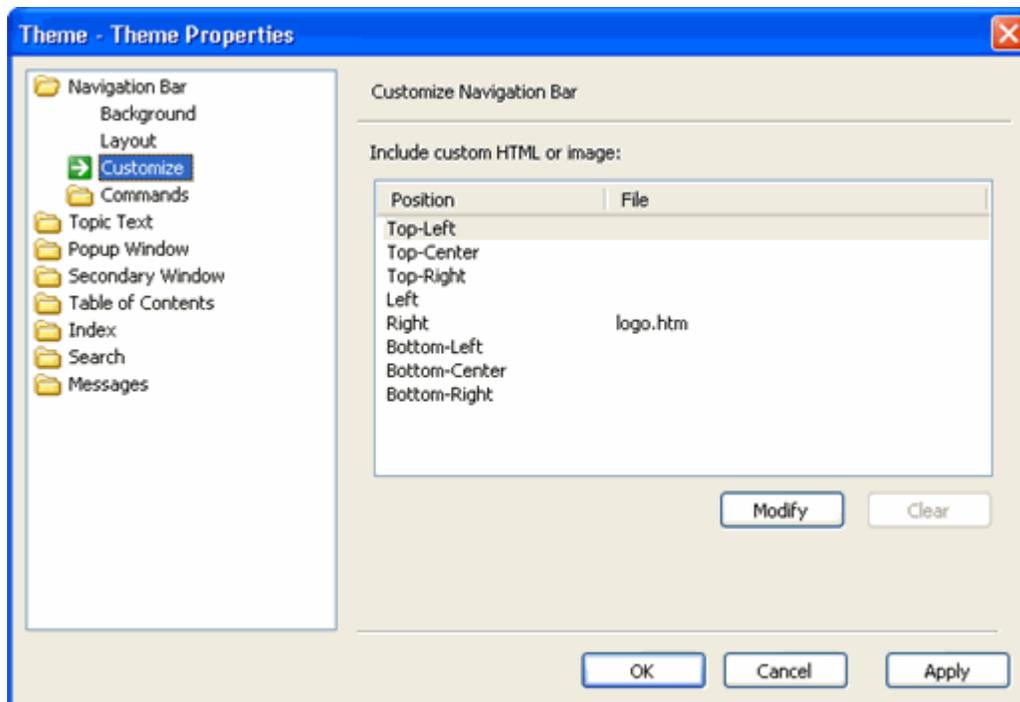
This text is read by the accessibility device for the frame containing the search text box. The default is **Search lookup pane**.

Search result list

This text is read by the accessibility device for the frame containing the list of topics returned when a search is performed. The default is **Search result list**.

Including Custom HTML Content in Themes

In Doc-To-Help 2006 you can include additional arbitrary HTML content in your themes. For example, to include a company logo or a link in a theme, prepare an HTML file (only the body, without <BODY> tags and anything outside them) and include it in the theme by selecting an appropriate Customize node and Position in Theme Designer. Specify the include file there by clicking the **Modify** button.



What is an Include File?

An include file is either an image file or an HTML file. You can have multiple include files in a single theme. The result of the inclusion is the *.theme files that are used to build the help target. For an HTML include file, only its body, the content that is normally placed inside the <BODY> tags should be present in the file. The <BODY> tags themselves and everything outside (<HTML> tags, <HEAD> tags, etc.) should be removed from the include file. Doc-To-Help does not verify this or change the included content. The inclusion is performed literally, the whole contents of the file is included.

Restrictions and Guidelines

Include HTML files and ancillary files (images, links) that are referenced in the include files must reside in the Customize sub-directory of the theme. When you include a file, it will be automatically copied to that directory. If you have ancillary files referenced in that file, you are responsible for putting them in the same directory with the include file. All paths to such ancillary files in HTML must be relative paths to the Customize directory, as in the following example of an HTML include file showing a logo image:

```

```

This and other restrictions on the content of HTML include file are the author's responsibility. Doc-To-Help does not change or verify the include files. Since everything outside <BODY> tags (and the <BODY> tags themselves) must be removed, the author of an include file should put everything that is needed inside the body, including styles and scripts.

Only use unique names of styles, scripts, etc. These names must not conflict with the names used by Doc-To-Help for other purposes. As a recommendation, it is better to avoid using names in include files altogether. Use only inline unnamed constructs.

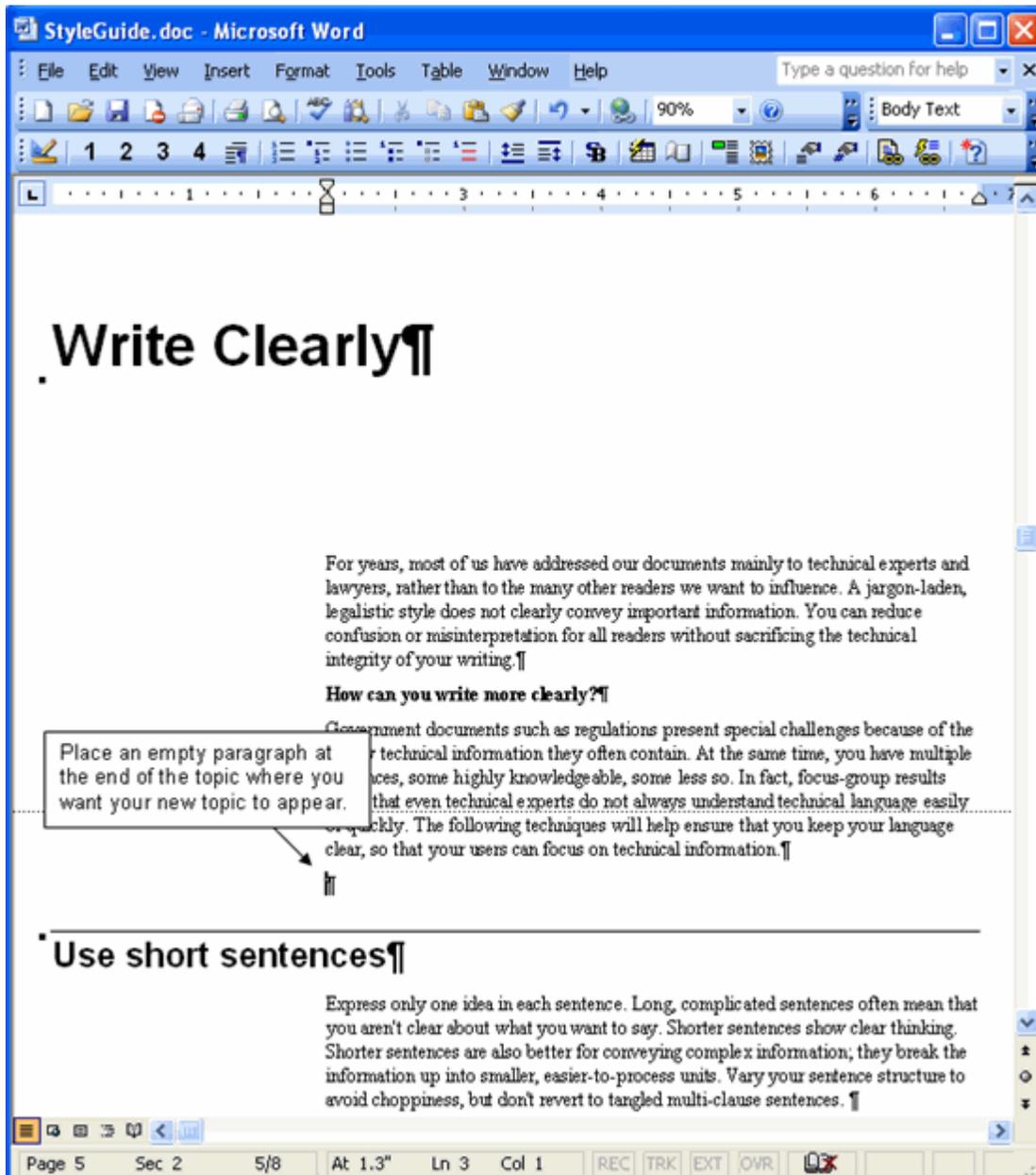
Using the Topic Tools in Word

The **Doc-To-Help** toolbar for Word includes three buttons to make adding and defining topics a simple process. The **Add Topic**, **Rename Topic** and **Delete Topic** buttons give you complete control over all the topics from the Word source documents in your project. Simply select the location for the new topic, click the **Add Topic** button and type the topic text, to create a fully functional topic in your project. More importantly, all the features for Word related to a topic (adding topic links, cross-references, etc.) can be accessed without first building your project. This eliminates the need for time consuming multiple builds.

Using the Add Topic Button

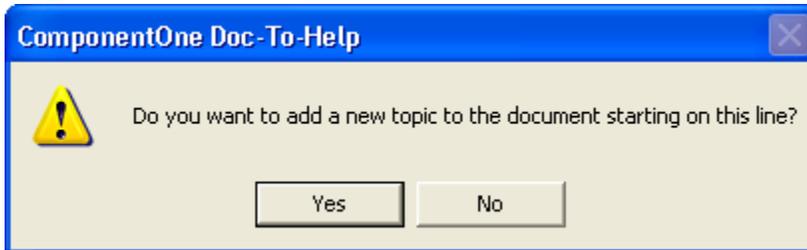
Adding a topic in your source document with the **Add Topic** button is a simple procedure.

1. In your Word document, add an empty paragraph at the end of the topic where you want to add the new topic.

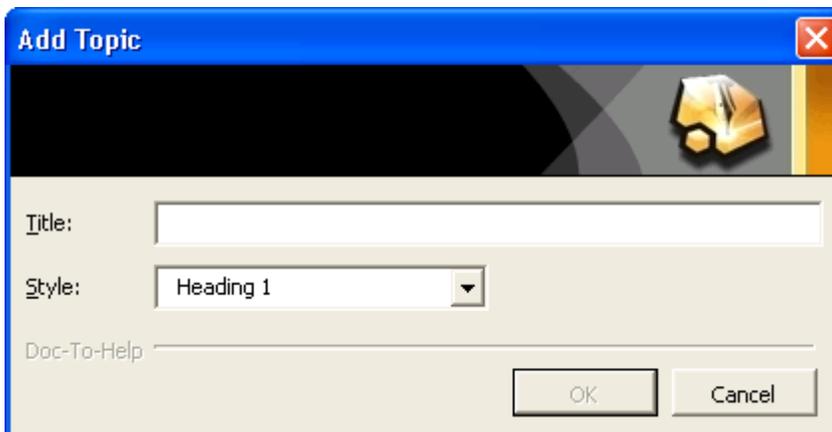


Note: It may be helpful to turn on formatting using the **Tools | Options** menu and checking **Paragraph Marks** under *Formatting marks* on the **Views** tab.

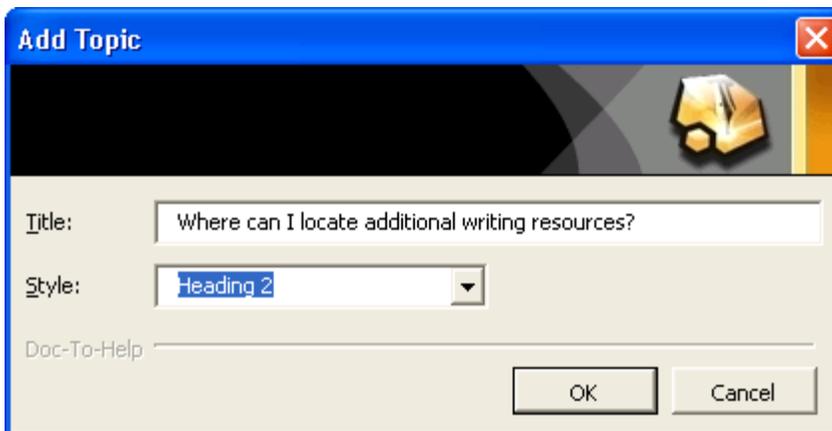
2. Click the **Add Topic** button . Doc-to-Help prompts you with a dialog box asking if you want to add a new topic on the line you have selected.



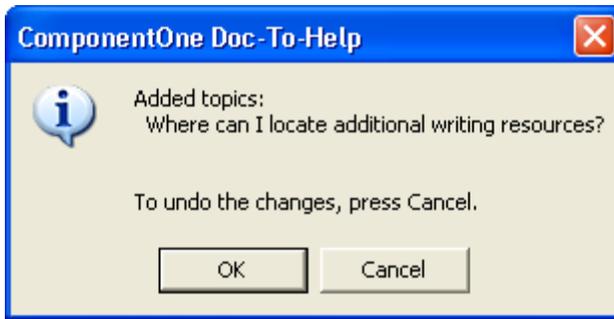
3. Click **Yes**. The **Add Topic** dialog box opens.



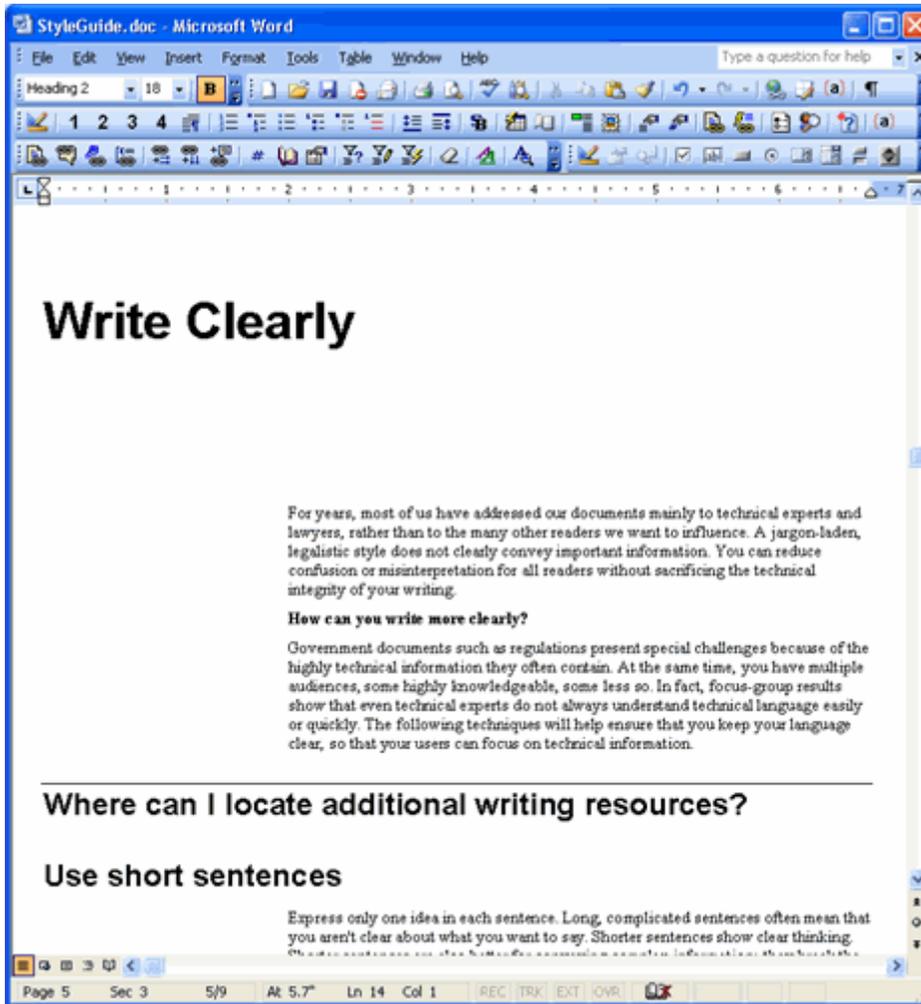
4. Enter the text for your topic and the heading style you want to use.



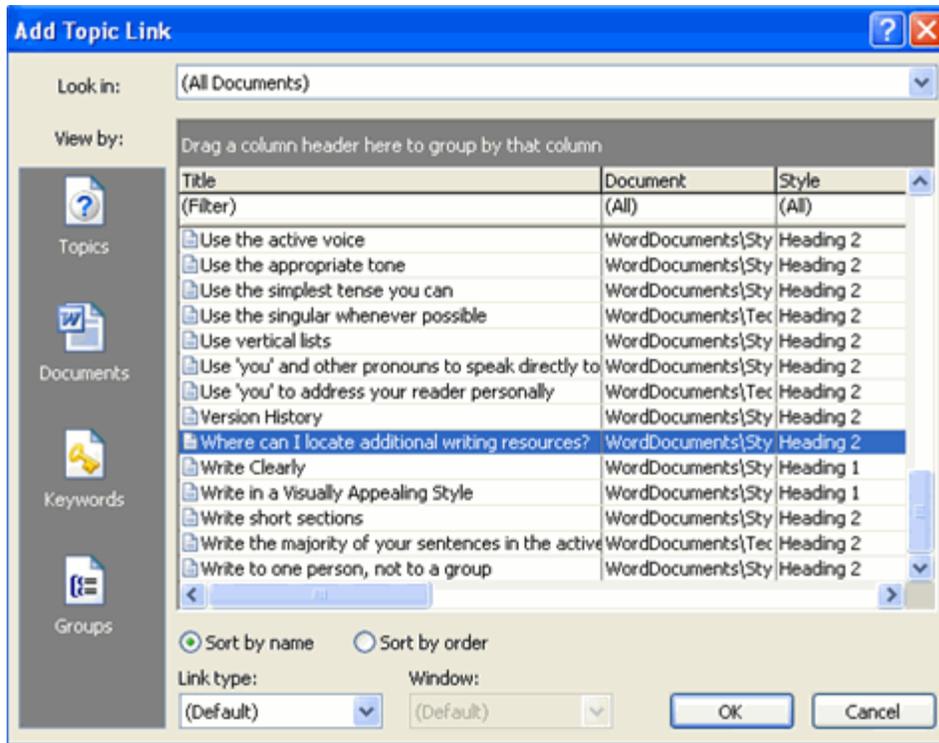
5. Click **OK**. Doc-to-Help prompts you with a dialog box that allows you to see your new topic text and accept it as is or cancel the topic insert.



The new topic is added to your document in the selected location.



You can perform any topic function, for example **Add Topic Link** , without having to rebuild your project first.

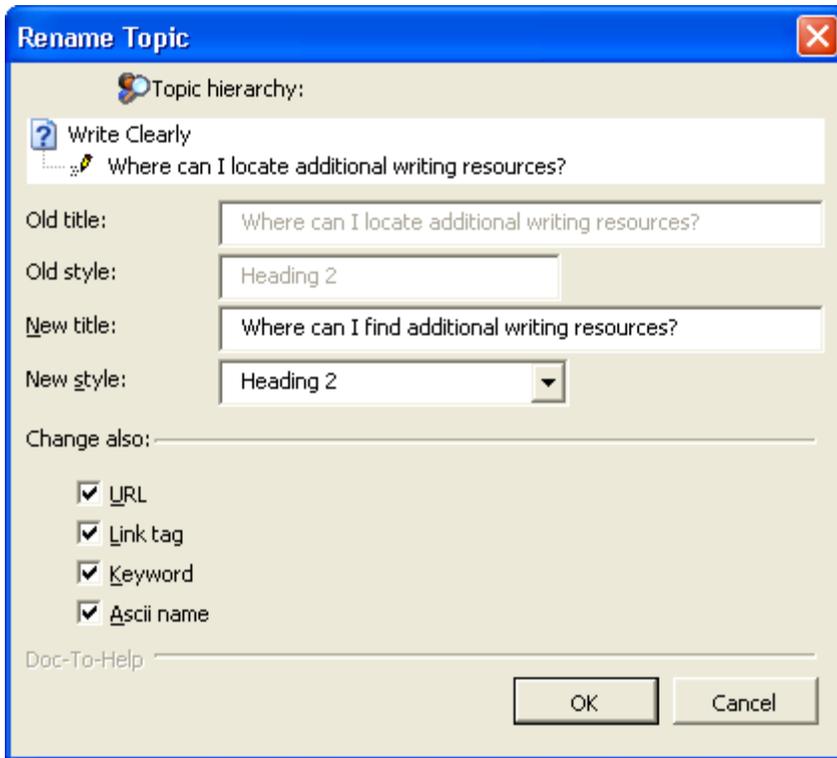


Using the Rename Topic Button

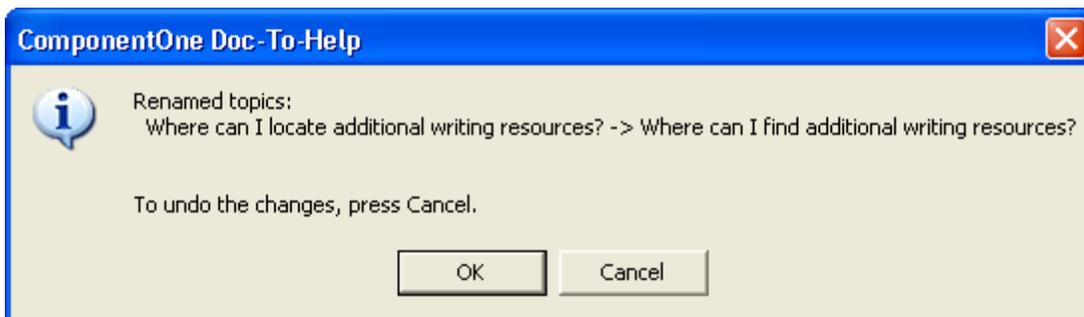
In cases where it is necessary to change the name of a topic, the **Rename Topic** button allows the author to change the topic name, as well as the automatically generated [URL](#), [Link Tag](#) and [Index Keyword](#) related to that topic.

Place your cursor anywhere within the topic to be modified.

1. Click the **Rename Topic** button . The **Rename Topic** dialog opens.



2. Modify the topic in the **New title** textbox. If you do not want Doc-To-Help change the **URL**, **Linktag** or **AutoKeyword** properties, uncheck the related checkboxes.
3. Click **OK**. Doc-to-Help prompts you with a dialog box that allows you to see the changes to your topic text and accept it as is or cancel the topic rename.



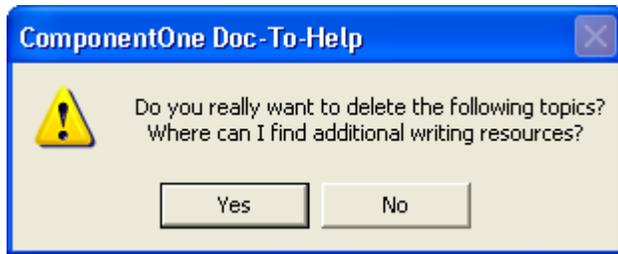
4. After examining the change, click **OK**. Note that the topic is changed as well as the related properties.

Using the Delete Topic Button

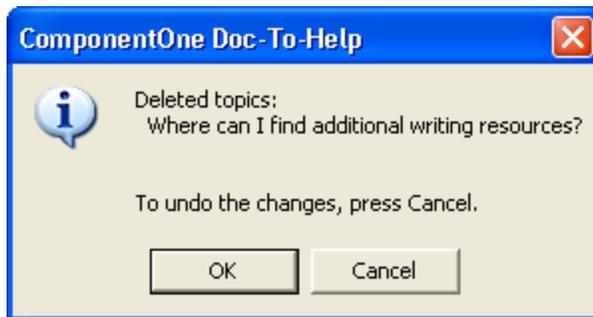
To delete a topic, select the entire topic.

Where can I find additional writing resources?

1. Click the **Delete Topic** button . Doc-to-Help prompts you with a dialog box that allows you to delete the topic or cancel the delete.



2. Click **Yes**. Doc-to-Help prompts you with a dialog box confirming the delete.



3. Click **OK**.

Object Model Reference

Accessibility Property

Determines whether special features for 508 compliance are enabled.

Type	Applies To
Enum	Help Target (NetHelp)

Value	Description
Normal	No special features for Section 508 compliance are enabled.
Section 508 Compliance	Special features for 508 compliance are enabled.

Remarks

The following features are enabled, regardless of the value of the **Accessibility** property:

- All links and buttons in NetHelp themes are accessible from the keyboard using the **Tab** key.
- When the [DynamicTOC](#) property is set to **True** and the NetHelp target is built, the user can expand and collapse books in the table of contents using the Num +/- buttons on the keyboard.
- Each frame of a NetHelp theme has a title string that can be read by accessibility devices. By default, the title strings match the frame titles. These strings can be changed in the [Theme Designer](#) (page 572).

The following features are enabled when the **Accessibility** property is set to **Section 508 Compliance**:

- All links generated by Doc-To-Help have title strings, which indicate the link type, and appear as tooltips that are read by accessibility devices. These strings can be changed in the [Theme Designer](#) (page 572).
- Icons in the table of contents have titles, their text equivalents, indicating whether the item is a book or a topic. If the [DynamicTOC](#) property of the NetHelp target is set to **True** and, therefore, the icon is a book, this title indicates whether it is open or closed. The strings can be changed in the [Theme Designer](#) (page 572).
- Pop-up links become jump links to allow easier accessibility. For the same reason, margin notes and glossary term links, which usually appear as pop-up windows in normal mode, are not shown as pop-ups but as normal HTML pages in the main frame.
- Inline pop-up text is shown as inline, or expanding, text rather appearing in a pop-up box.
- When a user clicks a group or keyword link or a keyword in the index that has multiple destinations, or target topics, the destinations are shown in the main frame as a normal HTML page. The page heading can be changed in the [Theme Designer](#) (page 572).

Location

- Click the Project Icon.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Click on the **Help Targets** item in the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

AdjustForScreenSize Property

Determines whether the help window conforms to the resolution of the reader's display.

Type	Applies To
Boolean	Window (WinHelp)

Remarks

If **True**, the help window will not conform to the resolution of the reader's display.

Location

- Click the Project Icon.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Click on the **Windows** item in the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

AdjustLeftIndent Property

Controls whether paragraph indentation is adjusted to account for wide margins when building online help.

Type	Applies To
Boolean	Document Project Settings

Remarks

By default, this property is **True** to accommodate the standard Doc-To-Help templates. Set this property to **False** if you are using custom templates and want to preserve the indentation used in your source documents.

Location (Document)

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Document Defaults group in the **Properties Pane**.

AffectsAppearance Property

Determines whether the style defines the appearance of the text (font, color, etc.) in the target help file, or is only used to create a hotspot or keyword and does not affect target appearance.

Type	Applies To
Boolean	Character Style

Remarks

When set to **False**, the selected style is used to define hot spots (links) and keywords. It does not determine text formatting (font, color, etc.). The hot spot appearance is determined by the default rules of the target. (For example, links in HTML based targets are underlined blue.)

When set to **True**, the selected style defines the appearance of the text in the help target as well as the functionality of the links and keywords. In this case, formatting in the target help file will be exactly as specified in the document or the target template.

Character styles with an AffectsAppearance Property set to **True** can be used to specify the appearance of links generated by other means. For example, a style with AffectsAppearance Property = **True** and [Type](#) = None can be used to format topic links, dynamic links and margin notes if you need to override the default link appearance for them.

Location

- Click the **Project Icon**.
- Select the **Character Style** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

AlwaysOnTop Property

When **True**, the help window remains on top of all other windows in the desktop.

Type	Applies To
Boolean	Window (HTML Help) Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or WinHelp targets from the **Help Target Dropdown**.
- Click on the **Windows** item in the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

AnswerCount Property

Determines the maximum number of answers to be returned when a natural language search is performed.

Type	Applies To
Integer	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Remarks

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

AsciiName Property

Specifies the ASCII-only string used to identify the topic in situations where non-ASCII characters are not allowed.

Type	Applies To
String	Topic

Remarks

This property is important for Help authoring in languages that have non-ASCII alphabets, such as Cyrillic and Asian languages. In some instances, names generated by Doc-To-Help must be ASCII. Such instances include identifiers in

*.h and *.bas map files that are used for context-sensitive help in C and Visual Basic programming languages, see **MapFileForC**, **MapFileForVB**.

Also, topic file URLs in HTML-based Help are also usually ASCII-only (controlled by the property **URLMode** with three possible values Full/AsciiOnly/TopicID). This is not a problem for languages with ASCII alphabets (English), and it is usually not a problem for languages that have only few non-ASCII letters (such as German). In these cases, the names, formed automatically from topic titles, omit the non-ASCII characters. In languages that are completely non-ASCII (such as Russian or Japanese) no part of the topic title can be used (ALL characters are omitted), so the names become a construct of underscores and numbers. By using the **AsciiName** property, you can make these names mnemonic.

You can change the default **AsciiName** property value for any topic to an ASCII string, and that string will be used as the topic identifier in a C or Visual Basic map file. It will also be used as the default value for the topic URL property, if **URLMode=AsciiOnly** and the topic URL property is empty.

Location

- Click the **Topic Icon**.
- The property is located under the Misc group in the **Properties Pane**.

AttributeExpression Property

Specifies an advanced conditional build expression for a help target.

Type	Applies To
String	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp) Help Target (Manual)

Remarks

By default, this property is empty, meaning that the condition is determined by the target's attributes, or the **Attributes** tab in the property browser for this target. If you enter a non-empty condition in the **AttributeExpression** property, the **Attributes** tab in the property browser becomes unavailable because the condition is now determined by this property.

The **AttributeExpression** property allows you to use any combination of attributes with the logical operators OR, AND and NOT. For example, if you enter the following condition in the **AttributeExpression** property, only conditional text with both attribute values *Novice* and *Expert* applied or with the attribute value *Administrator* applied appears in the Help target.

Skill:Novice AND Skill:Expert OR Skill:Administrator

If you specifying the following condition in the **AttributeExpression** property, only conditional text that has the attribute value *Expert* and not *Novice* applied appears in the Help target:

Skill:Expert AND NOT Skill:Novice

Clear the **AttributeExpression** property to access the **Attributes** tab in the property browser again.

See [Using Attributes](#) (page 422) for more information on attributes and the **AttributeExpression** property.

Location

- Click the **Project Icon**.

- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

AutoButtons Property

When **True**, displays subtopic buttons for paragraph styles with numeric outline levels greater than the parent level.

Type	Applies To
Boolean	Paragraph Style Topic Type

Remarks

To enable the subtopic buttons for a paragraph style, highlight the specific paragraph style in the upper right pane, then set the AutoButtons property to **True**.

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

AutoContextID Property

When **True**, automatically creates Context ID's for all topics using this paragraph style or topic type.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

AutoGlossaryLinks Property

When **True**, Doc-To-Help examines the project documents for text that matches glossary entries in the glossary document during compilation, converting these matches into hyperlinks. Whenever the user clicks on one of the hyperlinks, a DHTML pop-up opens containing the corresponding glossary entry description.

Type	Applies To
Boolean	Paragraph Style

Location

- Click the **Project Icon**.
- Select the **Paragraph Styles** item from the left pane Tree View.
- The property is located under the Glossary group in the **Properties Pane**.

AutoHideNavigationPane Property

Minimizes the HTML help navigation pane when HTML help is not the active window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Click on the **Windows** item in the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

AutoIndex Property

When **True**, this property automatically creates index keywords from text formatted with that character style and associates them with the topic that contains the text.

Type	Applies To
Boolean	Character Style

Location

- Click the **Project Icon**.
- Select the **Character Styles** item from the left pane Tree View.
- The property is located under the Indexing group in the **Properties Pane**.

AutoKeyword Property

When **True**, this property automatically creates index keywords from topic titles formatted with that paragraph style or topic type and associates them with the appropriate topic.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Indexing group in the **Properties Pane**.

AutoLink Property

Creates unique link tags for topics formatted with the paragraph style or topic type by enabling the auto link tag feature.

Type	Applies To
Boolean	Paragraph Style Topic Type

Remarks

Link tags created in this manner are identical to the topic title except that spaces, hyphens and periods are converted into underscores.

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

AutoNavigate Property

Determines which paragraph styles and topic types are included in the navigation sequence.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

AutoNext Property

Enables the next topic button for any active paragraph style.

Type	Applies To
Boolean	Paragraph Style Topic Type

Remarks

Displays a next topic button at the end of each associated topic that links to the next topic.

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

AutoSizeHeight Property

When **True**, the height of secondary Help windows are automatically resized to fit the length of the current topic.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The Property is located under the Position group in the **Properties Pane**.

AutoSyncNavigationPane Property

Allows the heading or topic in the Navigation pane to automatically synchronize with whatever topic appears in the Topic pane.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The Property is located under the Navigation group in the **Properties Pane**.

BackgroundColor Property

Displays the current background color setting for the Help pane. To change the color, click the adjacent change button to open a standard color selection dialog.

Type	Applies To
RGBColor	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Appearance group in the **Properties Pane**.

BackgroundImage Property

Displays the image file to be used as a background in the Help pane. To insert a background image, click the adjacent change button to open the selection dialog.

Type	Applies To
String	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Appearance group in the **Properties Pane**.

BackgroundRepeat Property

Defines how the background image is arranged in the Help pane.

Type	Applies To
Enum	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Value	Description
None	Single image, no repeat.
Tiled	Images repeat in a tiled pattern.
Horizontal	Images repeated horizontally.
Vertical	Images repeated vertically.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Appearance group in the **Properties Pane**.

BaseName Property

Base file name to be used for generating help target files.

Type	Applies To
String	Help Target

Remarks

By default, this name will be the same as the file name for the help project itself.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Behavior Property

Determines the help authoring behavior associated with a character style at compile time.

Type	Applies To
Enum	Character Style

Value	Description
None	No hot spot
ContextString	Context string hot spot
Group	Group hot spot
Topic Link	Topic link hot spot
Group Link	Group link hot spot
Keyword Link	Keyword link hot spot
Link Tag	Link tag hot spot

Value	Description
Context ID	Context ID hot spot
Conditional Text	Conditional text hot spot
Contents Title	Contents title hot spot
Topic Properties	Topic properties hot spot
Keyword	Keyword hot spot
Inline Expand	Inline expand hot spot
Inline Popup	Inline popup hot spot
Inline Text	Inline text hot spot
Inline Dropdown	Inline dropdown hot spot

Location

- Click the **Project Icon**.
- Select the **Character Styles** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

BinaryIndex Property

When **True**, generates a binary index instead of a site map index.

Type	Applies To
Boolean	Help Target (HTML Help)

Remarks

In modular HTMLHelp projects, BinaryIndex can have any value in the module projects, but must be set to **True** in the hub project.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

BinaryTOC Property

When **True**, generates a binary table of contents instead of a site map table of contents.

Type	Applies To
Boolean	Help Target (HTML Help)

Remarks

In modular HTMLHelp projects, BinaryTOC can have any value in the module projects, but must be set to **False** in the hub project.

Set this property to **True** if your main window uses the built-in Next and Previous buttons.

Location

- Click the **Project Icon**.

- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

ButtonLabel Property

For Help targets, specifies the text that precedes the subtopic buttons.

For topics, overrides the Help target ButtonLabel text on a topic by topic basis.

Type	Applies To
String	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp) Topic

Remarks

Set this property to an empty string to suppress the default **More:** label.

Location (Help Targets)

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

Location (Topics)

- Click the **Topics Icon**.
- Select the property you wish to modify from the upper right Pane.
- The property is located under the Display group in the **Properties Pane**.

ButtonLabelStyle Property

Lists the name of the style used to format the label that precedes the subtopic buttons.

Type	Applies To
Style	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ButtonSeparator Property

When enabled, a dividing line separates the topic text from the subtopic buttons.

Type	Applies To
Boolean	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ButtonStyle Property

Lists the name of the style used to format automatically generated subtopic buttons.

Type	Applies To
Style	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ByLine Property

Specifies the byline text used on the title page of printed manuals.

Type	Applies To
String	Help Target (Manual)

Location

- Click the Project Icon.
- Make sure you have selected the Manual help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Title Page group in the **Properties Pane**.

CancelKey Property

When **True**, this property disables the [AutoKeyword](#) property on a topic by topic basis.

Type	Applies To
Boolean	Topic

Location

- Click the **Topic Icon**.

- Make sure you have selected **All Topics**, **Topics With**, or **Topics Without** from the left pane Tree View.
- The property is located under the Behavior group in the **Properties Pane**.

CancelMap Property

When **True**, this property disables the [AutoContextID](#) property on a topic-by-topic basis.

Type	Applies To
Boolean	Topic

Location

- Click the **Topic Icon**.
- Make sure you have selected **All Topics**, **Topics With**, or **Topics Without** from the left pane Tree View.
- The property is located under the Behavior group in the **Properties Pane**.

CancelTag Property

When **True**, this property disables the [AutoLink](#) property on a topic by topic basis.

Type	Applies To
Boolean	Topic

Location

- Click the **Topic Icon**.
- Make sure you have selected **All Topics**, **Topics With**, or **Topics Without** from the left pane Tree View.
- The property is located under the Behavior group in the **Properties Pane**.

Caption Property

Represents the string to be displayed in the caption bar of the help window.

Type	Applies To
String	Help Target (Help 2.0) Window (HTML Help) Help Target (NetHelp) Window (WinHelp) Help Target (JavaHelp)

Remarks

Because of limitations imposed by the Windows help compiler, Winhelp captions are limited to 50 characters.

Location

- Click the **Project Icon**.
- Make sure you have selected a HTML Help or WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item in the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.
- or
- Make sure you have selected the JavaHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.

- The property is located under the Appearance group in the **Properties Pane**.

Charset Property

Specifies one of the character set aliases listed in HTML Character Sets and Named Entities.

Type	Applies To
String	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the Help 2.0, NetHelp, HTML Help or JavaHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Localization group in the **Properties Pane**.

Citation Property

Appends a citation to the end of any information (except in a context-sensitive popup window) that is copied from the Help file.

Type	Applies To
String	Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Code Property

When activated, this property produces an editable textbox containing the script code.

Type	Applies To
String	Script

Location

- Click the **Project Icon**.
- Select the **Scripts** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

CodePage Property

Defines the windows codepage number to be used by the help file.

Type	Applies To
Integer	Help Target

Remarks

This property is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Localization group in the **Properties Pane**.

ColorReduction Property

Determines the color threshold for images embedded within Word source documents.

Type	Applies To
Enum	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Value	Description
None	Images are not converted.
16 Colors	Images with more than 16 colors are reduced to 16.
256 Colors	Images with more than 256 colors are reduced to 256.

Remarks

Has no effect on linked images that are not saved with the document. Also, this property does not apply to HTML source documents.

Location

- Click the **Project Icon**.
- Make sure you have selected the correct Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

Comments Property

When activated, this property provides an editable textbox to be used for comments by the help author. These comments are not accessible by the end user.

Type	Applies To
String	Topic

Location

- Click the **Topics Icon**.
- The property is located under the Misc group in the **Properties Pane**.

Compression Property

Determines the level of file compression used by the WinHelp compiler.

Type	Applies To
Enum	Help Target (WinHelp)

Value	Description
None	No compression.
Choose best	The WinHelp compiler determines the best algorithm to use.
Phrase	For help files under 100K.
Hall	For help files that will be compressed again by another utility.
Zeck	Minimal compression for quick builds.
Zeck + Phrase	Minimal compression for help files under 100K.
Zeck + Hall	Maximum compression.

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Concatenate Property

When **True**, subdocuments are concatenated into a single master document in printed manual builds.

Type	Applies To
String	Help Target (Manual)

Remarks

Set this property to **False** to generate a master document that uses Word RD fields to reference subdocuments.

Location

- Click the **Project Icon**.
- Make sure you have selected the Manual Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

Condition Property

For a topic, defines a condition for including this topic in a help target.

For a character style, makes this style a conditional text style, places the specified condition on text formatted with this character style.

For a source document, defines a condition for including this document in a help target.

Type	Applies To
String	Topic Character Style Document

Remarks

Doc-To-Help provides a rich set of options to make certain text, whole topics, or entire documents appear only in certain help targets. You can specify conditions for inclusion in help targets on four levels:

Topic level: To include or exclude topics from certain help targets, specify this property for each topic you want to make conditional. By specifying a condition for a topic you include it in only those help targets that satisfy this condition. A condition can be a help target name, a platform (meaning all targets of the same kind) or an attribute (meaning help targets that have this attribute value defined).

Text level: You can make any text selection appear only in certain help targets by formatting that selection with a C1H Conditional style markup and assigning a condition to this particular text selection.

Style level: By assigning a condition to a character style you make all text formatted with that style appear only in help targets satisfying that condition. In that case you do not need to specify a condition every time you format a text with that style. D2HML includes two predefined character styles with conditions already defined: C1H Online for all online targets and C1H Manual for printed manual targets.

Document level: To include or exclude a document from certain help targets, specify this property for each document you want to make conditional. By specifying a condition for a document, you include it in only those help targets that satisfy this condition. A condition can be a help target name, a platform (meaning all targets of the same kind) or an attribute (meaning help targets that have this attribute value defined).

See the description of the Doc-To-Help conditional text features for details. See [Conditional Text Hot Spot](#) (page 273) in the D2HML description for the exact syntax of the condition string.

Location (Topic)

- Click the **Topics Icon**.
- Select **All Topics** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Character Style)

- Click the **Project Icon**.
- Select the **Character Styles** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

ConfidenceLevel Property

Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed.

Type	Applies To
Integer	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Remarks

Matching answers that have a lower score than the value of the ConfidenceLevel property will be discarded.

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.

- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

ContentsOnly Property

When **True**, omits topics defined with this paragraph style or topic type from the help target, but uses them as book titles in the help contents.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ContentsTab Property

When enabled, the Contents tab is added to the help window.

Type	Applies To
Boolean	Help Target (NetHelp) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.

ContentsTitle Property

Allows for modification of a topic title in the help window contents area without changing the actual topic title in the source document.

Type	Applies To
String	Topic

Location

- Click the **Topics Icon**.
- The property is located under the Display group in the **Properties Pane**.

ContextString Property

When using context-sensitive and dynamic help with Microsoft Help 2.0, specifies a context string for the topic.

Type	Applies To
String	Topic

Remarks

The ContextString property applies to topics in the MS Help 2.0 target. It is used to build a context-sensitive index that is used in Microsoft Visual Studio .NET for context-sensitive help (F1) and dynamic help.

Each topic can have one or more context strings, or none. The ContextString property contains all context strings of the given topic separated by semicolon. Topic context strings form a hierarchical tree structure. A context string consists of dot-separated context names for each hierarchy level.

For example, a topic "Property MyProperty" can have:

ContextString = "MyCompany.MyProduct.MyProperty"

Topic "Properties MyProperty1 and MyProperty2" can have:

ContextString = "MyCompany.MyProduct.MyProperty1;MyCompany.MyProduct.MyProperty2"

Location

- Click the **Topics Icon**.
- The property is located under the Display group in the **Properties Pane**.

CopyFolder Property

Defines the folder name where ancillary build files will reside.

Type	Applies To
String	Help Target

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Copyright Property

Places a custom copyright notice in the Version dialog box of WinHelp.

Type	Applies To
String	Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

DefaultFile Property

File name to be used as part of the base URL for the NetHelp target.

Type	Applies To
String	Help Target (NetHelp)

Remarks

Typically, the filename will be *default.htm* or *index.html*.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the HTML group in the **Properties Pane**.

DefaultTemplate Property

Specifies the document template that will be attached to new documents created in the Project Editor.

Type	Applies To
Template	Project

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

DefaultWindow Property

Specifies the default window definition for the compiled help file.

Type	Applies To
Window	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

DisplayTitle Property

Allows for modification of a topic title with respect to help file searches without changing the actual topic title in the source document.

Type	Applies To
String	Topic

Remarks

Use this property to add qualifying text to like-named topics. For example, a search for “intro” may yield several topics named Introduction, but by modifying the DisplayTitle, you can get results such as Introduction (Help Authoring), Introduction (HTML), Introduction (WinHelp) without adding the text in parentheses to the source documents.

This is the equivalent of adding a \$ footnote in WinHelp or a <Title> in HTML.

Location

- Click the **Topics Icon**.
- The property is located under the Display group in the **Properties Pane**.

DocumentType Property

Determines the type(s) of documents that can be used in the project.

Type	Applies To
Enum	Project Settings

Value	Description
MS Word	Microsoft Word documents can be used.
HTML	HTML documents can be used.
Mixed	Both Word and HTML documents can be used in the project.

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

DynamicTOC Property

Generates a Dynamic HTML version of the contents page with an expandable outline.

Type	Applies To
Boolean	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the HTML group in the **Properties Pane**.

EmailAddress Property

Allows modification of the default email address in the email created by the Email button in NetHelp.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is *support@mycompany.com*.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.

- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

ExplicitAccess Property

When **True**, defines a specific paragraph style or topic type as accessible only through a hyperlink. Paragraph styles and topic types defined as explicit are not accessible from the help contents, the index list or the text search.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

FavoritesTab Property

When enabled, the Favorites tab is added to the help window.

Type	Applies To
Boolean	Window (HTML Help) Help Target (JavaHelp) Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.
- or
- Make sure you have selected the JavaHelp or NetHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.

FixLists Property

Controls how numbered lists are corrected in the printed manual output when the [Concatenate](#) property is set to **True**.

Type	Applies To
Boolean	Help Target (Manual)

Value	Description
Never	Never corrects the numbered lists.
Always	Always corrects the numbered lists.
Word 2000 Only (Default)	Corrects lists with Word 2000, but not with other versions of Word.

Remarks

Because of the sophisticated manner in which Word 2000 tries to make list numbering intuitive, there are known issues with preserving numbering when performing copy/paste and other text manipulations. These issues are largely solved in Word 2002 (XP).

The default setting (Word 2000 Only), allows Doc-To-Help to automatically fix, re-number and re-format lists where it detects problems in the printed manual output.

On very rare occasions, attempts to detect and fix list problems can result in problems during Doc-To-Help builds. If you encounter such problems while the Doc-To-Help progress indicator reads "Restoring lists formatting...", you may want to skip the fix.

Set the FixLists property to Never when you want to skip the detect and fix functions.

Set the FixLists property to Always when you want to perform the detect and fix functions in all versions of Word.

Location

- Click the Project Icon.
- Make sure you have selected the Manual help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

Folder Property

Defines the folder name where the help files will reside.

Type	Applies To
String	Help Target

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Frameset Property

Generates a frameset version of the HTML help project with the left frame displaying the contents and the right frame displaying the help topics.

Type	Applies To
Boolean	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the HTML group in the **Properties Pane**.

GenerateCSS Property

When set to **False**, the existing cascading style sheet will not be overwritten when building the Help target.

Type	Applies To
Boolean	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- Select the Help target you wish to modify.
- The property is located under the Build group in the **Properties Pane**.

GeneratePDF Property

Enables generation of a PDF in the Manual target build.

Type	Applies To
String	Help Target (Manual)

Remarks

The default value of this property is *False*.

Location

- Click the **Project Icon**.
- Make sure you have selected the Manual target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

GenerateProjectFile Property

When set to **False**, the existing .hlp or .hpl files will not be overwritten when building the Help target.

Type	Applies To
Boolean	Help Target (HTML Help) Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- Select the Help target you wish to modify.
- The property is located under the Build group in the **Properties Pane**.

Global Property

When **True**, \$global_ is prepended to the window name in generated files and <OBJECT> tags.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

Use this property in modular help projects to ensure that like-named secondary window references in child help files will resolve to the same physical window.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Glossary Property

Specifies the document that contains glossary terms for the project.

Type	Applies To
Document	Project Settings

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

GraphicsScaleWithFonts Property

When set to **True**, all graphics in Word source documents are resized proportionally when system font size is changed.

Type	Applies To
Boolean	Help Target (Help 2.0) Help Target (HTML Help) Help Target (NetHelp)

Remarks

When system font size is changed, for example, in Windows XP, by selecting Large fonts in the Font size combo box, all text in the system is scaled accordingly. If you want all graphics in your help to scale proportionally with the fonts, set this property to **True**.

This property does not apply to HTML source documents.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

Height Property

Determines the height of the help window.

Type	Applies To
Integer	Window (HTML Help) Window (WinHelp) Window (JavaHelp) Window (NetHelp)

Location

- Click the **Project Icon**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

Hidden Property

Any text formatted with a paragraph or character style with Hidden property set to **True** is omitted from the help target.

Type	Applies To
Boolean	Character Style Paragraph Style

Remarks

If a topic heading is formatted with this style, the whole topic is omitted from the help target.

Location

- Click the **Project Icon**.
- Select **Character Style** or **Paragraph Styles** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

HideNavigationPane Property

When **True**, the navigation pane for the help file is hidden by default.

Type	Applies To
Boolean	Window (HTML Help) Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.
- or
- Make sure you have selected the NetHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

HTMLDefaultCSS Property

Specifies the cascading style sheet attached to each source HTML document when it is added to the project.

Type	Applies To
StyleSheet	Project Settings

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

HTMLName Property

Defines the name to be used to identify the style as a cascading style sheet style in HTML source documents.

Type	Applies To
String	Character Styles Paragraph Styles

Remarks

Every special-meaning style specified in a Doc-To-Help project has two names: (1) the Style.[Name](#) property that identifies it as a Word style or a Style object in Doc-To-Help if the project does not use Word, and (2) the Style.**HTMLName** property that identifies it as a CSS style, or selector.

When you use a style in an HTML document and that style is defined in your Doc-To-Help project, make sure you use the HTML name of the style specified in the Style.HTMLName property, not the Style.Name property.

According to the standard CSS rules, HTMLName can have one of three forms:

1. `<stylename>` - The style can be used with any HTML tag.
2. `<tag>` - The specified tag, for example, H1 for style Heading 1, is considered by Doc-To-Help as having this style, even if the user did not format this tag with any particular CSS style.
3. `<tag>.<stylename>` - The style can be used only with the specified tag; it is ignored if used in other tags.

Location

- Click the **Project Icon**.
- Select the **Paragraph Styles** or **Character Styles** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

HTMLSourceFolder Property

Defines the folder name where the HTML source documents will reside.

Type	Applies To
String	Project Settings

Remarks

The default folder name is HTMLDocuments. This folder is automatically created when you create a project.

Although you can add any HTML documents to the project even if they are not located in this folder or any of its subfolders, it is strongly recommended to keep all HTML source documents inside this folder, both because it helps to keep the project files organized and because doing this will ensure that any links to files of any kind (images, multimedia, other HTML files) that you may have in this document are preserved in the help target provided that those files also reside inside this folder or one of its subfolders. If either the source file or a file it links are placed outside this folder, the links can be broken in the help target because Doc-To-Help does not copy outside files to the target folder.

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

HubContents Property

Displays the name of the contents (.cnt) file to associate with the Help file.

Type	Applies To
String	Help Target (WinHelp)

Remarks

For modular WinHelp systems, use this property to associate a component Help file with its hub.

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- The property is located under the Misc group in the **Properties Pane**.

ID Property

Returns a unique numeric identifier assigned to the topic during compilation (read-only).

Type	Applies To
Long	Topic

Location

- Click the **Topics Icon**.
- The property is located under the Misc group in the **Properties Pane**.

IgnoreFontNames Property

When set to **True**, font names specified in the source document are ignored. Default font names are used in the help target.

Type	Applies To
Boolean	Help Target (JavaHelp)

Remarks

Default: **True**.

This property is used in conjunction with IgnoreFontSizes.

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Fonts group in the **Properties Pane**.

IgnoreFontSizes Property

When set to **True**, font sizes specified in the source document are ignored. Default font sizes are used in the help target.

Type	Applies To
Boolean	Help Target (JavaHelp)

Remarks

Default: **True**.

This property is used in conjunction with IgnoreFontNames. If this property is set to **False**, the ScaleFontSizes property takes effect.

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Fonts group in the **Properties Pane**.

ImageFormat Property

Determines graphic format in which images originated from Word source documents are stored in image files in the online help target.

Type	Applies To
Enum	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Value	Description
As is	All graphics are stored in the original format without conversion.
Always JPG	All graphics are stored in JPG files.
JPG if transformed	Graphics are converted to JPG if they undergo any transformations in building the help target.
Always PNG	All graphics are stored in PNG files.
PNG if transformed	Graphics are converted to PNG if they undergo any transformations in building the help target.

Remarks

This property does not apply to HTML source documents.

Location

- Click the **Project Icon**.
- Make sure you have selected the correct help target from the **Help Target Dropdown**.
- The property is located under the Display group in the **Properties Pane**.

IndexTab Property

When enabled, the Index tab is added to the help window.

Type	Applies To
Boolean	Help Target (NetHelp) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the correct help target from the **Help Target Dropdown**.
- The property is located under the Tabs group in the **Properties Pane**.

Jump1Caption Property

Enter the caption for the Jump1 button here.

Type	Applies To
String	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

Jump1URL Property

Enter the URL of the HTML file for the Jump1 button here.

Type	Applies To
String	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

Jump2Caption Property

Enter the caption for the Jump2 button here.

Type	Applies To
String	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

Jump2URL Property

Enter the URL of the HTML file for the Jump2 button here.

Type	Applies To
String	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

KeepOutlineNumbers Property

When enabled, outline numbers are included as part of topic titles and are present in help targets.

Type	Applies To
Boolean	Document Project Settings

Remarks

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

This property only affects the RTF files generated from source documents; it does not affect the source documents themselves.

Location (Documents)

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Document Defaults group in the **Properties Pane**.

KeepPageBreaks Property

When **True**, this property retains the page break characters in the source documents during compilation of a Printed Manual Help target. Set this property to **False** to discard page break characters.

Type	Applies To
Boolean	Project Settings

Remarks

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Document Defaults group in the **Properties Pane**.

Left Property

Determines the location of the left edge of the help window.

Type	Applies To
Integer	Window (HTML Help) Window (WinHelp) Window (JavaHelp) Window (NetHelp)

Remarks

In conjunction with the [Top](#) property, determines the position of the help window on the screen.

Location

- Click the **Project Icon**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

Level Property

Returns the level of a document in the document tree.

Type	Applies To
Integer	Document

Remarks

Multiple topic documents always appear on Level 1. Only single-topic HTML documents can be placed on deeper levels of the tree hierarchy.

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

LinkType Property

Determines whether a link hot spot is a jump or a pop-up.

Type	Applies To
Enum	Character Styles

Value	Description
Jump	Topic jump hot spot.
Popup	Topic pop-up hot spot.

Remarks

This property is applicable only to character styles defining links, that is, having their **Behavior** property equal to Topic Link. The value of this property determines whether the linked topic is shown in the regular help window (Jump) or in a pop-up window (Popup).

Location

- Click the **Project Icon**.
- Select the **Character Styles** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

LiveLinks Property

Determines whether links are live hyperlinks in the manual Word document.

Type	Applies To
Boolean	Help Target (Manual)

Remarks

The default value of this property is **True**, meaning all topic links, regardless of the way they are specified in the source document, become Word hyperlinks in the manual document. When the document is converted to a PDF, the hyperlinks are live PDF links.

If this property is set to **False**, only cross-references are live links in the manual Word document.

The [Concatenate](#) property must be set to **True** in order for the **LiveLinks** property to take effect.

There is an issue with the Adobe Acrobat PDF converter when *Enable accessibility and reflow with Tagged PDF* is checked in Adobe Acrobat, including the latest version 7. When the PDF is produced and the Word hyperlinks are converted to PDF links, the links jump to the top of the page containing the destination topic instead of jumping to the desired topic location. To fix this problem, uncheck the *Enable accessibility and reflow with Tagged PDF* checkbox in Adobe Acrobat before creating a PDF or use a PDF converter other than Adobe Acrobat.

Location

- Click the **Project Icon**.
- Make sure you have selected the **Manual** help target from the **Help Target Dropdown**.
- The property is located under the Display group in the **Properties Pane**.

Locale Property

This property is used to specify the language in the resulting target files. Each target has its own way of specifying language. For example, HTML Help has a Language option in the [OPTIONS] section of the project file (.hhp).

Type	Applies To
Integer	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp)

Remarks

Some common values for this property are as follows:

English (U.S.)	1033
English (U.K.)	2057
French	1036
German	1031
Italian	1040
Spanish	1034

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- The property is located under the Localization group in the **Properties Pane**.

MapFileForC Property

Creates C or C++ header files for context sensitive help.

Type	Applies To
Boolean	Help Target (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

MapFileForVB Property

Creates Visual Basic header files for context sensitive help.

Type	Applies To
Boolean	Help Target (HTML Help) Help Target (WinHelp)

Location

- Click the **Project Icon**.

- Make sure you have selected the HTML Help or the WinHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

MapNumberOffset Property

The value of this property is added to the automatically generated map numbers to prevent numbering conflicts in modular Help systems.

Type	Applies To
Integer	Project Settings

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

MapNumbers Property

Determines whether a unique Context ID (map number) is generated for each topic.

Type	Applies To
Enum	Help Target (HTML Help) Help Target (WinHelp)

Value	Description
Automatic	Map numbers are generated automatically for each topic.
Automatic+Custom	Map numbers are generated automatically for each topic, but can be customized as required.
Custom	No map numbers are generated automatically.

Remarks

For the Context IDs to be generated, the **AutoContextID** property must be set to **True** for each relevant Topic Type or Paragraph Style.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or WinHelp targets from the **Help Target Dropdown**.
- Select the **Help Targets** item in the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Maximized Property

If **True**, the help window is automatically maximized when displayed.

Type	Applies To
Boolean	Window (HTML Help) Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or WinHelp targets from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

MergeContents Property

For a placeholder topic in a modular hub project, this property specifies the platform-dependent contents file to be loaded dynamically.

Type	Applies To
String	Topic

Remarks

This property only applies to modular hub projects for the WinHelp and HTML Help platforms.

Location

- Click the **Topics Icon**.
- The property is located under the Modular group in the **Properties Pane**.

MergeFile Property

For a placeholder topic in a modular hub project, this property specifies the platform-dependent help file to be loaded dynamically.

Type	Applies To
String	Topic

Remarks

This property only applies to modular hub projects for the WinHelp, HTML Help and NetHelp platforms.

Location

- Click the **Topics Icon**.
- The property is located under the Modular group in the **Properties Pane**.

MergeTitle Property

For a placeholder topic in a modular hub project, this property specifies the text used to disambiguate like-named topics in keyword search lists.

Type	Applies To
String	Topic

Remarks

This property only applies to modular hub projects for the WinHelp platform. If not specified, the value of the MergeFile property is used.

Location

- Click the **Topics Icon**.

- The property is located under the Modular group in the **Properties Pane**.

MidTopic Property

Defines a paragraph style or a topic type as an “in topic” jump location similar to an HTML anchor. By defining a paragraph style or topic type as MidTopic and defining a character style as a jump, end users will have jump functionality within a lengthy topic.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

Modified Property

Returns the date and time of the last saved modification in the source document.

Type	Applies To
DateTime	Document

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

ModularHub Property

If **True**, the generated help file can dynamically load the contents of other help files, if present.

Type	Applies To
Boolean	Project

Remarks

This property only applies to the WinHelp, HTML Help and NetHelp platforms.

For WinHelp and HTML Help targets, to specify a component help file in a modular hub project, create a placeholder topic, then set the MergeFile and MergeContents properties of that topic to the component filenames. When testing your project, you will need to copy the component files into the output folder of each modular help target.

For NetHelp targets, to specify a component help file in a modular hub project, create a placeholder topic, then set the MergeFile property of that topic to the component filename.

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

MultiLink Property

When **False**, only the first occurrence of a "jump" character style (in a given topic) generates a link. All other formatted occurrences are skipped.

Type	Applies To
Boolean	Character Style

Location

- Click the **Project Icon**.
- Select the **Character Style** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

Name Property

Returns a string containing the name of a specified object (read-only, except for a help target).

Type	Applies To
String	Attribute Character Style Document Group Help Target Keyword Paragraph Style Script Template Topic Type Window

Remarks

This property is modifiable (not read-only) for a Help Target. Changing this property the user can rename a help target.

Location

- Click the **Project Icon**.
- Select any of the items in the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Namespace Property

Specifies the Help 2.0 namespace that is used to register the Help file after a successful build.

Type	Applies To
String	Help Target (Help 2.0)

Remarks

If the Namespace property is empty, the [BaseName](#) property value is used. If [BaseName](#) is empty, the file name of the project itself is used.

NOTE: The Namespace and [ParentNamespace](#) properties only affect Help file registration on the author's machine. Registration on the user machine is handled by standard Help 2.0 means, with Windows Installer.

Location

- Click the Project Icon.

- Make sure you have selected the Help 2.0 help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

NavigationPaneWidth Property

Defines the width of the navigation pane of the help file in pixels.

Type	Applies To
Integer	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

Nonscrolling Property

When the Nonscrolling property of a paragraph style or topic type is set to **True**, any topic headings formatted with that paragraph style or topic type will appear in the non-scrolling region of the help window.

Type	Applies To
Boolean	Paragraph Style Topic Type

Remarks

This option only applies to WinHelp builds.

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

NonscrollingColor Property

Displays the current color setting for the non-scrolling help window. To change the color, click the adjacent change button to open a standard color selection dialog.

Type	Applies To
Long	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Appearance group in the **Properties Pane**.

OneBrowseSequence Property

When **True**, this property provides a continuous browse sequence that spans multiple source documents.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **HelpTargets** item from the left pane Tree View.
- The property is located under the Navigation group in the **Properties Pane**.

OnlineOnly Property

When **True**, omits paragraphs formatted with this style from the printed manual target only.

Type	Applies To
Boolean	Paragraph Style

Remarks

This property has no effect in online Help targets.

Location

- Click the **Project Icon**.
- Select the **Paragraph Styles** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

OnPage Property

Specifies the default text used when completing a cross-reference.

Type	Applies To
String	Project

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Localization group in the **Properties Pane**.

Order Property

Determines the order of the documents in the help contents.

Type	Applies To
Integer	Document

Location

- Click the **Project Icon**.

- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

OutlineInPDF Property

Indicates whether an outline, or PDF bookmarks found in the TOC pane on the left side of the PDF viewer, is generated in the PDF when a PDF is generated in the Manual target build.

Type	Applies To
String	Help Target (Manual)

Remarks

The default value of this property is *True*.

Location

- Click the **Project Icon**.
- Make sure you have selected the Manual target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Build group in the **Properties Pane**.

OwnTOC Property

If this property is set to **True** for a help target, that target's table of contents is customizable independently of other targets.

Type	Applies To
Boolean	Help Target (Help 2.0) Help Target (HTML Help) Help Target (JavaHelp) Help Target (Manual) Help Target (NetHelp) Help Target (WinHelp)

Remarks

This property determines whether TOC customization performed on the Contents tab applies only to the current help target or to other targets as well. TOC customization for a target with OwnTOC=**False** affects all targets with OwnTOC=**False**. In other words, all help targets with OwnTOC=**False** always have the same table of contents. Customizing TOC for a help target with OwnTOC=**True**, you are customizing it for the current help target only, other targets' tables of contents are not changed.

Setting OwnTOC to **True** is useful in cases where you need to have different TOC for different help targets. For example, you might want to change the order of topics in a certain target without affecting other targets, or you might want to duplicate some topics in a certain target without duplicating them in other targets.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

PageRef Property

When producing a manual, this property places the page number of the referenced topic next to the text formatted with this style.

Type	Applies To
Boolean	Character Style

Remarks

This option is only available for Jump and Popup links.

Location

- Click the **Project Icon**.
- Select either the **Character Style** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

Parent Property

Shows the ID number of the related parent topic.

Type	Applies To
Long	Topic

Location

- Click the **Topics Icon**.
- The property is located under the Misc group in the **Properties Pane**.

ParentNamespace Property

Specifies the namespace for the built help collection to plug into. Usually, that will be the namespace of the Visual Studio .NET Combined Collection. Set this property if you want the built Help to be automatically registered as a plugin for Visual Studio help. For example, for Visual Studio .NET 2003, set ParentNamespace to "MS.VSCC.2003".

Type	Applies To
String	Help Target (Help 2.0)

Remarks

NOTE: The Namespace and ParentNamespace properties only affect Help file registration on the author's machine. Registration on the user machine is handled by standard Help 2.0 means, with Windows Installer.

Location

- Click the Project Icon.
- Make sure you have selected the Help 2.0 help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

Popup Property

When the **Popup** property of a paragraph style or topic type is set to **True**, any topic formatted with that paragraph style or topic type will appear in a pop-up window on top of the help window.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

PrefixIndexListFile Property

Allows modification of the default prefix for topic list files corresponding to the main index.

Type	Applies To
String	Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

PrefixTopicListFile Property

Allows modification of the default prefix for topic list files corresponding to keyword hot spots.

Type	Applies To
String	Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

RelatedJumps Property

Controls the order of automatic and/or custom jumps on a topic by topic basis.

Type	Applies To
Enum	Topic

Value	Description
None	No Jumps are displayed
Automatic	Only Automatic jumps are displayed
Automatic + Custom	Automatic then Custom jumps are displayed
Custom + Automatic	Custom then Automatic jumps are displayed
Custom	Only Custom jumps are displayed

Location

- Click the **Topic Icon**.
- The property is located under the Navigation group in the **Properties Pane**.

Replacement Property

In certain cases, returns a string that overrides the generated default text.

Type	Applies To
String	Character Style Paragraph Style Topic Type

Remarks

For paragraph styles and topic types with [AutoNext](#) set to **True**, this string is used instead of the topic title next to the generated button.

For character styles with [PageRef](#) set to **True**, this string specifies the format of page number references in printed manual targets. This string is ignored unless it contains a pound sign, which is replaced with a PAGEREF field.

Location

- Click the **Project Icon**.
- Select either the **Character Style**, **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

SaveUserPosition Property

When **True**, stores the size and position of the help window in the registry when the user modifies it.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

ScaleFontSizes Property

If IgnoreFontSizes = **False**, this property is used to scale all font sizes in the help target.

Type	Applies To
Real	Help Target (JavaHelp)

Remarks

This property is used in conjunction with IgnoreFontSizes property. If IgnoreFontSizes = **True**, this property is ignored.

JavaHelp has a known problem that makes all fonts appear smaller (approximately 1.3 times smaller) than they should be. By setting IgnoreFontSizes property to **True**, you will have consistent default JavaHelp font sizes. Alternatively, you can set the ScaleFontSizes property to a real number other than default 1.0 that will scale all fonts to adjust their sizes. For example, if JavaHelp displays fonts 1.3 times smaller that they should be, you can set ScaleFontSizes to 1.3 to fix that.

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Fonts group in the **Properties Pane**.

ScaleGraphics Property

Determines what scaling is applied to the graphics in Word source document.

Type	Applies To
Enum	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp) Help Target (WinHelp)

Value	Description
No scale	Graphics are displayed at their original size. Scale that the graphics may have in the source Word document is reset.
Scale in target	Graphics have the same size as they are displayed in Word (where they may be scaled). Linked graphics are scaled by the viewer (browser) displaying the online help target. The linked image file has the original size, scaling is delegated to the browser by setting the IMG tag dimensions.
Scale in build	Graphics have the same size as they are displayed in Word (where they may be scaled). Linked graphics are scaled by Doc-To-Help when the help target is built, the linked image file is resampled to the required size.

Remarks

With **No scale** option, the size of the picture in the help target is different from the size displayed in Word if the picture has scale in the Word document.

For graphics embedded within Word documents (as opposed to linked graphics that are not stored in the document), **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in Word. If the picture is scaled in Word, Doc-To-Help gets the image file already scaled from Word.

For linked graphics, options **Scale in target** and **Scale in build** also produce pictures of the same size, the size it is displayed in Word. The only difference between them is that quality of scaling (resampling) can differ for some images.

With **Scale in build** option, Doc-To-Help is responsible for resampling the image, and with **Scale in target** option that task is left to the viewer (browser) that displays the online help target. Choose whichever produces best results.

For WinHelp target, **Scale in target** and **Scale in build** options are equivalent for all graphics, resampling is performed by Doc-To-Help.

This property does not apply to HTML source documents.

Location

- Click the **Project Icon**.
- Make sure you have selected the correct help target from the **Help Target Dropdown**.
- The property is located under the Display group in the **Properties Pane**.

Script Property

Displays the script to be run during compilation whenever text formatted with this style is encountered.

Type	Applies To
Script	Character Style Paragraph Style Topic Type

Remarks

Will show None if no script is associated with the style.

Location

- Click the **Project Icon**.
- Select either the **Character Style**, **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

SearchCaption Property

Specifies the text added to the tab caption (HTML Help) or button caption (WinHelp) when ComponentOne Natural Search is enabled.

Type	Applies To
String	Help Target (HTML Help) Help Target (WinHelp)

Remarks

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

SearchEnabled Property

Controls whether ComponentOne Natural Search is enabled for a Help target, document, or topic.

Type	Applies To
Boolean	Document Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp) Topic

HTML Help or WinHelp

Set this property to **True** to generate Natural Search indexes for a Help target and add a custom tab (HTML Help) or button (WinHelp) to the Help file.

NetHelp

Set this property to **True** to generate Natural Search indexes for a Help target. Note that the Help target itself will not have a Natural Search interface. This option is provided solely for importing HTML content and indexes into **ComponentOne Response**.

Document

Set this property to **False** to exclude an individual document from Natural Search processing.

Topic

Set this property to **False** to exclude an individual topic from Natural Search processing.

Remarks

This property is only available in Doc-To-Help Enterprise.

Location (Help Targets)

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

Location (Documents)

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- Select the appropriate document from the right pane.
- The property is located under the Natural Search group in the **Properties Pane**.

Location (Topics)

- Click the **Topics Icon**.
- Select the appropriate topic from the right pane.
- The property is located under the Natural Search group in the **Properties Pane**.

SearchStopList Property

Specifies the path to a text file containing noise words.

Type	Applies To
String	Help Target (NetHelp)

Remarks

This property is empty by default; therefore, the default noise word list is used for searches. If you specify a noise word list using the **SearchStopList** property, the default list is overridden. In this case, you must add the standard noise words to your list if you want them to be ignored in the search. Each noise word must appear on a separate line in the file.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- The property is located under the Misc group in the **Properties Pane**.

SearchTab Property

When enabled, the Search tab is added to the help window.

Type	Applies To
Boolean	Window (HTML Help) Help Target (NetHelp) Help Target (Java Help)

Location (Window)

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.

Location (Help Target)

- Click the **Project Icon**.
- Make sure you have selected the NetHelp or JavaHelp target from the **Help Target Dropdown**.
- Select the **HelpTargets** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.

SearchTabAdvanced Property

Adds additional functionality to the Search tab.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

Only available when the [SearchTab](#) property is **True**.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Tabs group in the **Properties Pane**.

ShowBackButton Property

When selected, the Back button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help) Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or the WinHelp targets from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowBrowseButtons Property

When selected, the Browse buttons are included in the help window.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowContentsButton Property

When selected, the Contents button is included in the help window.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowDropdownText Property

If set to **True**, drop-down text is shown in the help target, otherwise it is removed.

Type	Applies To
Boolean	Help Target (JavaHelp) Help Target (Manual) Help Target (Winhelp)

Remarks

Dropdown text is a Dynamic HTML effect that some help targets do not support, so they can only be either permanently shown in such targets or permanently omitted.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ShowExpandText Property

If set to **True**, expanding text is shown in the help target, otherwise it is removed.

Type	Applies To
Boolean	Help Target (JavaHelp) Help Target (Manual) Help Target (Winhelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

ShowFindButton Property

When selected, the Find button is included in the help window.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowForwardButton Property

When selected, the Forward button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.

- The property is located under the Buttons group in the **Properties Pane**.

ShowHelpTopicsButton Property

When selected, the Help Topics button is included in the help window.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowHideShowButton Property

When selected, the Hide and Show buttons are included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowHomeButton Property

When selected, the Home button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowIndexButton Property

When selected, the Index button is included in the help window.

Type	Applies To
Boolean	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowJump1Button Property

When selected, the Jump1 button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

Choose the caption and URL for this button with the [Jump1Caption](#) and [Jump1URL](#) properties.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowJump2Button Property

When selected, the Jump2 button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

Choose the caption and URL for this button with the [Jump2Caption](#) and [Jump2URL](#) properties.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowLocateButton Property

When selected, the Locate button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.

- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowNextButton Property

When selected, the Next button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

The **BinaryTOC** property must be set to **True** for the navigation buttons to function.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowOptionsButton Property

When selected, the Options button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help) Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowPreviousButton Property

When selected, the Previous button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Remarks

The **BinaryTOC** property must be set to **True** for the navigation buttons to function.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.

- The property is located under the Buttons group in the **Properties Pane**.

ShowPrintButton Property

When selected, the Print button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help) Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowRefreshButton Property

When selected, the Refresh button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

ShowStopButton Property

When selected, the Stop button is included in the help window.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Buttons group in the **Properties Pane**.

SingleTopic Property

Determines whether the source document can contain multiple topics or just one topic.

Type	Applies To
Boolean	Document

Remarks

Word documents are always multi-topic. HTML documents can be either multi-topic or single-topic.

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Size Property

Returns the size of the source document.

Type	Applies To
Integer	Document

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

SkipGlossary Property

When **True**, omits the Glossary topic from the generated Help contents.

Type	Applies To
Boolean	Help Target (HTML Help) Help Target (WinHelp)

Remarks

For each component Help file in a modular system, set this property to **True** to eliminate duplicate Glossary topics when viewing the hub.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help or the WinHelp targets from the **Help Target Dropdown**.
- The property is located under the Misc group in the **Properties Pane**.

SplitFile Property

Controls whether a document is split into separate topic files for HTML-based platforms.

Type	Applies To
Boolean	Document (Help 2.0) Document (NetHelp) Document (HTML Help) Project Settings

Help 2.0 or NetHelp

When **True**, generates a separate HTML file for each topic. When **False**, generates a single HTML file containing a named anchor for each topic.

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

HTML Help

When the `SplitFile` and the related [SplitFileWithHHCtrl](#) properties are **True**, a single HTML file is generated using intrinsic HTML Help commands to give the appearance of separate HTML files for each topic. When the `SplitFile` property is **True** and the `SplitFileWithHHCtrl` property is **False**, a separate HTML file is generated for each topic in the document.

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

Location (Documents)

- Click the **Project Icon**.
- Make sure you have selected the Help 2.0, the NetHelp, or the HTML Help target from the **Help Target Dropdown**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Document Defaults group in the **Properties Pane**.

SplitFileWithHHCtrl Property

When **True**, a single HTML file is generated using intrinsic HTML Help commands to give the appearance of separate HTML files for each topic in the document. When **False**, individual HTML files are generated for each topic in the document.

Type	Applies To
Boolean	Document (HTML Help) Project Settings

Remarks

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

The [SplitFile](#) property must be **True** for this property to be functional.

Location (Documents)

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.

- The property is located under the Document Defaults group in the **Properties Pane**.

StemPercent Property

Determines the number of characters (as a percentage of the total) that are added to the root word to create a new word group when building natural language search indexes.

Type	Applies To
Integer	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Remarks

The default value for this property is 33. That is, 1 character is added to the root word for every 3 additional characters in the word. Some international languages, where lengthy suffixes are common, may benefit from a higher percentage.

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

StemSize Property

Determines the maximum number of characters used as a root word when building natural language search indexes.

Type	Applies To
Integer	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Remarks

The default value of this property is 5. Highly technical documentation, where longer search terms are common, may benefit from a longer stem size.

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

StringContents Property

Allows modification of the **Contents** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (JavaHelp) Help Target (Manual)

Remarks

For printed manuals, this property is used only if the [Template](#) property is empty or the [UseTemplateText](#) property is set to **False**. Otherwise, the string from the template text is used.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringEmail Property

Allows modification of the **Email** string value.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Email**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFavorites Property

Allows modification to the **Favorites** string value.

Type	Applies To
String	Help Target (JavaHelp) Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the JavaHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFavoritesAdd Property

Allows modification of the **Add to Favorites** string value.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Add to Favorites**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFavoritesCaption Property

Allows modification of the **Favorites** string value on the **Favorites** tab of the NetHelp navigation pane.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Favorites**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFavoritesDelete Property

Allows modification of the **Delete** string value on the Favorites tab of the NetHelp navigation pane.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Delete**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFoundMany Property

Allows modification of the **%d topics found:** string value.

Type	Applies To
String	Help Target (JavaHelp)

Location

- Click the **Project Icon**.

- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFoundOne Property

Allows modification of the **1 topic found:** string value.

Type	Applies To
String	Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringFoundZero Property

Allows modification of the **No topics found.** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringHideNavigationPane Property

Allows modification of the **Hide Navigation Pane** string value.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Hide Navigation Pane**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringIndex Property

Allows modification of the **Index** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (JavaHelp) Help Target (Manual)

Remarks

For printed manuals, this property is used only if the [Template](#) property is empty or the [UseTemplateText](#) property is set to **False**. Otherwise, the string from the template text is used.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringIndexCaption Property

Represents the string to be displayed in the Index window above the index text box.

Type	Applies To
String	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringNext Property

Allows modification of the **Next** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringNoMatch Property

Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Type	Applies To
String	Help Target (NetHelp) Help Target (HTML Help) Help Target (WinHelp)

Remarks

This property is only available in Doc-To-Help Enterprise.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Natural Search group in the **Properties Pane**.

StringPrevious Property

Allows modification of the **Previous** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringPrint Property

Allows modification of the **Print** string value.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Print**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringReturnToIndex Property

Allows modification of the **Return to Index** string value.

Type	Applies To
String	Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.

- The property is located under the Strings group in the **Properties Pane**.

StringSearch Property

Allows modification of the **Search** string value.

Type	Applies To
String	Help Target (NetHelp) Help Target (JavaHelp)

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringSearchCaption Property

Represents the string to be displayed in the Search window above the search text box.

Type	Applies To
String	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringSearchGo Property

Represents the string to be displayed in the search button within the Search window.

Type	Applies To
String	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

StringSearchHighlight Property

Defines the text to be used for the checkbox controlling highlighting search phrases.

Type	Applies To
String	Help Target (NetHelp)

Remarks

The default value of this property is **Highlight search hits**.

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- The property is located under the Strings group in the **Properties Pane**.

StringSyncTOC Property

Allows modification of the **Sync TOC** string value.

Type	Applies To
String	Help Target (NetHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the NetHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Strings group in the **Properties Pane**.

Style Property

Determines the style of the topic contained in a single-topic document.

Type	Applies To
ParagraphStyle	Document

Remarks

This property applies only to single-topic HTML source documents. Multi-topic documents don't need it because their topics are defined by the styles of their headings in the document. With a single-topic document, there is no header to derive the title, TOC position and other properties of the topic, so that information has to be specified explicitly in the Style and Title properties of the document. This is done by the user on adding the document to the project in Adding HTML Documents or New HTML Document dialog box.

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

StyleSheet Property

Specifies the name of the cascading style sheet applied to intermediate files during compilation.

Type	Applies To
String	Help Target (Help 2.0) Help Target (HTML Help) Help Target (JavaHelp) Help Target (Manual) Help Target (NetHelp) Help Target (WinHelp)

Remarks

This style sheet defines the help target appearance of the topics derived from HTML source documents. Appearance of the topics derived from Word source documents is defined by the Template property.

This is the target style sheet. Building a help target, Doc-To-Help replaces the source style sheet (the one defined by the HTMLDefaultCSS property) with this target style sheet. The HTML document can also have other style sheets attached to it where you can define appearance of styles that don't have special semantical meaning for Doc-To-Help, those additional style sheets are preserved by Doc-To-Help when it builds help targets.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

SuppressDefaultScript Property

Specifies whether a user script augments the default style behavior or completely overrides it.

Type	Applies To
Boolean	Script

Remarks

If you specify a script in the Script property of a character style, and that style has default D2HML behavior determined by its [Behavior](#) property, you can use the **SuppressDefaultScript** property to completely override the default behavior of the style. By default, **SuppressDefaultScript = False**, which means that your script will be executed after the default action defined by the style Behavior is performed. If you set the **SuppressDefaultScript** to **True**, your script will be the only action triggered by the style.

Location

- Click the **Project Icon**.
- Select the **Scripts** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

SuppressEmptyTopics Property

If set to **True**, omits topics defined with this paragraph style or topic type that have no text except heading from online help except in TOC.

Type	Applies To
Boolean	Paragraph Style Topic Type

Remarks

This property is usually set to **True** in the Contents topic type that is commonly used for the top-level style, such as Heading 1. Top-level topics often do not contain any text, appear only in TOC as book titles containing lower-level topics. Then it is often preferable that these (empty) topics are not shown in the help window when the user selects the corresponding book title.

Location

- Click the **Project Icon**.
- Select the **Paragraph Styles** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

SuperTitle Property

Specifies the text that appears above the title string on the title page of printed manuals.

Type	Applies To
String	Help Target (Manual)

Location

- Click the **Project Icon**.
- Make sure you have selected the Manual target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Title Page group in the **Properties Pane**.

Template Property

Specifies the name of the document template applied to intermediate files during compilation.

Type	Applies To
Template	Help Target

Remarks

This Word template defines the help target appearance of the topics derived from Word source documents. Appearance of the topics derived from HTML source documents is defined by the StyleSheet property.

Location

- Click the **Project Icon**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

TextPopups Property

If **True**, generates a plain text only version of the help file for context sensitive help topics.

Type	Applies To
Boolean	Document (HTML Help) Project Settings

Remarks

This property can be set as default in the Project Settings properties pane. Modifying this property for an individual document will override the global settings for that document only.

Location (Documents)

- Click the **Project Icon**.

- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Project Settings)

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Document Defaults group in the **Properties Pane**.

Theme Property

Determines the appearance of the Help pane and framesets for HTML-based targets.

Type	Applies To
Folder	Help Target (Help 2.0) Help Target (NetHelp) Help Target (HTML Help) Help Target (JavaHelp)

Remarks

You can use the BackgroundColor, BackgroundImage, and BackgroundRepeat properties in conjunction with the selected theme to customize the appearance of all topics.

Certain themes, such as Nonscrolling, will only work with Internet Explorer 5.0 or greater.

Location

- Click the **Project Icon**.
- The property is located under the Appearance group in the **Properties Pane**.

Title Property

Represents the title string of an object.

Type	Applies To
String	Help Target (Manual) Help Target (WinHelp) Topic

Remarks

For printed manuals, this property sets the text that appears below the [SuperTitle](#) string on the title page.

For topics, this property is read-only.

For the WinHelp target, this property represents the string to be displayed in the caption bar of the help window.

Location (Help Target)

- Click the **Project Icon**.
- Make sure you have selected the Manual or WinHelp target from the **Help Target Dropdown**.
- Select the **HelpTargets** item from the left pane Tree View.
- The property is located under the Title Page group in the **Properties Pane**.

Location (Topic)

- Click the **Topics Icon**.
- The property is located under the Misc group in the **Properties Pane**.

Title Property (Document)

Determines the title of the topic contained in a single-topic document.

Type	Applies To
String	Document

Remarks

This property applies only to single-topic HTML source documents. Multi-topic documents do not need it because their topics are defined by their headings in the document. With a single-topic document, there is no header to derive the title, table of contents position and other properties of the topic, so that information has to be specified explicitly in the **Title** and [Style](#) properties of the document. This is done by the user on adding the document to the project in **Adding HTML Documents** or **New HTML Document** dialog box.

Location (Documents)

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Top Property

Determines the location of the top edge of the help window.

Type	Applies To
Integer	Window (HTML Help) Window (WinHelp) Window (JavaHelp) Window (NetHelp)

Remarks

In conjunction with the [Left](#) property, determines the position of the help window on the screen.

Location

- Click the **Project Icon**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

TopicColor Property

Displays the current color setting for the topic help window. To change the color, click the adjacent change button to open a standard color selection dialog.

Type	Applies To
Long	Window (WinHelp)

Location

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Appearance group in the **Properties Pane**.

TopicsFound Property

Determines whether ALink/KLink topic lists are displayed in a dialog box (the default) or a pop-up menu.

Type	Applies To
Enum	Help Target (HTML Help)

Remarks

This property controls the appearance of links created with the Add Dynamic Link command or character styles. It does not override <OBJECT> tags for the HTML Help ActiveX control.

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

TopicType Property

Specifies a named set of display, navigation, and indexing characteristics to be associated with an object (such as what window the help topic appears in, how the help topic is accessed, and whether it gets a map number).

Type	Applies To
TopicType	Paragraph Style Topic

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Types** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

TriPaneWindow Property

When **True**, enables the standard tripane format for HTML Help.

Type	Applies To
Boolean	Window (HTML Help)

Location

- Click the **Project Icon**.
- Make sure you have selected the HTML Help target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.

- The property is located under the Appearance group in the **Properties Pane**.

Type Property

Determines the help authoring behavior associated with a paragraph style at compile time.

Type	Applies To
Enum	Paragraph Style

For Paragraph Styles:

Value	Description
Body Text	Topic body.
Heading Text	Topic with no outline level.
Level 1	Topic at outline level 1.
Level 2	Topic at outline level 2.
Level 3	Topic at outline level 3.
Level 4	Topic at outline level 4.
Level 5	Topic at outline level 5.
Level 6	Topic at outline level 6.
Level 7	Topic at outline level 7.
Level 8	Topic at outline level 8.
Level 9	Topic at outline level 9.

Location

- Click the **Project Icon**.
- Select the **Paragraph Styles** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Type Property (Document)

Indicates whether the source document is a Word document or an HTML document.

Type	Applies To
Enum	Document

Value

Microsoft Word document

HTML Document

Location

- Click the **Project Icon**.
- Select the **Documents** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Untitled Property

When **True**, the topic title is not displayed in the help window.

Type	Applies To
Boolean	Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles** or **Topic Type** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

URL Property

Specifies the name of the generated .htm file for a topic.

Type	Applies To
String	Topic

Remarks

By default, the name of the .htm file that is produced matches the topic text with spaces, hyphens and underscores omitted.

Location

- Click the **Topics Icon**.
- The property is located under the Miscellaneous group in the **Properties Pane**.

URLMode Property

Specifies the rules used for naming the .htm files generated for topics.

Type	Applies To
Enum	Project Settings

Value	Description
Full	File name contains all letters and digits from the topic title, even non-ASCII national alphabet letters.
ASCIIOOnly	Non-ASCII characters are removed from the file name. File name complies with URL standard.
TopicId	File name is formed from the numeric topic ID.

Remarks

These rules apply when the topic URL is generated automatically, building the project after the topic was first added.

By changing the URLMode property value, you can optionally apply the new rules to all existing topics.

You can modify the URL property of a topic later, manually overriding the default rules.

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

UseTemplateText Property

If set to **True**, the text of the **Template** associated with the Manual help target is included in the resulting document. Otherwise, the text is ignored, only template styles are used to format the document.

Type	Applies To
Boolean	Help Target (Manual)

Remarks

The template associated with the Manual target via the **Template** property is used for two different purposes:

Template styles determine the appearance of the generated document. This is always done if the Template property is non-empty, regardless of the value of this property.

Template text can also be used to determine such sections of the document as its title, table of contents and index. This is done if the UseTemplateText property is set to **True**. In that case, building the Manual target document, Doc-To-Help locates a bookmark with the name *ChapterHeading* in the template text and considers this bookmark a placeholder for the generated contents of the document, all document chapters. This allows the author of the template to specify the remaining sections of the document, that is, usually, Title, TOC and Index in the template text. If the UseTemplateText property is set to **False**, template text, even if present in the template, does not appear in the resulting document. In that case Doc-To-Help adds default title, TOC and index sections to the target document without using the template text.

Location

- Click the **Project Icon**.
- Make sure you have selected the Manual target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Value Property

Value portion of an arbitrary named value pair assigned to topics or help targets.

Type	Applies To
String	Attribute

Location

- Click the **Project Icon**.
- Select the **Attributes** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Width Property

Determines the width of the help window.

Type	Applies To
Integer	Window (HTML Help) Window (WinHelp) Window (JavaHelp) Window (NetHelp)

Location

- Click the **Project Icon**.
- Select the **Windows** item from the left pane Tree View.
- The property is located under the Position group in the **Properties Pane**.

Window Property

Shows the name of the window in which topics formatted in this style are displayed.

Type	Applies To
Window	Character Style Paragraph Style Topic Type

Location

- Click the **Project Icon**.
- Select either the **Paragraph Styles**, **Character Styles** or **Topic Type** item from the left pane Tree View.
- The property is located under the Display group in the **Properties Pane**.

WinHelpMacro Property

Specifies a macro to run when a Help file, topic, or secondary window is opened.

Type	Applies To
String	Help Target (WinHelp) Topic Window

Remarks

The WinHelpMacro property set for the WinHelp target will override the WinHelpMacro property set for the “main” window.

This property only applies to the WinHelp platform.

Location (WinHelp)

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Help Targets** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Location (Topics)

- Click the **Topics Icon**.
- The property is located under the Misc group in the **Properties Pane**.

Location (Windows)

- Click the **Project Icon**.
- Make sure you have selected the WinHelp target from the **Help Target Dropdown**.
- Select the **Windows** item from the left pane Tree View.

- The property is located under the Misc group in the **Properties Pane**.

WordSourceFolder Property

Defines the folder name where the Word source documents will reside.

Type	Applies To
String	Project Settings

Remarks

The default folder name is WordDocuments. This folder is automatically created when you create a project.

Although you can add any documents to the project even if they are not located in this folder or any of its subfolders, it is recommended to keep all Word source documents inside this folder because it helps to keep the project files organized.

Location

- Click the **Project Icon**.
- Select the **Project Settings** item from the left pane Tree View.
- The property is located under the Misc group in the **Properties Pane**.

Index

A

About 1

Accessibility property 611

Active Character Style 318

ActiveX Control 378

Add Topic Link Dialog Box 319

Adding

- Attributes 423
- Cascading Style Sheets 203, 207
- Character Style 216
- Conditional Text 87, 273
- Context IDs 268
- Context Strings 569
- D2HML 235, 403, 404
- Document 73, 99, 191, 192
- Glossary Entries 85, 334, 336
- Group 265, 349
- Help Targets 166
- Jumps 76, 252, 309
- Keywords 121, 256, 263, 343, 344, 348
- Link Tags 266, 289, 290
- Links 76, 233, 285, 288, 310, 311, 328, 484
- Margin Notes 323, 325
- Paragraph Styles 69, 215
- Scripts 442
- Source Documents 73, 92, 181, 192
- Styles 215, 219, 233
- Tables 75
- Templates 199
- Topic Types 225
- Topics 294
- Windows 45

AdjustForScreenSize property 611

AdjustLeftIndent property 612

AffectsAppearance property 612

ALinks 311

AlwaysOnTop property 613

AnswerCount property 613

Anti-virus 13

AsciiName property 613

AttributeExpression property 614

Attributes

- Adding 423
- Associating with Help Targets 424
- Conditional text 88, 401, 423
- D2HML Attribute 244
- Using 46, 244, 422

Authoring Templates 13

AutoButtons property 615

AutoContextID property 615

AutoGlossaryLinks property 615

AutoHideNavigationPane property 615

AutoIndex property 616

AutoKeyword property 616

AutoLink property 616

AutoNavigate property 617

AutoNext property 617

AutoSizeHeight property 617

AutoSyncNavigationPane property 618

B

Back Button 659

Background

- Color 359
- Image 360
- Window Pane 359

BackgroundColor property 618

BackgroundImage property 618

BackgroundRepeat property 618

BaseName property 619

Behavior property 619

BinaryIndex property 620

BinaryTOC property 620

Books 293

- Converting a Subtopic 293
- Creating 293
- Renaming 295

Browse Buttons 659

Build Icon 49

ButtonLabel property 621

ButtonLabelStyle property 621

ButtonSeparator property 621

ButtonStyle property 622

ByLine property 622

C

CancelKey property 622

CancelMap property 623

CancelTag property 623

Caption property 623

Cascading Style Sheets

- Adding 203, 207
- for Help Targets 206
- for Source Documents 201
- Importing Styles from 220
- Modifying 231
- Opening 209
- Setting the Default 15, 203

- Using 199, 201, 231
 - Character Styles
 - Active 318
 - Applying 228
 - Creating 219, 235
 - D2HML 235
 - Default 217
 - Properties 216, 219
 - Using 45, 216
 - Charset property 624
 - Citation property 624
 - Code property 624
 - CodePage property 624
 - ColorReduction property 625
 - Comments
 - Conditional text 245, 403, 404, 405, 414
 - XML 477
 - Comments property 625
 - Compacting 162
 - Compiling
 - Glossary Entries 334
 - Source Documents 196
 - Compression property 625
 - Concatenate property 626
 - Condition property 626
 - Conditional Text 401
 - Adding 87, 403, 404, 405
 - Applying 87, 88, 403, 416, 417, 419, 421, 423
 - Attributes 88, 401, 422, 424
 - Comments 245, 403, 404, 405, 414
 - Editing 410, 414, 417
 - Font Color 408
 - Format 410
 - in HTML 416
 - in Word 403, 405, 412
 - Printed Manual 87
 - Removing 409, 414, 417
 - Specifying 273, 401
 - View 406, 407
 - ConfidenceLevel property 627
 - Contents Button 659
 - Contents Icon 48
 - ContentsOnly property 628
 - ContentsTab property 628
 - ContentsTitle property 628
 - Context IDs 297
 - Adding 268, 300
 - Generating 298
 - Modifying 430, 433
 - Context Sensitive Help 297
 - Context Strings
 - Adding 569
 - Linking Topics 570
 - ContextString property 628
 - Converting
 - Doc-To-Help 2000 134
 - HTML Help projects 54, 143
 - RoboHelp 57, 136
 - WinHelp 59
 - CopyFolder property 629
 - Copying
 - Data 376
 - Styles 71
 - Copyright property 629
 - Creating
 - ALinks 311
 - Character Styles 219
 - Context Sensitive Help 298
 - Cross References 320
 - Hot Spots 367
 - Jumps 309
 - Keywords 343, 445
 - KLinks 310
 - Modular Help 428, 431, 435
 - Paragraph Styles 215
 - PopupTopics 310
 - Projects 130, 427, 457, 495
 - Scripts 441
 - Styles 215, 219, 227, 235, 309, 318
 - Topic Groups 349, 446
 - Windows 355
 - Cross Reference 320
- ## D
- D2HML
 - Attributes 244
 - Comments 245, 403
 - Defined 233
 - Dialog Boxes 236, 239
 - Formatting Rules 241
 - Quick Reference 246
 - Styles 233, 235, 312
 - Toolbar 233
 - DefaultFile property 629
 - DefaultTemplate property 630
 - DefaultWindow property 630
 - Deleting
 - Help Topics 378, 608
 - Margin Notes 325
 - Project Elements 377
 - Removing items 189, 196, 324, 347, 350, 378, 414, 417
 - Topics 378
 - DisplayTitle property 630
 - Distributing Files 12
 - Document Tree 183
 - Documenter for .NET 449
 - Controls 470
 - Document 473
 - Link Colors 493

- Narrative 463
- Project 457
- Special Characters 492
- Styles 484, 485, 486, 487, 488, 489, 490, 491
- Toolbar 472
- Tutorial 449

Documents

- Adding 44, 73, 99, 191, 192
- Compiling 196
- Opening 185, 190
- Order of 194, 307
- Saving 68
- Single and Multiple 183
- Source 67, 73, 92, 99, 192, 196

DocumentType property 631

Dynamic Links

- Adding 328

DynamicTOC property 631

E

Editing

- Character Styles 219
- Conditional Text 414, 417
- Glossary 334
- Help Contents 292
- Help Window 356
- Hyperlink Fields 372
- Macros 393
- Paragraph Styles 215
- Scripts 442
- Topic 289
- Topic Type 225

EmailAddress property 631

Existing Projects

- Converting 51, 134, 136

ExplicitAccess property 632

F

FavoritesTab property 632

Features 495

Find Button 660

FixLists property 632

Folder property 633

Forward Button 660

Frameset property 633

G

GenerateCSS property 633

GeneratePDF property 634

GenerateProjectFile property 634

Global property 634

Global Template 15

Glossary

- Creating 114, 333

- Definition 20

- Document 333

- Editing 334

- Filtering 335

- Links 114, 338

- Opening 334

- Sorting 338

- Terms 85, 334, 336

Glossary property 635

Graphics

- Hot Spots 367

- Image Map 111, 329, 560

- Inserting 365

- Scaling 365

GraphicsScaleWithFonts property 635

Group Links 311, 349

Groups

- Assigning Topics 349

- Renaming 349

Guided Tour 51, 449

H

Height property 635

Help 2.0

- Context Strings 569

Help Authoring 19

Help Contents

- Adding Topics 294

- Editing 291, 292

- Organizing 23, 293

- Saving 296

- Specifying 282

Help Project

- Creating 428, 431, 435

- Documents 73, 99, 191

- Starting 61, 92

Help Systems

- Building 22

- Defined 19

- Managing 427

- Modular 427

Help Target 46, 164

- Adding 166

- Building 164

- Changing 164

- Directory 167

- Viewing, current 166

Help Topics Button 661

Help Windows 353

Hidden property 636

HideNavigationPane property 636

Home Button 661

Hot Spots

- creating 233, 249, 319, 367, 561
- defined 233, 309
- Linking to Internet Sites 369
- properties 242
- showing hidden 275
- types 252

HTML Editors

- Adding 188
- Default 187
- Removing 189

HTML Help 171

- ActiveX control 378

HTML Help Object Tags 378**HTML Help Project 54, 143****HTML Passthrough Code 414****HTML Source Documents**

- Adding Editors 188
- Applying Conditional Text 416
- Converting HTML Help Projects 54, 143
- Converting RoboHelp Projects 136
- Editing 187
- Importing Styles 220
- Removing Editors 189
- Using 92
- Using HTML Styles 230
- Using Image Maps 329

HTMLDefaultCSS property 637**HTMLName property 637****HTMLSourceFolder property 637****HubContents property 638****Hyperlink 370**

- Editing 372
- Inserting 371

I**ID property 638****IgnoreFontNames property 638****IgnoreFontSizes property 639****Image Map 329, 560****ImageFormat property 639****Images**

- Background 360
- Image maps 329, 560

Importing

- Styles 221

In Place Editing 185**Index 20, 24**

- Building 341
- Constructing 24
- Keyword 343, 344, 345, 348

Index Icon 48**Index Keywords**

- Adding 343, 344

Creating 343, 445**Secondary 344****Indexing 24****IndexTab property 640****Installation 13****Internet Sites 369****J****JavaHelp 30, 177****Jump Button 640, 641****Jump1Caption property 640****Jump1URL property 640****Jump2Caption property 640****Jump2URL property 641****Jumps 286****K****KeepOutlineNumbers property 641****KeepPageBreaks property 642****Keyboard Shortcuts 397****Keyword**

- Adding 343, 344, 348, 445
- Assigning Topics 345
- Renaming 345

KLinks 310**L****Language 233, 481****Left property 642****Level property 642****License 12****Link Tags 266, 289, 290****Linking**

- Internet Sites 369
- Margin Notes 85, 323
- Topics, related 27

Links

- Adding 76, 104, 289, 328, 477, 479
- Alinks 311
- Cross-module 434, 437
- Dynamic 328
- Glossary 338
- Hot spots 309, 319
- in Documenter 484
- Klinks 310
- Live links 126
- Margin note 85, 323, 325, 326
- Next Topic 288
- Specific topics 285
- Subtopics 285, 287, 328

LinkType property 643**Lists**

- Applying 90
- Creating 90
- Nesting 230
- Printed Manual 230
- Live links 126
- Live update 161
- LiveLinks property 643
- Locale property 644
- Locate Button 662

M

- Macros 379
 - Deleting 393
 - Hotspot 380, 381, 382
 - Startup 384, 388
 - Topic 391
- Map Numbers 298, 300, 615
- MapFileForC property 644
- MapFileForVB property 644
- MapNumberOffset property 645
- MapNumbers property 645
- Margin Notes
 - Adding 323
 - Creating 84
 - Definitions, deleting 325
 - Linking 323
 - Links, clearing 326
 - Removing 324
- Maximized property 645
- MergeContents property 646
- MergeFile property 646
- MergeTitle property 646
- Microsoft Help 2.0
 - Context Strings 569, 570
- MidTopic Jump
 - Inserting 312, 327
- MidTopic property 647
- Modified property 647
- Modular Help 427
 - Creating 428, 431, 435
 - Defined 427
 - Managing 427, 645
 - Skip Glossary 665
 - TOC Utility 563
- Modular TOC Utility 563
- ModularHub property 647
- MultiLink property 648

N

- Name property 648
- Namespace property 648
- Natural Search
 - Content 553

- Distribution 558
- No Match Message 556
- Noise List 557
- Properties
 - AnswerCount 613
 - ConfidenceLevel 627
 - SearchCaption 656
 - SearchEnabled 656
 - StemPercent 667
 - StemSize 667
 - StringNoMatch 671
- Returns 556
- Searches 558
- Support 557
- Tab Caption 555
- Navigation 41
- Navigation Sequence 284
- NavigationPaneWidth property 649
- NetHelp
 - Modular help 435, 437
 - Properties 169
 - Searching 35
 - Target 32
 - Theme 572
 - Using 33
- New Features 3
- Next Button 663
- NonScrolling property 649
- NonscrollingColor property 649

O

- Object Model 611
- OneBrowseSequence property 650
- OnlineOnly property 650
- OnPage property 650
- Opening
 - Documenter 161
 - Files 185, 190, 200
 - Glossary 334
 - Projects 134, 161
- Options Button 663
- Order property 650
- Organizer 221
- OutlineInPDF property 651
- OwnTOC property 651

P

- PageRef property 652
- Paragraph Style
 - Active 281
 - Applying 69, 226, 227
 - Creating 215
 - Default 213

- Editing 215
 - Properties 211, 215
- Parent property 652
- ParentNamespace property 652
- PDF 127
- Picture 367
- Popup property 653
- Popup topics 310
- PrefixIndexListFile property 653
- PrefixTopicListFile property 653
- Previous Button 663
- Print Button 664
- Print Preview 374
- Printed Manual
 - Creating 124
 - Examining 124
 - Lists 230
 - Properties 173
 - Target 39
- Printing
 - Current view 374
 - Previewing 374
 - Reports 375
- Project Editor 41
- Project Files
 - Compacting 162
 - Managing 376
 - Organizing 190
- Project Icon 42
- Projects 129
 - Removing files 196
 - Settings 46
 - Toolbar 43
- Properties
 - Character Style 216, 219, 484
 - Document 191
 - Help Target 168
 - Help Window 356
 - Paragraph Style 211, 215
 - Project Settings 163
 - Topic 280, 289
 - Topic Type 222, 225
 - Window 353, 354, 355, 356

R

- Redistributable Files 12
- Refresh Button 664
- RelatedJumps property 653
- Renaming
 - Books 295
 - Groups 349
 - Keywords 345
 - Topics 295
- Replacement property 654

- Reports
 - Available 375
 - Generating 373
 - Selecting 375
- RoboHelp Projects 51, 57, 136, 147

S

- SaveUserPosition property 654
- ScaleFontSizes property 654
- ScaleGraphics property 655
- Script property 656
- Scripts
 - Creating 441
 - Debugging 443
 - Defined 45, 439
 - Editing 442
 - Module 442
 - Operations 444
 - Techniques 439
 - Working with 441
- SearchCaption property 656
- SearchEnabled property 656
- SearchStopList property 657
- SearchTab property 658
- SearchTabAdvanced property 658
- Secondary Keywords 344, 445
- Shortcut Keys 246, 397
- ShowBackButton property 659
- ShowBrowseButtons property 659
- ShowContentsButton property 659
- ShowDropdownText property 659
- ShowExpandText property 660
- ShowFindButton property 660
- ShowForwardButton property 660
- ShowHelpTopicsButton property 661
- ShowHideShowButton property 661
- ShowHomeButton property 661
- ShowIndexButton property 661
- ShowJump1Button property 662
- ShowJump2Button property 662
- ShowLocateButton property 662
- ShowNextButton property 663
- ShowOptionsButton property 663
- ShowPreviousButton property 663
- ShowPrintButton property 664
- ShowRefreshButton property 664
- ShowStopButton property 664
- SingleTopic property 664
- Size property 665
- SkipGlossary property 665
- Sorting
 - Glossary 338
 - Topics 306, 307
- SplitFile property 665

SplitFileWithHHCtrl property 666
 Standard Table 368
 StemPercent property 667
 StemSize property 667
 Stop Button 664
 StringContents property 667
 StringEmail property 668
 StringFavorites property 668
 StringFavoritesAdd property 668
 StringFavoritesCaption property 669
 StringFavoritesDelete property 669
 StringFoundMany property 669
 StringFoundOne property 670
 StringFoundZero property 670
 StringHideNavigationPane property 670
 StringIndex property 670
 StringIndexCaption property 671
 StringNext property 671
 StringNoMatch property 671
 StringPrevious property 672
 StringPrint property 672
 StringReturnToIndex property 672
 StringSearch property 673
 StringSearchCaption property 673
 StringSearchGo property 673
 StringSearchHighlight property 673
 StringSyncTOC property 674
 Style property 674
 Style Sheets

- Help Target 205
- Source Document 201
- Using 15, 201, 231

 Styles 211

- Assigning Scripts 442
- Assigning Windows 356
- Character 216
- Creating 215, 219, 227, 235, 318
- Documenter 463, 485, 486, 487, 488, 489, 490, 491
- Importing 221
- Link 485, 486, 487, 488, 489, 490, 491
- List 229
- Paragraph 211, 281

 StyleSheet property 674
 Subtopics 285, 293

- Labels 287

 SuperTitle property 676
 Support 12
 SuppressDefaultScript property 675
 SuppressEmptyTopics property 675
 System Requirements 13

T

Tables

- Insert 368

Team authoring

- Auxiliary files 535
- Building a target 542
- Documents 532, 534, 539
- Menu 531
- Node in project editor 533
- Removing support 548
- Restrictions 514
- Team project 515, 526, 536, 543
- Toolbar icons 532

 Technical Support 12
 Template property 676
 Templates

- Adding 199
- Authoring 13
- Global 15

 TextPopups property 676
 Theme Designer 572
 Theme property 677
 Themes

- Custom 603
- Descriptions 358, 359
- Designer 572
- Using 357

 Title property 677, 678

- of Document 678

 Toolbars

- Components 65
- Contents 48, 292
- D2HML 233
- Doc-To-Help 64
- Documenter 472
- Image Map 561
- Index 49, 342
- Project 43
- Project Properties 43
- Topic Properties 47

 Top property 678
 Topic Groups 349
 Topic Tools

- Add Topic Button 604
- Delete Topic Button 608
- Rename Topic Button 607

 Topic Types

- Adding 225
- Applying 225, 226
- Default 224
- Using 45, 222

 TopicColor property 678
 Topics 285

- Customizing 286
- Defining 23, 279
- Filtering 308
- Found 311
- Popup 310

- Properties 289
- Removing 350
- Sorting 306
- Topics Icon 46
- TopicsFound property 679
- TopicType property 679
- TriPaneWindow property 679
- Tutorial 51
- Type property 680
 - of Document 680

U

- Untitled property 680
- URL property 681
- URLMode property 681
- UseTemplateText property 682

V

- Value property 682
- View
 - Conditional Text 406, 407
 - Glossary terms 336
 - Help target 166
 - Hyperlinks 81
 - Margin note links 326
- View Conditional Text 406
- Views Icon 46

W

- Width property 682
- Window property 683
- Windows 45
 - Background 359
 - Creating 355
 - Customizing 353, 363
 - Editing 362
 - Themes 357
- WinHelp 4.0 175, 355
- WinHelpMacro property 683
- WordSourceFolder property 684

X

- XML 454, 457, 477