ComponentOne

Doc-To-Help 2009 User Guide

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.

2009 V3

Contents

Welcome to Doc-To-Help 2009!

Doc-To-Help 2009 is a single source authoring tool that makes it possible to write once and publish many different deliverables. Technical communicators, Help authors, policy writers, and other content creators can author any type of information in Microsoft Word, HTML, or XML, and publish to the Web, Help systems, or printed manuals.

The unmatched flexibility of **Doc-To-Help 2009 Enterprise** allows you to author in the environment of your choice. If you prefer to work in XML, you can do so using our built-in XHTML editor. Integrated toolbars and dialog boxes in Microsoft® Word and FrontPage®, as well as Adobe® Dreamweaver®, simplify the use of Doc-To-Help Markup Language (D2HML) (on page 210) to quickly and easily create Help systems by defining topic links, conditional text, pop-ups, glossary terms, and more.

Doc-To-Help 2009 is the perfect choice for creating innovative online user assistance — compiled HTML Help, NetHelp (browser-based, platform independent Help for Web deployment), Help 2.0, JavaHelp and WinHelp — converting existing Word documents to online materials, or producing professional quality printed manuals.

The crisp, clean interface incorporates the new (and easy-to-use) Ribbons, Accordion Navigation, Expanded ToolTips, and Dynamic Help. To learn more about the Doc-To-Help interface, see Navigating Doc-To-Help on page 95.

Doc-To-Help 2009 is compatible with Microsoft® Windows Vista, Microsoft® Windows 7, and Office 2007.

Quick Start resources:

- Doc-To-Help Guided Tour (on page 49)
- How-To Video Library (click the **Doc-To-Help** tab)
- Free Introductory Webcasts (http://www.componentone.com/D2HWebcastSignup/)
- Training Information (http://www.componentone.com/SuperProducts/DocToHelpTraining/)
- Introduction to Single Sourcing (on page 1)
- Doc-To-Help Outputs and Deliverables (on page 30)

Introduction to Single Sourcing

Single sourcing documentation can save time, as well as help cut down on errors. This section provides an overview of what single sourcing is, how Doc-To-Help makes it easy, a single sourcing methodology, and handy tips and tricks.

Definition

Techniques used to create some combination of documentation for:

- **Multiple output formats*** (print manual, online help, online document, etc.)
- **Multiple audiences** (For example: Administrator version, Manager version)
- **Multiple deliverables** (For example: user documentation, training, etc.)

Use the conditional text and variables (on page 194) features to customize output for multiple formats, audiences, and deliverables.

*Formats you can create with Doc-To-Help (Doc-To-Help refers to these as "Targets").

Doc-To-Help Targets

- **Online Help**
    - **HTML Help** (Deliverable: compiled HTML Help; .chm – pronounced "chum") Help displays in a Tri-Pane window. Newer PC security measures can block viewing of .chm files across a network.
    - **NetHelp** (Also referred to as browser-based Help, the final deliverable is uncompiled HTML Help – meaning you will create and deliver multiple HTML files) The NetHelp viewer is unique to Doc-To-Help.
    - **Help 2.0** (Only used to create Help for .NET components; projects are integrated with Microsoft® Visual Studio .NET)
    - **WinHelp** (Deliverables: .cnt and .hlp files) (Older Help format being phased out. For more, read this article.)
    - **JavaHelp** (Deliverable: Help for Java applications)
- **Printed Manuals**
    - "Printed" manuals don't have to be printed, although they can be. With Doc-to-Help, you can generate Microsoft® Word files (.doc or .docx [Office 2007]), and .pdf files (portable document format). Either format can be sent to a printer, or provided as an online document. If you don't want your file altered, it is best to provide it as a .pdf.

You can also create your own custom Targets based on these outputs. See Creating Help Targets on page 155.

How Doc-To-Help Makes Single Sourcing Easier

Author in Word, HTML, and/or XHTML; automatically create a manual and online Help.

1. Automatically structures content and navigation of online Help.
    a. Heading 1's automatically become parent topics, and all of the Heading 2's under it become subtopics.
    b. Parent topics automatically include "See Also" links to subtopics. See Managing Related Topics (on page 225) for more information.
2. Printed manual cross references (with page numbers) are automatically converted to hyperlinks (Live Links) in online Help. Also, manuals can be set up to include both hyperlinks and page number cross references. (Useful for both types of end-users – those who prefer to read on screen or print out the manual.)
3. Text can be flagged so that it appears as drop-down or expanding text in online Help. This text will automatically be included in the printed manual.
4. Margin notes included in printed manuals automatically become pop-ups in online help.
5. Doc-To-Help automatically generates the Title Page, Table of Contents, and Index for printed manuals, so you don't have to.
6. The conditional content feature makes it easy to customize text, graphics, topics, and documents for different formats, audiences, and deliverables. See Utilizing Conditions (on page 174) for more information.

Three Easy Steps to Single Sourcing

Step 1: Access Requirements

- Determine format, audience, and deliverable requirements. Things to consider:
    - What is possible given time constraints? (If deliverable is software documentation, also consider development constraints – will product developers have time to add Help buttons to dialogs and provide Context IDs, add a Help menu and links, etc.)
    - What deliverables would make our customers the most successful?
    - What deliverables do customers expect?
    - What deliverables could enhance training efforts?

Step 2: Get Ready

1. Write one book chapter (or help topic) – make sure it has all objects and styles represented – several levels of headings, bullets, numbered tasks, tables, etc. Use greeked text as a placeholder if complete information isn't available. (See http://www.duckisland.com/GreekMachine.asp or http://www.lipsum.com/ for a greeked text generator.) You can use this to experiment with the look of your output online vs. print.
    a. If you are creating software documentation, design a consistent structure for topics that map to Help buttons. For example:
        - **Introduction**
        - **How to access the screen/dialog** (Can be drop-down or expanding text online to save space and cut down on clutter.)
        - **See Also/Related Topic links**

    b. Standardize on a consistent heading (such as "Using the… Screen") which makes sense in both the book and help.

    c. Standardize on one (at most two) Heading styles that Help button topics will map to. It can be confusing to the user if multiple styles appear from Help buttons. Make sure to structure information so that Help buttons have only one logical mapping. (Dialogs that are reused throughout application can make this challenging.)
2. If using conditional text, create a few basic conditions and experiment. (Online only, Print only, Draft, etc.) **Tip:** You can exclude illustrations from online Help using conditional text (some illustrations used in printed output are unnecessary or too detailed for online use); if you still want illustrations in Help, create unique, focused ones and make their condition "Online only."
    a. Note: in addition to sentences, you can conditionalize topics and even entire documents with Doc-To-Help. See Utilizing Conditions (on page 174) for more information.
3. Keep in mind that document structure can affect the look of the automatically-generated Help TOC.
    a. If a Heading 1 has no Heading 2's under it, it will not appear as a "book" in the Help TOC.

    b. You can add Heading 2's (subtopics) under it to avoid this — or change a "topic" to a "book" in the Doc-To-Help Contents pane (on page 109).

    c. Sometimes it is a good way to make a topic stand out.

Step 3: Get Set & Go

1. Create a new project.
2. Choose your source/target templates and Style Sheets (CSSs) (on page 5). Target templates/CSSs determine how your target output will look. You can always keep the defaults at this point and change them later.
3. Apply styles to your text. Please note that the formatting in your source document is for your eyes only. Your target template(s) or CSS(s) control the look of your output. (Unless your source template/CSS includes styles not available in your target template/CSS; then the source style passes right through to the target.)
4. Create any necessary variables (on page 194).
5. If using conditional text, create conditions.

Note: In addition to sentences and graphics, you can conditionalize topics and even entire documents.

6. Add topics to the index (on page 110), add glossary items, add links, etc.
7. If you would like to change the way your online Help "skin" looks, use the Theme Designer (on page 11) to create a new theme.
8. Generate your desired target(s) and see what you think!

Tips for Logical Output

To help your documentation read logically in both printed manual and online Help:

- Never use the word "chapter" or any other word specific to books or help. (For example, "page" "topic" "help system" "manual") "Section" is a good substitute.
- Use "See *cross reference*" rather than "See below," "See above," etc.
- Avoid conditionalizing cross references. Cross references are usually needed in both the printed manual and the online Help.

Additional Tip: Create a variable (on page 194) for your product name (or names). This can make it easy to do a quick swap if your product name changes, or if you need to create deliverables for multiple products.

Guide to Templates and Styles

Doc-To-Help uses predefined templates and style sheets to determine how content will look in both the Source documents (when authoring in Word, an HTML editor, or the XHTML editor) and the final Targets – the online Help and printed manual outputs.

The predefined templates and style sheets contain styles that you apply to your source. These source styles are interpreted by the Target template and/or style sheet you have chosen so that your target files **look and behave** the way you desire. Because styles control your final targets in multiple ways, it is important to use them and avoid applying local formatting.

- For Doc-To-Help projects containing Microsoft® Word (.doc or .docx) files, the look of their Source/Target is controlled by separate Word templates (.dot or .dotm files). See Word File Templates on page 7.
- For Doc-To-Help projects containing HTML and XHTML files, the look of their Source/Target is controlled by separate cascading style sheets (.css files). See HTML File Style Sheets on page 10.

The look of your output depends on your source files – if you are working in Word, the Target Template (.dot/.dotm file) will control the formatting and behavior of ALL outputs (even HTML help). If you are working in HTML or XHTML, the Target Style Sheet (.css file) will control the formatting and behavior of your outputs. Word File Templates (on page 7) and HTML File Style Sheets (on page 10) explain this in detail.

If your project uses Word, HTML, and XHTML source documents, the Target Template will control the output of the Word documents and the Target Style Sheet will control the output of the HTML and XHTML documents.

Note: A Doc-To-Help project can include Word files, HTML files, XHTML files, or a combination of all three.

Templates and Styles Quick Facts

- Doc-To-Help automatically assigns the complimentary Target template or style sheet based on the Source template or style sheet you have chosen.
- You can customize templates and cascading style sheets to make your source and target(s) content look and behave any way you wish. You should always create your documents with the templates and style sheets included with Doc-To-Help and edit them as desired; this will guarantee full Doc-To-Help functionality, including D2HML.
- D2HML is a special style set that is used to create links, dropdown or expanding text, apply conditions, insert variables, and more. See Using D2HML (Doc-To-Help Markup Language) (on page 210) for more information.
- Character Styles, Paragraph Styles, and Topic Types can be edited within Doc-To-Help to manipulate their behavior in your final targets. See Defining Character/Paragraph Styles and Topic Types (on page 181) for more information.
- Styles (specifically Heading Styles) also control the topic hierarchy for your project – for example, Heading 1's automatically become Parent topics, and all of the Heading 2's under it are its children (subtopics). Parent topics automatically include "See Also" links to subtopics. See Subtopic Links for more information. The same Heading style hierarchy is also used when a Table of Contents is automatically generated. (You can, of course, edit your Table of Contents — you can even create a customized TOC for each Target. See Contents pane (on page 109) for more information.)
- Any styles that exist in the Source templates or style sheets but not in the Target templates or style sheets will "passthrough" to the final generated output.
- Please note that the "skin" that surrounds your online Help content (HTML Help, NetHelp, Help 2.0, and JavaHelp targets specifically) is configured in the Theme Designer (on page 11). (The skin includes the items in the Help window that surround the content — breadcrumbs, tabs, buttons, etc.)

To apply a source template

In existing Word documents:
The document must be open in Word to apply a template. The procedure to apply templates varies slightly between Word 2007 and earlier versions, but both use the **Templates and Add-ins** dialog box.

- Word 2003 and earlier: **Tools > Templates and Add-Ins**. The current template will be displayed in the **Document Template** field. Click the **Attach** button to attach a different template.
- Word 2007: **Office button > Word Options button >** click **Add-Ins** in the left pane > from the **Manage drop-down** list, choose **Templates** > click **Go**.

Setting the default for new Word Documents:
You may set the default Source Template for all new Word documents created in your project. In the **Home tab** > Source ribbon group (on page 99), click the **Source Template** button. Choose a different template from the list or choose **Add Template** to add another. This source template will be applied to all new documents. This option may also be set in the Project Settings (on page 196) dialog box (**Default Template** field).

To apply a source style sheet

In existing HTML documents:
The document must be open in your HTML editor (FrontPage or Dreamweaver) to apply a CSS.

- In Microsoft® FrontPage®: **Format > Style Sheet Links**. The **Link Style Sheet** dialog box will open. Click the **Add** button to add your Doc-To-Help style sheet.
- In Adobe® Dreamweaver®: **Text > CSS Styles > Attach Style Sheet**. The **Attach External Style Sheet** dialog box will open. Click the **Browse** button to navigate to the Doc-To-Help style sheet.
- In Doc-To-Help's XHTML Editor: **Home tab > Source ribbon group**, click the **Source CSS** button.

Setting the default for new HTML or XHTML Documents:
You may set the default Source Style Sheet for all new HTML or XHTML documents created in your project. In the **Home tab** > Source ribbon group (on page 99), click the **Source CSS** button. Choose a different style sheet from the list or choose **Add CSS** to add another. This option may also be set in the Project Settings (on page 196) dialog box (**Default CSS** field).

To apply the target template and style sheet

By default, when you choose a Doc-To-Help **Source** Template or Style Sheet, its accompanying **Target** Template or Style Sheet is chosen. See Word File Templates (on page 7) and HTML File Style Sheets (on page 10) for details.

- To change the **Target** Template: In the **Home** tab, Target Design ribbon group (on page 98), click the **Target Template** button. Choose a different template from the list or choose **Add Template** to add another.
- To change the **Target** Style Sheet: In the **Home** tab, Target Design ribbon group (on page 98), click the **Target CSS** button. Choose a different style sheet from the list or choose **Add CSS** to add another.

To edit a template

See Editing a Template on page 8.

To edit a style sheet

You may edit CSS files using your own editor, or you may use Doc-To-Help's editor. See Editing a CSS (on page 189) and HTML File Style Sheets on page 10.

Applying Styles

To learn more about applying styles, see:

- Editing Word Documents on page 128
- Editing HTML Documents on page 135
- Editing XHTML Documents on page 119

Word File Templates

When working with Word Source Files

If you want to…

- Change the look of your **Source Word files** – Edit the **Source Template** (Column 1)
- Change the look of your **Printed Manual Target** – Edit the **Printed Manual Target Template** (Column 2)
- Change the look of your **NetHelp/HTML Help Target** – Edit the **Online Help Target Template** (Column 3 — C1H_HTML.DOT)
- Change the look of your **WinHelp Target** – Edit the **Online Help Target Template** (Column 3 — C1H_HELP.DOT)

When working with Word files, using a Source Template together with its accompanying Target Template makes it easier to see how doc files will look when converted to the Printed Manual Target.

The Word Doc-To-Help toolbars (on page 128) (**Doc-To-Help** and **Doc-To-Help Special Formatting**) reside in individual Doc-To-Help templates, so you should always create your files using one of the predefined Doc-To-Help templates and edit it to your specifications.

Predefined Source Templates	Predefined Target Templates	
Pick One	Printed Manuals Pick One	Online Help Determined by Target
C1H_NOMARGIN.DOT: The Doc-To-Help default source template starting with Doc-To-Help 2007. **It will be used unless you use your own or choose an alternate from this list.** In the New Project Wizard called "Normal 8.5" x 11" Template (no left indent)"	**C1H_PRNOMARGIN.DOT**: This is the target template used to format the printed manual target. It differs from C1H_PRNORM.DOT in that it does not have the wide two inch left margin.	**C1H_HELP.DOT**: This is the target template used to format WinHelp.
C1H_NORM.DOT: This is Doc-To-Help's default source template For all versions prior to Doc-To-Help 2007. This matches C1H_NOMARGIN.DOT, but it has a two-inch left margin. In the New Project Wizard called "Normal 8.5" x 11" Template"	**C1H_PRNORM.DOT**: This is the target template used to format the printed manual target. It differs from C1H_PRNOMARGIN.DOT in that it does not have the wide two inch left margin.	**C1H_HTML.DOT**: This is the target template used to format NetHelp, Microsoft HTML Help, Help 2.0, and JavaHelp.
C1H_NORM_A4.DOT: This is the source template for A4 sized paper. In the New Project Wizard called "Normal A4 Template"	**C1H_PRNORM_A4.DOT**: This is the target template used to format the printed manual target (A4 size paper).	

Predefined Source Templates	Predefined Target Templates	
C1H_SIDE.DOT: This is the source template for sidehead source documents. In the New Project Wizard called "Sidehead 8.5" x 11" Template"	**C1H_PRSIDE.DOT**: This is the target template used to format the standard sidehead printed manual.	
C1H_SIDE_A4.DOT: This is the source template sidehead source documents on A4 sized paper. In the New Project Wizard called "Sidehead A4 Template"	**C1H_PRSIDE_A4.DOT**: This is the target template used to format the standard sidehead printed manual (A4 size paper).	
C1H_SMAL.DOT: This is the source template used to format small-sized manuals. In the New Project Wizard called "Small 7" x 9" Crop-Marked Template"	**C1H_PRSMAL.DOT**: This is the target template used to format the standard small printed manual.	
C1H_SMAL_A4.DOT: This is the source template used to format small-sized manuals on A4 sized paper. In the New Project Wizard called "Small 17.78 cm x 22.68 cm Crop-Marked Template"	**C1H_PRSMAL_A4.DOT**: This is the target template used to format the standard small printed manual (A4 size paper).	

Templates (.dot files) are edited in Microsoft Word. See Editing a Template (on page 8) for more information.

Editing a Template

When editing a Microsoft® Word template (.dot) file, keep in mind that changes to the Source template affect only the look of the Source documents. Editing the Target template (or templates) changes the look of your final target. See Word File Templates (on page 7) for the list of available templates.

It is good practice to backup your original template(s) so that you can restore them if needed.

You can open your templates directly from Doc-To-Help.

To open a Source or Target template

1. Open the **Home** tab in Doc-To-Help.
2. To open Target templates for editing:
   - From the **Target Design** ribbon group, click the **Target Template** drop-down arrow.
   - Choose **Edit Template**. The selected template will open in Microsoft Word.
3. To open **Source** templates for editing:
   - From the **Source** ribbon group, click the **Source Template** drop-down arrow.
   - Choose **Edit Template**. The selected template will open in Microsoft Word.

To add a template to the **Source Template** or **Target Template** drop-downs, choose **Add Template**. To remove a template from the drop-downs, choose **Remove Template**. Removing a Template in this manner will only remove it from the drop-down list; not delete it permanently.

To edit a Template

1. After the template is open, the process will vary depending upon your version of Microsoft Word.
   - In Word 2007: Click the **Home ribbon** > **Styles ribbon group** dialog box launcher. The **Styles** window will open.
   - In Word 2003 and earlier: From the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** window will open.
2. Click the style you would like to edit from the list and choose the drop-down next to it. Choose **Modify**. The **Modify Style** dialog box will open. Make the changes desired by selecting the **Format** drop-down. When you are done, choose **New Documents Based on this Document** radio button (Word 2007) or **Add to Template** check box (Word 2003) in the **Modify Style** dialog box and click **OK**.
3. In Word 2003, save the template as a .dot ("Document Template") file. If you modify a template in Word 2007, you should save it as a .dotm ("Word Macro-Enabled Template") file.

This will only affect the look of a style. If you wish to change its behavior in the final Target (for example, its level or whether the style is automatically added to the TOC or index), see Defining Character/Paragraph Styles and Topic Types on page 181.

Please Note: Templates are stored in Windows® Vista at C:\Users\ (user name)\AppData\Roaming\Microsoft\Templates; in Windows XP at C:\Documents and Settings\(user name)\Application Data\Microsoft\Templates.
If you uninstall Doc-To-Help and reinstall a newer version, the templates (including customized ones) will be saved to the \Backup folder of those directories (Doc-To-Help will save the last ten backups).

The Application Data or App Data folder is a hidden folder. To show hidden folders, open the **Folder Options** dialog box in your operating system (**Control Panel** > **Folder Options**). Click the **View** tab, under **Advanced Settings** > **Files and folders** > **Hidden files and folders**, select the **Show hidden files and folders** radio button. Click **OK**.

HTML and XHTML File Style Sheets

When working with HTML or XHTML Source Files

If you want to …

- Change the look of your **Source HTML or XHTML files** — Edit the **Source Style Sheet** (Column 1)
- Change the look of your **Printed Manual Target*** — Edit the **Printed Manual Target Style Sheet** (Column 2)
- Change the look of your **NetHelp/HTML Help Target** — Edit the **Online Help Target Style Sheet** (Column 3 — C1H_HTML_full or short.css)
- Change the look of your **WinHelp Target*** — Edit the **Online Help Target Style Sheet** (Column 3 — C1H_HELP_full or short.css)

<table>
<tr><th>Predefined Source Style Sheets</th><th colspan="2">Predefined Target Style Sheets</th></tr>
<tr><td>Pick One</td><td>Printed Manuals
Pick One</td><td>Online Help
Pick One for HTML or XHTML/One for WinHelp</td></tr>
<tr><td>C1H_Source_full.css: Source style sheet with the full set of styles.
In the New Project Wizard called “Full Set of Styles”</td><td>C1H_Print_full.css: Target style sheet for Manual target with the full set of styles.</td><td>C1H_HTML_full.css: Target style sheet for all HTML-based targets with the full set of styles.</td></tr>
<tr><td rowspan="3">C1H_Source_short.css: Source style sheet with the minimum set of styles.
In the New Project Wizard called “Minimum Set of Styles”</td><td>C1H_Print_short.css: Target style sheet for Manual target with the minimum set of styles.</td><td>C1H_HTML_short.css: Target style sheet for all HTML-based targets with the minimum set of styles.</td></tr>
<tr><td rowspan="2">C1H_Print_nomargin.css: Target style sheet for Manual with the full set of styles, without the wide two inch left margin.</td><td>C1H_Help_full.css: Target style sheet for WinHelp target with the full set of styles.</td></tr>
<tr><td>C1H_Help_short.css: Target style sheet for WinHelp target with the minimum set of styles.</td></tr>
</table>

***Please Note:** If you are creating Manual or WinHelp Targets, your Target Template Styles will override those in your Target CSS if there are any differences between the two.

You may edit CSS files using your own editor, or you may use Doc-To-Help's editor.

If you use Doc-To-Help's style sheet editor, you will not edit these style sheets directly. See Editing a CSS on page 189 for details.

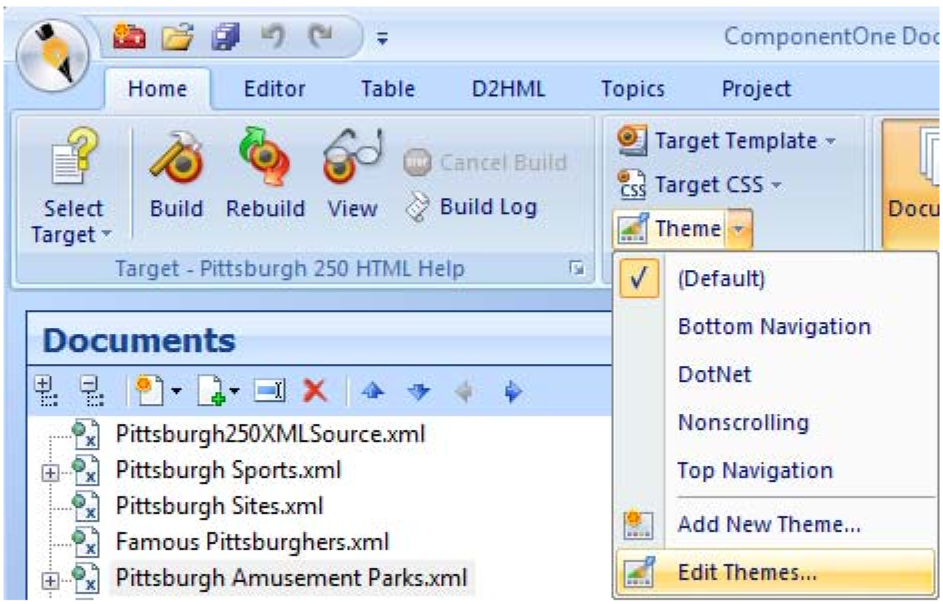
- **In new projects:** When you create a new project in Doc-To-Help 2009, the project style sheet chosen when creating the project will become read-only and be stored in the \\(project folder)\CSSFiles folder. Any changes you make will be saved in the same folder, with the style sheet name truncated so that the prefix and suffix are removed (for example, “C1H_Source_Full.css” will become “Source.css.” See Style List Window Tour on page 192 for more on how CSS files are inherited.
- **In existing projects:** When you open a project created in an earlier version of Doc-To-Help, the Style Sheets for the project will be found in the \\(project folder)\CSSFiles folder. You should edit the Style Sheets found in this folder. You can do so using the Style List Window (on page 192) if you wish.
- Copies of the original Style Sheets are stored in \\Program Files\ComponentOne\DocToHelp\DefaultCSSFiles.

Customizing with the Theme Designer

The **Theme Designer** is used to modify the "skin" that surrounds your online Help content. The skin includes the items in the Help window that surround the content — breadcrumbs, tabs, buttons, etc. Window options that are not specified with the Theme Designer are controlled by the Help window settings (on page 176).

To select a theme for a Help Target

- Click the **Theme** drop-down list on the **Home** tab. Choose one from the list. The Themes available will vary depending on the Target chosen (to change the Target, click the **Select Target** button).



See Editing Theme Properties on page 13 for information on adding or editing a **Theme**.

Note: Themes are stored in \\Program Files\ComponentOne\DocToHelp\Themes in the CHM, NetHelp, or Java folders. It is good practice to back up your custom themes as well as your projects. Please note that if you uninstall Doc-To-Help and reinstall a newer version, older themes (including customized ones) will be saved to the \\Program Files\ComponentOne\DocToHelp\Backup\Themes folder.

Available Themes

Doc-To-Help has default themes for each Help Target — HTML Help, NetHelp, Help 2.0, and JavaHelp. Several targets have several other pre-defined themes.

To preview any theme

1. Choose the appropriate Target from the **Home** tab using the **Select Target** button.
2. Click the **Theme** button on the **Home** tab. The **Theme Preview** dialog box will open.
3. Choose any Theme from the **Theme** drop-down in the dialog box.

Target	Themes Available	Description
HTML Help or Help 2.0	**Default**	• Tripane window with Contents, Index, Search tabs at left • Breadcrumbs above Help content
	DotNet	• Tripane window with Contents, Index, Search tabs at left • Previous/next arrows and breadcrumbs above Help content • Topic heading, arrows, and breadcrumbs displayed in a colored non-scrolling area similar to the MSDN Library in Microsoft Visual Studio.NET
	Top Navigation	• Tripane window with Contents, Index, Search tabs at left • Previous/next icons and breadcrumbs above Help content
	Bottom Navigation	• Tripane window with Contents, Index, Search tabs at left • Previous/next icons below Help content • Breadcrumbs above Help content
	Nonscrolling	• Tripane window with Contents, Index, Search tabs at left • Breadcrumbs above Help Topic title • “Previous” and “Next” hyperlinks below Help Topic title • Breadcrumbs, Help Topic title, and navigational hyperlinks are displayed in a colored non-scrolling area
NetHelp	**Default**	• Contents at left; Help content at right • Contents/ Index/Search/Previous/Next icons, “Print” and “Email” hyperlinks, and breadcrumbs above Help content • Topic heading, icons, hyperlinks, and breadcrumbs displayed in a colored non-scrolling area similar to the MSDN Library in Microsoft Visual Studio.NET
	DotNet	• Contents at left; Help content at right • Contents/ Index/Search/Previous/Next icons, “Print” and “Email” hyperlinks, and breadcrumbs above Help content • Topic heading, icons, hyperlinks, and breadcrumbs displayed in a colored non-scrolling area to the MSDN Library in Microsoft Visual Studio.NET
	TripaneXP	• Contents at left; Navigation pane at top • Help content at right • Contents and Navigation pane have Windows XP look and feel • Navigation pane includes Contents/Index/Search buttons; Previous and Next icons • Breadcrumbs, “Print” and “Email” hyperlinks above Topic heading in Help content area
	Tripane Classic	• Contents at left; Navigation pane at top • Help content at right • Navigation pane features Contents/Index/Search/Previous/Next buttons on a dark blue background. • Breadcrumbs, “Print” and “Email” hyperlinks above Topic heading in Help content area
	Nonscrolling	• Contents at left; Help content at right • Breadcrumbs above Help Topic title • “Contents”, “ Index”, “Search”, “Previous”, “Next”, “Print” and “Email” hyperlinks below Help Topic title • Breadcrumbs, Help Topic title, and navigational hyperlinks are displayed in a colored non-scrolling area

Target	Themes Available	Description
	DocToHelp 2005	• Contents at left; Navigation pane at top • Help content at right • Navigation pane features Contents/Index/Search/buttons on a light blue background • Previous/Next buttons in Contents pane, also on a light blue background • Breadcrumbs, “Print” and “Email” hyperlinks above Topic heading in Help content area
	TextOnly	• Contents at left; Help content at right • “Contents”, “ Index”, “Search”, “Previous”, “Next”, “Print”, “Email” hyperlinks and breadcrumbs above Help Topic title
JavaHelp	**Default**	• “Previous” and “Next” hyperlinks appear at the top of the window

Editing Theme Properties

Theme Properties are edited in the **Theme Properties** dialog box. The properties available will depend upon the Target, which is chosen from the **Home tab** > **Target ribbon group** > **Select Target** button. You can use/create a different theme for each Target in your project. See Available Themes on page 11 for the list of themes included with Doc-To-Help.

After you have selected your Target, click the **Theme** button on the **Home** tab to open the **Theme Preview** dialog box. This dialog box previews the Theme for the selected Target. Click the **Theme** drop-down to see other available themes for this Target.

To add a theme

New themes are based on existing, built-in themes.

1. On the **Home** tab, click the drop-down arrow next to the **Theme** button.
2. Choose **Add New Theme**. The **Add New Theme** dialog box will open.
3. **Name** the new theme, and choose the **Source theme** (see Available Themes on page 11 for a description of each built-in theme).
4. Click **OK**. The **Theme Preview** dialog box will open. An example of the current theme is displayed, complete with navigation buttons and other elements.

Click the **Edit Current Theme** button to make changes to the new theme using the **Theme Properties** dialog box.

To modify a theme

Please note that you cannot edit or delete Doc-To-Help’s built-in themes. Create a new Theme first and customize it.

1. On the **Home** tab, click the drop-down arrow next to the **Theme** button.
2. Choose **Edit Themes**. The **Theme Preview** dialog box will open. An example of the current theme is displayed, complete with navigation buttons and other elements.
3. Click the **Edit Current Theme** button to modify the theme using the **Theme Properties** dialog box. (You can not edit a built-in theme, first add a new one — see "To add a theme" above.)

Note: Themes are stored in \\Program Files\ComponentOne\DocToHelp\Themes in the CHM, NetHelp, or Java folders. It is good practice to back up your custom themes as well as your projects.

Navigation Bar Properties (HTML, Help 2.0, JavaHelp only)

The **Navigation Bar** folder includes sections that allow you to define the background, layout, and next and previous commands.

Please note: The Navigation bar folder is not available for NetHelp themes.

Background Fields

Please note these fields will not be enabled unless the Navigation Bar is enabled in the selected Theme.

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 palettes.

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Layout Fields

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.

Customize Fields

See Including Customized HTML Content in Themes on page 25.

Commands Fields

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 Toolbars Properties (NetHelp Only) (on page 15) **Command Fields** for a full description. All of these controls have the following properties:

Style — Defines the **Previous** or **Next** navigation as text, image or text and image.

Text — Defines text for the **Previous** or **Next** link.

Inherit from project — Determines if the **Previous** or **Next** link text is inherited from the project.

Default style — Determines if the **Previous** or **Next** 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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Disabled picture — Defines the disabled image for the button. Note: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

General Properties (NetHelp Only)

The **General** folder includes sections that allow you to define the background of the NetHelp window.

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.

Toolbars Properties (NetHelp Only)

The **Toolbars** folder 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.

Configuration Options

Please note: every button must belong to a Group.

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.

Use the < and > 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.

Separator — Separates button groups on the selected toolbar. In most cases, use non-breaking spaces; for example,   shows a three-space separator. However, you can use any other string as a separator. The string will be shown on the toolbar between button groups. If the group alignment is set to **Left** or **Center**, the separator will appear to the right of the group. If group alignment is set to **Right**, the separator will appear to the left of the group.

Border — Indicates if the buttons in the group have a border.

Main Toolbar, Left Toolbar, and Topic Toolbar Options

The **Main Toolbar**, **Left Toolbar**, and **Topic Toolbar** folders allow you to modify the layout and background of each toolbar in your NetHelp target. The controls and their properties for these three folders are identical.

Layout Fields

Position — Places the toolbar either at the top or bottom of the pane where it is located. The **Main** toolbar is at the top or bottom of the page. The **Left** toolbar is on top or at the bottom of the navigation pane. The **Topic** toolbar is at the top or bottom of the topic area.

Appearance — There are two appearance options: *Modern* and *Classic*. The Modern toolbar has rounded edges and semi-transparent buttons for button groups that have a border. The Classic has sharp edges and the standard toolbar look.

Border — Determines whether the toolbar has a border.

Three-dimensional buttons — This option is for the *Classic* Appearance only. The buttons will show a 3D effect when the mouse is hovered over them and when a toggle button is selected.

Toolbar height — Specifies the height of the toolbar.

Button width — If the **Fixed button width** checkbox is cleared, 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 Fields

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.

Background picture — Defines the background image in the toolbar area. Click the browse button to access the image directory. Note: Images should be stored in the following folder: ComponentOne\DocToHelp\Themes\Images\Backgrounds.

Repeat — Defines where the image will appear in the background. This option is only available if the **Gradient** checkbox is cleared.

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds.

Commands Fields

The **Commands** folder 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 toolbar buttons can be added/removed and the toolbar location of each command can be changed using the **Configuration Options** of this folder.

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 **Close Window** (**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**.

Synchronize TOC — The default Sync TOC button that appears when the **Synchronize TOC** button is added under the **Configuration Options** of this folder; when this button is added, automatic TOC synchronization is disabled.

Favorites — The **Favorites** pane appears if the **Show Favorites tab** check box is selected in the Windows dialog box (see Setting the Help Window Display on page 176) 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 using the **Configuration Options** of this folder. 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 Messages Properties (NetHelp Only) on page 23 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 it 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.

Breadcrumbs — The **Breadcrumbs** text and/or picture that appears in the Topic toolbar, by default.

Topic Text Properties

The **Topic Text** folder allows you to modify the formatting of the topic pane of your Help target.

Layout Fields

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, clear 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 selected, 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.

Background Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Customize Fields

See Including Customized HTML Content in Themes on page 25.

Breadcrumbs Controls

Breadcrumbs are added by default to HTML Help, NetHelp, Help 2.0, and JavaHelp, but can be disabled in this dialog box.

They will appear above the topic title in HTML Help, Help 2.0, and JavaHelp targets.

In NetHelp, they can be on the topic toolbar, above the topic title, or both. To include breadcrumbs in the Topic toolbar, you must create a custom button with a Breadcrumbs variable in its text. See Toolbars Properties (NetHelp Only) (on page 15) **Configuration Options** for more information about creating a custom button.

Show Breadcrumbs (In NetHelp **Show Breadcrumbs even if not included in Topic toolbar**) — If checked, breadcrumbs will be displayed above the topic title. If your NetHelp Target displays breadcrumbs in the Topic toolbar, you may want to clear this checkbox because the breadcrumbs will appear in two places.

Show separator line under breadcrumbs — If checked, a line will appear under the breadcrumbs.displayed above the topic title. (Does not apply for NetHelp breadcrumbs in the topic toolbar.)

Include current topic in breadcrumbs — If checked, the topic displayed will appear as the last topic in the breadcrumbs. Otherwise, the breadcrumb path will stop at the topic's parent. Please note that top-level topics will not have breadcrumbs.

Label — Text entered here will be displayed before the breadcrumb links (for example, "You are here:"). If the **Inherit from Project** check box is selected, the text specified in the Help Targets dialog box (on page 176)(**Breadcrumbs Label** field) will be used.

Link Separator — Determines the text that will separate breadcrumbs links. Most commonly used separators are: ">", "::", or "|".

Text — Six style controls are available to specify the font and color used for breadcrumbs links. Select the **Use default styles** check box to choose the defaults for all.

ContentsOnly Style — Controls font/color of TOC items with no topic(s) (for example, a top-level heading). These items do not have links.

Selected Style — Enabled if **Include current topic in breadcrumbs** is selected. Controls font/color of last topic displayed in breadcrumbs path.

Normal Style — Controls font/color of breadcrumb links.

Hover Style — Controls font/color of the breadcrumb link when mouse hovers over it.

Label Style — Controls font/color of breadcrumbs label.

Link separator style — Controls font/color of link separator text.

Popup Window Properties

The **Popup Window** folder allows you to modify the formatting of the pop-up windows of your Help target.

Background Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Customize Fields

See Including Customized HTML Content in Themes on page 25.

Secondary Window Properties

The **Secondary Window** folder allows you to modify the formatting of the secondary windows of your Help target.

Background Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Customize Fields

See Including Customized HTML Content in Themes on page 25.

Table of Contents Properties (NetHelp Only)

The **Table of Contents** folder allows you to modify the background and text formatting of the Table of Contents area of your NetHelp target.

Background Controls

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

TOC Item Commands

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Open book picture — Defines the opened image for the TOC. Note: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Topic picture — Defines the topic image for the TOC. Note: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Buttons and Icons

Index Properties (NetHelp Only)

The **Index** folder allows you to modify the background and text formatting of the Index area of your NetHelp target.

Lookup Pane Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Strings Commands

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.

Item List Pane Commands

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Index Item Commands

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.

Search Properties (NetHelp Only)

The **Search** folder allows you to modify the background and text formatting of the Search area of your NetHelp target.

Search Pane Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Strings Commands

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

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Search Result Item Commands

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.

Highlight Commands

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.

Favorites Properties (NetHelp Only)

The **Favorites** folder allows you to specify the appearance of the Favorites pane of your NetHelp target. The Favorites pane is enabled in the **Windows** dialog box using the **Show Favorites tab** check box. See Setting the Help Window Display on page 176.

Favorites Pane Fields

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: Images should be stored in the following folder: \\Program Files\ComponentOne\DocToHelp\Themes\Images\Backgrounds

Background repeat — Defines where the image will appear in the background.

Strings Commands

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.

Favorites Item Fields

Default style — Determines if the Favorites item style is inherited from the project.

Selected style — Defines the text style for Favorites items that have been selected from the listed.

Normal style — Defines the text style for the Favorites items that have not been selected.

Hover style — Defines the hover text style for the Favorites items.

Messages Properties (NetHelp Only)

The **Messages** folder 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.

Message Files Commands

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 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 is set in the **Help Targets** dialog box using the **Email Address** field. See Creating Help Targets on page 155 for more information.

Menu Properties (NetHelp Only)

The **Menu** folder allows you to customize the menu that is displayed when the user clicks a link that has multiple destinations.

Normal Item Commands

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.

Highlighted Item Commands

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.

Accessibility Properties (NetHelp Only)

The **Accessibility** folder 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.

Text Equivalents for Images Commands

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."

Hot Spot Titles Commands

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 C1Hinline Dropdown and C1H Dropdown Text styles. The default is “dropdown text.”

Other Strings Commands

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 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** fields 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 entire contents of the file are 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 must put them in the same directory as 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:

```
<img src=”Customize/mylogo.gif”/>
```

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.

Generating XHTML Output

It is possible to create an XHTML version of NetHelp, HTML Help, Help 2.0, and JavaHelp targets. The XHTML output generated conforms to the W3C XHTML 1.0 strict specification.

Benefits of XHTML

- XHTML Output conforms to the latest Web standards.
- XHTML Output is compatible with XML parsers.

- XHTML gives you the ability to use transforms, giving you infinite customization possibilities.
- An easy way to convert Word and/or HTML to XHTML

To generate a Target as XHTML

1. With your project open, open the **Home** tab.
2. Click the **Target** ribbon group dialog box launcher. The **Help Targets** dialog box will open.
3. Choose any **NetHelp**, **HTML Help**, **Help 2.0**, or **JavaHelp** Target.
4. Select the **Generate XHTML** check box.

   The next time you build the Help Target, the output will be in XHTML.

XML Transforms

Transforms may be added to XHTML output to manipulate content. These transforms can be .NET or XSLT.

This advanced feature makes Doc-To-Help help target creation flexible, allowing for post-build modification in virtually any imaginable custom way; it allows you to modify topics in many ways after the topics are created by Doc-To-Help in the help target.

An example of a possible use of this feature could be adding specific markup to all topics; for example, adding a footer using an HTML table to ensure that the footer is always at the bottom of the browser window, or replacing some text (tag) with variable text depending on a parameter specified in that tag. These are just two examples of adding things to topics or transforming them in a way that can't be done by other means in Doc-To-Help, such as themes and others. Expanding/Collapsing Sections (on page 27) is also an example of an XML transform.

To use transforms, your HTML Help, NetHelp, or Help 2.0 Target output must be in XHTML, see Generating XHTML Output on page 25 for instructions. Transforms can not be used with WinHelp or Manual Targets.

There are two kinds of XML transforms:

- **Programmatic transforms** — These are programs written in any .NET language, such as C# or VB.NET.
- **XSLT transforms** — These are XSLT transformations.

To create a custom transformation, follow the instructions below, and also refer to the two examples provided with Doc-To-Help:

- The expanding sections functionality (see Creating Expanding/Collapsing Sections on page 27) is implemented as an XML transform (a programmatic one). Its full source code can be found in \\Program Files\ComponentOne\DocToHelp\Transforms\BuiltIn\Source\XMLInternal.
- An XSLT transform example can be found in \\Program Files\ComponentOne\DocToHelp\Transforms\Examples (to see how it works in Doc-To-Help, replace the standard config file \\Program Files\ComponentOne\DocToHelp\Transforms\Transforms.config with \\Program Files\ComponentOne\DocToHelp\Transforms\Examples\Transforms.config).

To create a transform, you must write a program (.NET assembly) or an XSLT file and add it to the configuration file in the Doc-To-Help directory. The default path is
\\Program Files\ComponentOne\DocToHelp\Transforms\Transforms.config

Transforms registered in the configuration file are applied to every topic in the order of their appearance in the configuration file.

The configuration file is XML where every transform is registered with a <transform> element with the following elements inside:

- **<description>** — Arbitrary string describing the transform, for explanatory purpose only.
- **<assembly>** — File name of the transform assembly (including the .dll extension) relative to the DocToHelp\Transform directory. For an XSLT transform that element should always be as follows: <assembly>BuiltIn\C1D2HXMLInternal.dll</assembly>
- **<type>** — Full class name in the assembly that implements the IXMLTransform interface. That is the class whose methods are called to perform the transformation. For an XSLT transform that element should always be as follows: <type>C1.D2H.XMLTransform.XSLTTransform</type>
- **<params>** — This element contains whatever elements the transform may need as its parameters. Its contents are passed as they are to the transform method when it is executed. For programmatic transforms, their interpretation is entirely dependent on the code implementing the transform methods. XSLT transforms always have two parameters: <xsl-file> and <copy-files>. The <xsl-file> element contains the name of the XSLT file (with the .xsl extension) relative to the DocToHelp\Transforms directory. The <copy-files> element contains any number (or none) of <copy-file> elements containing file names (relative to the DocToHelp\Transforms directory) that need to be copied to the help target directory to support this transform's functionality (these are usually resource files, such as graphics, scripts, style sheets, etc).

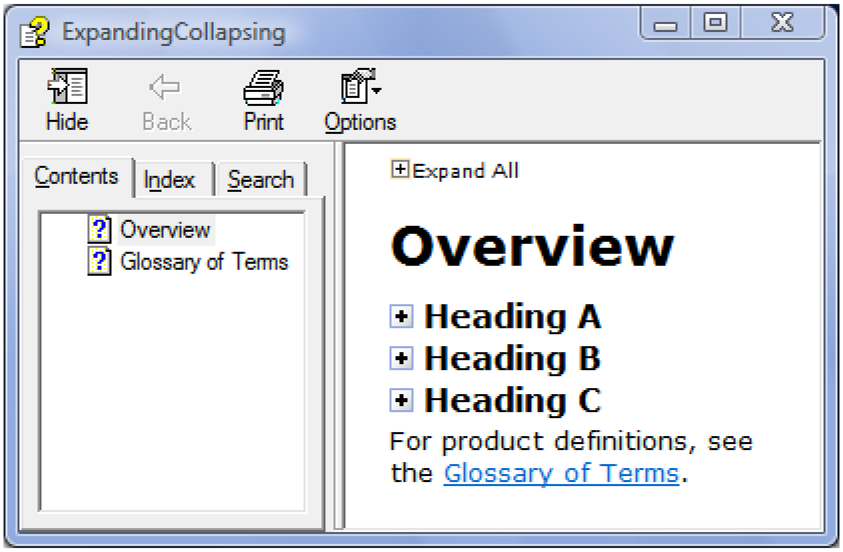
A programmatic transform assembly contains a class implementing the interface C1.D2H.XMLTransform.IXMLTransform. That interface is defined in the C1XHTML.dll assembly residing in the DocToHelp directory. It contains three methods (note that you do not need to implement this interface if you are creating an XSLT transform; creating XSLT transforms requires programming only in XSLT; no .NET programming is required):

- **void IXMLTransform.ReadParams(XmlNode node)**
  This method is called before executing the transform, to initialize the transform class with parameter values specified in the <params> tag in the configuration file. This method is called once in Doc-To-Help build, before this transform is applied to any topics. The 'node' parameter passed to this method contains the XmlDocument whose root element is the <params> element of the configuration file.
- **void IXMLTransform.Execute(XmlDocument doc)**
  This method executes the transform, applies it to a topic. The 'doc' parameter is the topic XML before applying the transform (all transforms preceding thistransform in the configuration file have already been applied to it. This method implementation modifies the XmlDocument passed to it.
- **void IXMLTransform.CopyRequiredFiles(string targetPath)**
  This method is used to copy any files that may be needed in the help target to support this transform's functionality (these are usually resource files, such as graphics, scripts, style sheets, etc). If you don't need any files, leave this method empty. The 'targetPath' parameter contains the path to the TransformFiles subdirectory of the help target directory (for example, MyProject\MyHTMLHelpTarget\TransformFiles). This method implementation copies required files (if any) to that subdirectory (possibly creating subdirectories inside it, if needed).

Creating Expanding/Collapsing Sections

In HTML Help, NetHelp, and Help 2.0 targets, you can create Inline, Dropdown and Popup hot spots (on page 215) using D2HML. You could also create **Expanding/Collapsing Sections**, which are ideal for subdividing a topic into sections. The sections can be displayed as expanded or collapsed by default, and include “+” and “-” icons as indicators for end users next to each section.

This is an example of text collapsed by default; Doc-To-Help automatically adds the “Expand All” control at the top of the topic.



Expanding/Collapsing sections are an XML Transform that can be used in **HTML Help**, **NetHelp**, or **Help 2.0** Target outputs. Your Target output must be in XHTML; see Generating XHTML Output on page 25 for instructions.

Specifying an Expanding or Collapsing section is done by applying Styles to your source documents. These styles are included in the templates included with Doc-To-Help. See Guide to Templates and Styles on page 5 for more information.

These styles are:

- **C1 Section Expanded** (section will be expanded — or open — by default)
- **C1 Section Collapsed** (section will be collapsed — or closed — by default)
- **C1 Section End** (ends an expanding/collapsing section if necessary)

The **C1 Section Expanded** and **C1 Section Collapsed** style names can be changed; they are customizable in the configuration file Transforms.config (see XML Transforms on page 26). Graphics used to indicate expanded/collapsed state of a section — as well as their tooltips — can also be customized by modifying the configuration file Transforms.config.

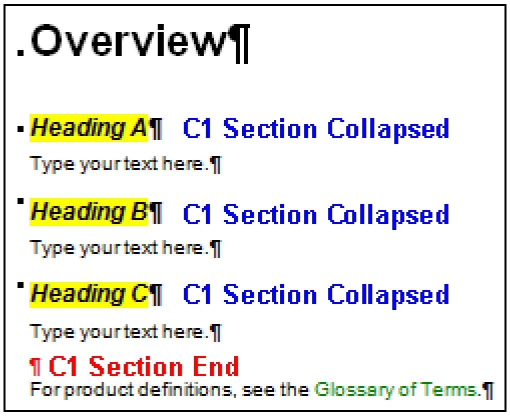
To create expanding/collapsing text

1. If you would like a section to be collapsed by default, apply the **C1 Section Collapsed** style to the Heading of that section. (Do not apply it to the contents of the section.)

   For text expanded by default, apply the **C1 Section Expanded** style to the Heading.

2. Continue applying the appropriate style to the Headings of all the sections you'd like to display as expanded or collapsed.

3. If you would like to display text under expanded or collapsed sections, you need to apply the **C1 Section End** style to end the final section. It can be applied to text, or a blank paragraph above the text (see the example below).



Please note: If you would like to change the look (font, color) of **C1 Section Collapsed** or **C1 Section Expanded** in your output, edit the style in your Target template or style sheet (by default, if you are authoring in Word, this will be **C1H_HTML.dot**, if you are authoring in HTML, **C1H_HTML_full** or **_short.css**). If you would like to change the look in your source document, edit the Source template or style sheet.

Doc-To-Help Outputs and Deliverables

With Doc-To-Help, you can author your documents in your preferred editor and output to several different Targets. This matrix is a handy guide to those Targets, as well as the pros/cons of each Target, the final deliverable files, and the location of those files. Doc-To-Help projects are .D2H files.

Deliverable	Details	Pros	Issues	Deliverable Files	File location
Manual	Can be online (pdf), hard-copy, or both.	**Hardcopy:** Familiar format and navigation. **Online:** Same familiar format, but no printing cost. Can create links to websites, video/audio files, etc. Searchable	**Hardcopy:** Can become obsolete quickly. Printing and delivery costs. Customers may not work in an environment where they can access manual.	.doc and/or .pdf	By default, the **Manual** folder of your project.
HTML Help	Compiled HTML; deliverable is .chm file.	One file; tri-pane interface is easy to use.	Format is 10+ years old. Security issues making it more difficult to use as a deliverable (fixes to XP made .chms unusable over a network)	.chm	By default, the **HTML Help** folder of your project. If **Natural Search** is enabled, you must distribute additional files to the end user. See Natural Search information below.
NetHelp (Browser-based Help)	Uncompiled HTML; deliverable is multiple html files.	Can be delivered on the Web, installed locally, or both. See [NetHelp Installation Requirements] on page 33 for more information. Delivering on the Web makes a continuous publishing model possible. The user's web browser is the Help window.	Number of files to distribute is daunting to some.	Multiple .htm/.html files. Default home page is **default.htm**. (You can change the default page name in the **Help Targets** dialog box)	By default the **NetHelp** folder of your project. The entire contents (including subfolders) of this folder must be included with your software application.

Deliverable	Details	Pros	Issues	Deliverable Files	File location
Notes on NetHelp			NetHelp may not display correctly initially because Windows XP SP2 and higher automatically disables active content (JavaScript). To disable this security block for all local content, in Internet Explorer, go to **Tools > Internet Options > Advanced > Security**. Select the **Allow active content to run in files on My Computer** check box. Users can also disable the block within the browser.		
WinHelp	Compiled; deliverable is .hlp and .cnt file.	None. If your product is a legacy one with no plans to change Help format (or work in Vista), you will be able to continue producing it.	Original Help format. Look and navigation not as user-friendly as newer formats. Very difficult for customers to use with Windows Vista (they must download the viewer; it can't be distributed with WinHelp files) See this article http://support.microsoft.com/kb/917607	.hlp and .cnt	By default, the **Help** folder of your project. If **Natural Search** is enabled, you must distribute additional files to the end user. See Natural Search Requirements on page 37 for more information.
JavaHelp	Java Help 1.1.3 is supported, as well as JavaHelp 2.0	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.	Multiple files; deliver the entire **JavaHelp** folder (its default name) in your project directory.	By Default, the **JavaHelp** folder of your project. To view a JavaHelp HelpSet, you must have the necessary file installed on your machine. See "Notes on JavaHelp" below

<table>
<tr><th>Deliverable</th><th>Details</th><th>Pros</th><th>Issues</th><th>Deliverable Files</th><th>File location</th></tr>
<tr><td colspan="3">Notes on JavaHelp</td><td colspan="3">
To build and view JavaHelp, you must first install the necessary files from Sun.
Do the following:

Install the Java Runtime Environment (known as JRE). If you don’t already have it, download it from http://www.java.com/en/download/manual.jsp
Download and unzip the JavaHelp files from http://java.sun.com/javase/technologies/desktop/javahelp/download_binary.html.
Update your Environment Variables to tell your machine where to find the Java files.

Go to Control Panel > System > Advanced Settings. The System Properties dialog box will open.

On the Advanced tab click the Environment Variables button. These variables need to be added or changed under System variables:

JAVA_HOME
Value: C:\Program Files\Java\jre1.6.0 (use the location of your JRE installation)
JAVAHELP_HOME
Value: C:\Program Files\Java\jh2.0 (use the location of your jh2.0 directory)
JHHOME
Value: C:\Program Files\Java\jh2.0 (use the location of your jh2.0 directory)
Path
This one will already exist, so edit it and append ;C:\Program Files\Java\jre1.6.0\bin (use the location of your JRE\bin directory)
(Note: semicolons separate multiple paths)

As noted above, the exact file paths will vary based on Java version numbers and where you have JavaHelp and JRE installed on your machine.
For more information, see Sun's JavaHelp Web site at: http://java.sun.com/products/javahelp/
For more on JavaHelp redistributables, see http://java.sun.com/products/javahelp/README.html#redistributable
</td></tr>
</table>

Deliverable	Details	Pros	Issues	Deliverable Files	File location
Help 2.0	Help authoring format for those developing Help for Visual Studio.Net only. Deliverable can be the reference documentation only, or Doc-To-Help can be used to add narrative text. Microsoft Sandcastle is used to generate reference documentation.	Resembles Microsoft MSDN collection and uses some of the same underlying technology	Format can only be included in Help for Visual Studio.NET. Programmers who distribute .NET components can integrate Help 2.0, because the .NET tools include the run-time components necessary to view Help 2.0 Help systems.	.HxT, NamedUrlindex.HxK, ProjectName_K.HxK, ProjectName_F.HxK, ProjectName_A.HxK, .HxS, .hxc, HxA H2Reg_cmd.ini Your deliverables may vary based on your application. For more on Help 2.0 redistributables, see http://www.helpware.net/mshelp2/h2reg.htm	By default, the **MSHelp** folder of your project.
Notes on Help 2.0			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). The Namespace and Parent Namespace are set in the Help Targets dialog box. See Help 2.0 Target on page 156 for more information.		

Watch the video: How to Deliver a Target (1:01) [http://helpcentral.componentone.com/videos/D2H2009_Tour/DeliveringATarget.html]

NetHelp Installation Requirements

NetHelp can be installed locally (on the end user’s machine), or it can be deployed on a web server.

Supported browsers:

- Internet Explorer 6.0 or higher
- Netscape 6.2.3 or higher
- Firefox (all versions)
- Mozilla 1.2 or higher
- Opera 7.54 or higher
- Apple Safari 4.0.2 or higher
- Google Chrome 2.0.172.33 or higher

Please note: In NetHelp targets, you can specify a **Java** or **JavaScript** search. The search option is set using the **Search Type** field in the Help Targets dialog box (on page 155). If NetHelp will be installed locally, JavaScript search is recommended to avoid dependence on Java being installed on end-user machines. JavaScript search supports wildcards * (any substring) and ? (any character). If NetHelp will be installed on a server, Java search is recommended because the search will be slightly faster, and yield higher-quality results. NetHelp deployed on a server does not require that Java be installed on end-user machines.

NetHelp Local Installation

Installing NetHelp on a client machine

1. Build the NetHelp target within Doc-To-Help.
2. Copy the contents of the NetHelp folder to the desired directory.

   You can also set the NetHelp destination folder using the Folder field (on page 167) of the Help Targets dialog box (on page 155).

Viewing NetHelp locally

- Open the **default.htm** file located in the directory where your NetHelp output is installed.
- If the NetHelp project is using **Java** search, the client machine must have Sun Java Runtime Environment (JRE) installed. The Java software is available at http://www.java.com/en/download/index.jsp. (If the project uses a **JavaScript** search this is not necessary). This option is set using the **Search Type** field in the **Help Targets** dialog box. See NetHelp Target on page 167 for more information.

NetHelp Server Installation

NetHelp may be installed on a Microsoft Internet Information Server (IIS) or on a non-IIS server. If you will be using a non-IIS server, it must support the Java Servlet API in order to run NetHelp. See **Installing NetHelp on a Java servlet-enabled web server** below.

IIS Requirements:

- Windows 2000 Server or later
- IIS 5.0 or later
- Sun Java Runtime Environment (JRE) (Only required if the NetHelp project is using **Java** search, if it is using **JavaScript** search this is not necessary. This option is set using the **Search Type** field in the **Help Targets** dialog box. See NetHelp Target on page 167 for more information.)

To install NetHelp on Microsoft Internet Information Server

1. Create an IIS virtual directory for each NetHelp target you want to deploy.
2. If the project is using the Java search option, make sure the Sun Java Runtime Environment (JRE) is installed on the server machine. Go to http://www.java.com/en/download/index.jsp to download the Java software.
3. Copy C1D2HASPHandler.dll from the NetHelp subdirectory of the D2H installation directory to the server machine and register it with 'regsvr32 C1D2HASPHandler.dll'. The destination directory does not matter, but the directory where NetHelp is deployed is the best choice.
4. Modify the platform.js file in the NetHelp directory. By default, this file contains var d2hServerPlatform = "jsp"; meaning that Search uses the Java servlet. Change it to var d2hServerPlatform = "asp"; to use ASP instead.

To view NetHelp over the Web

- Browse to the **default.htm** file in your virtual directory.
- It is not required that the Java Runtime Environment be installed on the client machine.

Search Troubleshooting

If the search functionality is not working properly, try the following.

- Re-register **C1D2HASPHandler.dll**.
- The search will not work if there are any documents in the document tree that are not .htm, .html, .xml, or .doc. Remove the appropriate documents. Note that this will not remove links to those documents from the Help file.

To install NetHelp on a Java servlet-enabled web server

Deploying NetHelp on any Java-enabled Web server (such as Tomcat, Resin, etc.) requires publishing Web pages on the server. Follow your Web server instructions for publishing content.

Example: Your compiled NetHelp project is in the ...\MyProject\NetHelp folder and you want to publish to a Jakarta Tomcat server. If the Tomcat folder is ...\jakarta-tomcat-5, copy the NetHelp folder to ...\jakarta-tomcat-5\webapps. Start Tomcat, run ...\jakarta-tomcat-5\startup.bat. NetHelp will open in your browser. Navigate to **http://localhost:8080/NetHelp/default.htm** on your server to view NetHelp.

Section 508 Compliant NetHelp

A Doc-To-Help NetHelp project can be configured to be Section 508 compliant.

Section 508 is part of the Rehabilitation Act of 1973, and requires Federal agencies to make their electronic and information technology accessible to people with disabilities. To find out more about Section 508, and obtain information about guidelines, tools and other resources, see http://www.section508.gov/.

Once you have set up your project to be Section 508 compliant, and provided alternative text where required (for example, images and tables), you should test your source documents and project with one of the third-party tools available for verifying accessibility.

To enable Section 508 compliance

1. Open the Help Targets dialog box (see Creating Help Targets on page 155).
2. Choose your NetHelp project from the panel on the left.
3. In the Advanced section, go to the Accessibility mode field and choose Section 508.

The following features are enabled when the Accessibility mode is set to **Section 508**:

- All links generated by Doc-To-Help have title strings (indicating 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 Accessibility properties. See the Hot Spot Titles Commands on page 24.
- Icons in the table of contents have titles (their text equivalents) indicating whether the item is a book or a topic. When the Dynamic table of contents check box is selected in the Help Targets dialog box (on page 155) (NetHelp Target) and therefore the icon is a book, this title indicates whether it is open or closed. These strings can be changed in the Theme Designer Accessibility properties. See the Text Equivalents for Images Commands on page 24.
- 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 destination topics, 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 associated with the group or keyword. The page heading can be changed in the Theme Designer Accessibility Properties. See the Other Strings Commands and specify the text in the Keyword/group link found multiple topics field (on page 24).

The following features are enabled, regardless of the Accessibility mode chosen:

- All links and buttons in NetHelp themes are accessible from the keyboard using the Tab key.
- When the Dynamic table of contents check box is selected 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. 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. These strings can be changed in the Theme Designer Accessibility properties. See the Other Strings Commands on page 24.

Note: For more information about the Theme Designer, see Customizing with the Theme Designer on page 11. For more on the Accessibility properties, see Accessibility Properties (NetHelp only) on page 24.

To create alternative text for images

In order for your project to be Section 508 compliant, all images in your documents must have alternative text, which is a description that can be read by accessibility devices. In HTML source documents, set the appropriate image options for your editor (<img src> *alt* attribute). In .xml documents, you can specify this text in the **Alternative Text** field when inserting the image in the XHTML editor with the **Picture Properties** dialog box (for movies, the **Alternative Text** field in the **Movie in Flash Format Properties** dialog box). In Microsoft® Word source documents do the following:

1. Right-click on the image and select Format Picture from the menu. The Format Picture dialog box will open.
2. In Word 2003, click the Web tab and enter the text in the Alternative text text box. (In Word 2007, the tab is named Alt Text.)
3. Click OK.

Doc-To-Help will provide warnings in the build log if any images without alternative text are found in the Word source documents of your project. These warnings can be ignored if certain images do not need alternative text and will not interfere with accessibility. Doc-To-Help inserts an empty alternative text tag in the NetHelp target if text is not provided.

To create table captions and summaries

Tables must have a caption and summary specified in the HTML in order to be Section 508 compliant.

The caption is specified using the **<caption>** tag, which must appear within the **<table>** tag. It is displayed in the target and read by accessibility devices. Although Word displays the caption as normal text with no special formatting, Doc-To-Help places the **<caption>** tag inside the **<table>** tag when NetHelp is generated.

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 is not displayed in the target, but is read by accessibility devices. Although Word does not have a feature for specifying a table summary, this can be done by inserting a comment within a table. Doc-To-Help will interpret the comment as the table's summary.

In HTML source documents, set the appropriate options for your editor. In .xml documents, you can specify this text in the **Caption** and **Summary** fields when inserting the table in the XHTML editor with the **Table Properties** dialog box. In Microsoft® Word source documents do the following:

Adding a caption to a table

1. Select the entire table and right-click.
2. In Word 2003, choose **Caption** from the menu. (In Word 2007, the menu item is named **Insert Caption**.) The **Caption** dialog box will open.
3. Click the **New Label** button. Enter the caption and click **OK**.
4. Click **OK** to close the **Caption** dialog box.

Adding a comment to a table (this will become the summary)

1. Select the entire table or place your cursor within a cell.
2. In Word 2003, choose **Insert > Comment.** (In Word 2007, click the **Review** tab and click the **New Comment** button.)
3. Enter the comment.

Doc-To-Help will provide warnings in the build log if any tables without captions and summaries are found in the Word source documents of your project. These warnings can be ignored if certain tables do not need captions and summaries and will not interfere with accessibility. Doc-To-Help inserts an empty caption and summary in the NetHelp target if text is not provided.

Natural Search Requirements

If you have enabled **Natural Search** (using the Help Targets dialog box on page 155) for a WinHelp or HTML Help project, you need to deliver the following files along with your help files. They will have the same name as your help project, and should be installed in the same directory in your application. These files can be found in the same directory folder as your Help project.

- Index file: **<project name>.smi**
- Noise word list: **<project name>.snz**
- Configuration file: **<project name>.cfg**

The **stoplist.txt file** can also be found in the same directory as your Help project and you may edit it. “Stop words” (or “Noise Words”) are those that are skipped in a search because they are so common (a, an, at, etc.) For information on editing stoplist.txt, see NetHelp Target on page 167.

In addition, the Natural Search Engine needs to be installed on **end user** machines. The Natural Search **setup.exe** program is 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.

Source Document locations

By default, Doc-To-Help will store your projects in the **My Doc-To-Help Projects** folder.

The **My Doc-To-Help Projects** folder can be found in Windows® Vista at \\Users\<username>\Documents\My Doc-To-Help Projects; in Windows XP at \\Documents and Settings\<username>\My Documents\My Doc-To-Help Projects. To change the default location for Doc-To-Help projects, use the **Options** dialog box. Choose the **Doc-To-Help button**  > **Tools > Options**. The **Options** dialog box will open. (Or, click the **Doc-To-Help button** and click the **Doc-To-Help Options** button.) Select the **File Locations** tab. See Setting Doc-To-Help Options (on page 45) for more information.

Within your Doc-To-Help project folder:

- Word, HTML, and/or XHTML (.xml) source documents will be located by default in the **Documents** folder.
- Graphics (images, movies, etc.) should be stored in the **Media** folder.

These locations can be confirmed in the Project Settings dialog box (on page 196).

If you are using Team Authoring, you should store your graphics in the root of the **Documents** folder. This will enable you to manage your graphics and other auxiliary files within Team Authoring.

Template and Style Sheet locations

Doc-To-Help Source and Target **Templates** are stored in Windows® Vista at C:\Users\ (user name)\AppData\Roaming\Microsoft\Templates; in Windows XP at C:\Documents and Settings\(user name)\Application Data\Microsoft\Templates. If you uninstall Doc-To-Help and reinstall a newer version, the templates (including customized ones) will be saved to the \Backup folder of those directories (Doc-To-Help will save the last ten backups).

The Application Data or App Data folder is a hidden folder. To show hidden folders, open the **Folder Options** dialog box in your operating system (**Control Panel** > **Folder Options**). Click the **View** tab, under **Advanced Settings** > **Files and folders** > **Hidden files and folders**, select the **Show hidden files and folders** radio button. Click **OK**.

For more on Doc-To-Help Source and Target **Style Sheets** storage, see HTML File Style Sheets on page 10.

Licensing and Technical Issues

Answers to questions about your Doc-To-Help license, technical support, system requirements, installation, and other issues can be found here.

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 <u>may not be disclosed or distributed</u> 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.**

2. **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.

1. **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.

2. **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.

3. **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 2009 is developed by ComponentOne LLC. You may distribute the following output files, royalty free, with any online Help system or printed manual you develop/author:

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

You may **not** redistribute any file not cited above.

End-users of your online Help systems and printed manuals are **not** licensed to use Doc-To-Help for authoring purposes. Such users must first obtain a license in order to be allowed such use.

You are **not** licensed to distribute Doc-To-Help to any users for purposes of allowing such users to edit, modify, or alter other existing online Help systems or printed manuals. You are **not** allowed to add or transfer the Doc-To-Help serial number to the registry of your users' computer(s).

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

Technical Support

Doc-To-Help 2009 is developed and supported by ComponentOne LLC.

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/SuperProducts/SupportServices/Options/.

Introductory Telephone Support

Standard Introductory support includes thirty (30) days of telephone support for Doc-To-Help customers. This service starts from the date of purchase. The Support phone number is 412-681-4738.

Support Plans

Telephone and online support is included with the Doc-To-Help 2009 Gold and Platinum Support plans.

Online support provides you with direct access to our Technical Support staff via an online incident submission form. When you submit an incident, you'll receive a response via e-mail confirming that you've successfully created an incident. This email will provide you with an Issue Reference ID. You will receive a response from a ComponentOne staff member via e-mail in 2 business days or less. The submission URL is http://www.componentone.com/Support/SubmitIncident.aspx.

We also offer annual or per-incident add-on telephone support. If you have not already purchased support for your Doc-To-Help license and are interested in doing so, please contact our sales team by email (sales@doctohelp.com) or telephone (1-800-858-2739 or 1-412-681-4343).

System Requirements

Computer/Processor	PC with Intel® Pentium® II 500 MHz or greater processor
Memory	512 MB of RAM Minimum
Hard Disk Space	125 MB of hard disk space for the Doc-To-Help application and related files
	13 MB of hard disk space for MDAC 2.6
Operating System	Microsoft® Windows XP (with Service Pack 2), Microsoft® Vista, or Microsoft® Windows 7
Microsoft Word	Microsoft® Word 2000 (version 9.0) or greater, Word 2003, or Word 2007
Microsoft FrontPage	Microsoft® FrontPage® XP, 2003
Adobe Dreamweaver	Adobe® Dreamweaver® MX2004, 8.0, 9.0, CS3
.NET Framework	Microsoft® .NET framework version 2.0
Microsoft Data Access Components (MDAC)	MDAC version 2.6 or later

Microsoft® Word, FrontPage®, and Adobe® Dreamweaver® are required only if you plan to use them as editors.

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 (http://www.componentone.com/SuperProducts/SupportServices/).

Installing Doc-To-Help

Prerequisites and installation notes

See System Requirements on page 44 for a complete list of software and hardware requirements.

- Doc-To-Help requires the Microsoft® .NET Framework 2.0. The Doc-To-Help installation will check for the presence of this Framework. If it is not found, you can visit Microsoft's web site for a free download. (If you purchased Doc-To-Help on CD, the Framework will also be available there.)
- Close Microsoft® Word before beginning Doc-To-Help installation. Microsoft® Outlook should also be closed if Microsoft Word is your default email editor.
- If you have projects, templates, and style sheets from existing Doc-To-Help projects, it is recommended that you back up those files before installing Doc-To-Help.

Installation Instructions

1. Insert the Doc-To-Help CD into your CD-ROM drive. If Autorun is setup on your system, skip step 2.
2. Select and run the **Setup.exe** file from your CD drive.
3. Choose the **Install** option.
4. Follow the wizard to complete the installation.

Using Live Update

Live Update is a utility that automatically checks for software updates every time you open Doc-To-Help.

If a new version is available, the **ComponentOne Doc-To-Help Live Update** dialog box will open. The update number and new feature list will be displayed. Click the **Download** button to download the update.

To disable Live Update

1. Choose the Doc-To-Help button > Tools > Options. The Options dialog box will open.
2. Click the **Live Update** tab.
3. Select the **Do not check for updates automatically** radio button.
4. Click **OK**.

To check for updates manually

Choose the **Doc-To-Help button** > **Tools** > **Check for Updates**.

Deactivating Doc-To-Help

If you need to move your copy of Doc-To-Help from the computer it is currently installed on to another one, you must deactivate the license before uninstalling. Please note that once you have deactivated Doc-To-Help, it will not be useable. It will resume working once the license is installed on another computer and activated.

For more information on activating/deactivating Doc-To-Help, see the Activation Overview web page.

To deactivate Doc-To-Help

Choose the **Doc-To-Help button** > **Tools** > **Deactivate**.

Setting Doc-To-Help Options

You can set Doc-To-Help project options for Startup, file locations, preferred HTML editors, updates, spelling, etc. using the **Options** dialog box.

General tab:

- Set Startup options for the **Getting Started With Doc-To-Help** wizard and automatic reloading of last project
- Change the color scheme for Doc-To-Help (Blue, Silver, or Black)
- Set various confirmation prompts. If these check boxes are selected, Doc-To-Help will display a message box when you perform these functions. These messages can also be turned off by selecting the "Don't show me this again" check box when the message box is displayed.

- **Confirm editing properties of multiple topics**. If you select more than one topic in the **Topics** window, and then click the **Properties** button in the **Topics** ribbon, Doc-To-Help will remind you that you have selected more than one topic and will be editing the properties for all of them.
- **Confirm adding topics outside source document folders**. If adding a document to your project using drag-and-drop, Doc-To-Help will inform you if the document is not stored in the correct folder. You should always copy documents to the correct folder before adding them so that the links don't break and the project is self-contained.
- **Confirm making a copy of a document with Save As**. If saving the currently open XML document under another name using **Doc-To-Help button > Save As**, Doc-To-Help will explain that the document will be stored in the same folder as the original document, but will not be automatically added to the project.
- **Confirm using Windows System colors**. If you change the System background color for any JavaHelp or NetHelp Target (**Help Targets dialog box > Color > System tab**), Doc-To-Help will inform you that system colors are specific to Windows and explain how this will be handled for the Target.
- **Prompt viewing help target after build**. After every build or rebuild, Doc-To-Help will ask if you would like to view the Help Target.
- **Confirm allowing local formatting**. If you click the **Local Formatting** button in the **Editor tab** of the XHTML editor, Doc-To-Help will display additional information about local formatting.
- **Confirm removing local formatting**. If you have applied local formatting in the XHTML editor, and then click the **Local Formatting** button in the **Editor tab**, Doc-To-Help will inform you that all the local formatting applied will be removed.

File Locations tab:

- Change the default location for Doc-To-Help projects.

HTML Editors tab:

- Set the default HTML editor for editing your Doc-To-Help HTML documents. You can add and remove editors from the list.

Team Authoring tab:

- Change the default “new project” colors for your Team Projects.

Live Update tab:

- Turn the Doc-To-Help **Live Update** feature on and off. See Using Live Update on page 45 for more information.

Spelling tab:

- Set the spell check options for the Doc-To-Help **XHTML** editor. (The **Spelling** button is on the **Editor** tab. See Editor tab on page 99 for more information.)

To set Doc-To-Help Options

1. Choose the **Doc-To-Help button** > **Tools** > **Options**. The **Options** dialog box will open. (Or, click the **Doc-To-Help** button and click the **Doc-To-Help Options** button.)
2. Select the **General**, **File Locations**, **HTML Editors**, **Team Authoring**, **Live Update**, or **Spelling** tab.
3. Select the desired options.
4. Click **OK**.

Trademark Statement

Doc-To-Help is a registered trademark of ComponentOne, LLC. Microsoft, FrontPage, Vista, NT, XP, and Windows are either registered trademarks or trademarks of Microsoft Corporation. Adobe, Dreamweaver, and RoboHelp are either registered trademarks or trademarks of Adobe Systems Corporation. Intel and Pentium are either registered trademarks or trademarks of Intel Corporation. Java is a registered trademark of Sun Microsystems, Inc. Google Chrome is a trademark of Google. Safari is a registered trademark of Apple, Inc. All other product and company names herein are the property of their respective owners.

Doc-To-Help Guided Tour

Doc-To-Help 2009 has an easy-to-use interface that incorporates ribbons, navigation panes, and tabbed windows. See the video at http://helpcentral.componentone.com/videos/D2H2009_Tour/D2H2009_Tour.html for a tour of Doc-To-Help's interface and features.

This video provides an overview of Doc-To-Help and its features: http://www.componentone.com/newimages/products/videos/doctohelp/onebox.html.

Sample Doc-To-Help projects can be found in the following location. These projects can be used to test drive Doc-To-Help features and functionality.
Windows® Vista: \\Users\<username>\Documents\My Doc-To-Help Projects\Samples
Windows XP: \\Documents and Settings\<username>\My Documents\My Doc-To-Help Projects\Samples

Also see Navigating Doc-To-Help on page 95 for details on each feature.

Doc-To-Help Terminology

Following are a list of common terms used within Doc-To-Help that you should be familiar with:

Target — Doc-To-Help has six default Help Targets. They are all specific output types:

- Help 2.0
- HTML Help
- JavaHelp
- Manual
- NetHelp
- WinHelp

See Introduction to Single Sourcing on page 1 for more information about each output type.

You may create multiple Help Targets of the same type, or rename the default Target names. You can also remove Targets from your project to streamline it. See Creating Help Targets on page 155 for more information.

See Doc-To-Help Outputs and Deliverables on page 30 for a complete guide to targets and deliverables.

Building a Target — When you build a Target, you create your output. After you have built a Target, you can view it and deliver it. See Building a Target on page 235 for complete instructions.

Source documents — In Doc-To-Help, you work (edit, apply styles, create links, etc.) in your source documents, which can be Microsoft® Word, HTML, or XML documents. A project can contain a mixture of Word, HTML, and XML documents if you wish.

Paragraph Styles, **Character Styles**, and **Topic Types** — Doc-To-Help uses styles to control both the look and behavior of your output. See Guide to Templates and Styles on page 5 and Defining Character/Paragraph Styles and Topic Types on page 181 for more information.

Quick Reference Page

Doc-To-Help Workflow

Following is an outline of a standard Doc-To-Help workflow.

Before beginning a project in Doc-To-Help, you may want to take a look at Navigating Doc-To-Help (on page 95) for a quick overview of the interface.

1. Create your project

You can create a project 3 ways. If you have existing documents, Wizards make it easy to import them.

1. Use the **Getting Started Wizard** that displays when you open Doc-To-Help. Choose **Start a New Project** to begin.

2. If Doc-To-Help is already open, click the **Doc-To-Help button** and choose **New Project**. The **New Project Wizard** will guide you through the process.
3. Convert a project created in another application by choosing **Convert an Existing Project to a Doc-To-Help Project** in the **Getting Started Wizard**, or click the **Doc-To-Help button** and choose **Convert**.

When you create a new project using a wizard, the default project settings will be applied, so you can create your project right out of the box – however, changing these settings gives you maximum flexibility to create the output the want, which looks and behaves exactly how you want it to.

- See Creating and Converting Projects on page 139 for more information.

2. Specify your look and feel

This is where you customize your project.

The templates and style sheets chosen for your project control the look of your final Targets. Templates are used for Word documents and style sheets are used for HTML documents. The templates/style sheets chosen can be changed at any time, and can also be edited to your specifications. They are a starting point that provides plenty of flexibility for your creativity.

Please see Guide to Templates and Styles on page 5 for more information about applying and editing templates and style sheets.

Four dialog boxes — **Project Settings**, **Help Targets**, **Project Styles**, and **Windows** are also instrumental in setting your project up and specifying the behavior of your final Targets.

- See Setting Project Properties (on page 196) for information on default source folder locations, context ID settings, keeping page breaks in manual targets and more.
- See Creating Help Targets (on page 155) to name and setup each of your targets, as well as delete the ones you don't need.
- See Defining Character/Paragraph Styles and Topic Types (on page 181) for information on setting up styles and topic types. You can set up auto indexing and auto context IDs for topic types, the hierarchy levels for paragraph styles, and the hotspot type for character styles, just to name a few common functions.
- See Setting the Help Window Display (on page 176) for information on setting the size, position, and navigation for your online Help Target windows.

3. Create content and apply styles

As you enter content in your source documents, you will apply styles to it that will determine the final look of your Targets, as well as how Doc-To-Help will process your documents. In addition to the pre-defined styles you can apply from the templates and style sheets (Heading 1, C1H Number, C1H Bullet, Table Heading, Table Text, etc.), D2HML Styles (applied with a click from the **Doc-To-Help** or **D2HML** toolbar or ribbon) make it possible to create topic links, expanding/dropdown/popup text, apply conditions and insert variables. D2HML makes it easy to create full-featured Help Targets in your favorite editor. It also makes single sourcing simple.

You can also define your own styles if you wish.

For more information, see:

- Introduction to Single Sourcing on page 1
- Working with Source Documents on page 115
- Using D2HML (Doc-To-Help Markup Language) on page 210

- Variables Window on page 114
- Marking Text as Conditional on page 220

4. Manage Topics

Once you have created content, you can check out the Topics window (on page 112) to work more closely with your topics. This window is key to creating your table of contents, index, and related topics — as well as assigning/editing context IDs for context-sensitive Help if you need them. Simply right-click on any topic in the **Topics window** and choose **Properties** to view its properties.

For more information, see:

- Managing Topics on page 225
- Creating an Index or Groups on page 231
- Creating a Table of Contents on page 232
- Creating a Glossary on page 138
- Implementing Context Sensitive Help on page 200

5. Build your project

Of course, you can build your project at any time to take a look at, but it is the final step in producing quality output.

See Building a Target on page 235 for a quick overview of choosing and creating a Target.

See Doc-To-Help Outputs and Deliverables on page 30 for a matrix that details which files need to be delivered for each Target.

Other features

Doc-To-Help provides many other advanced features, such as Team Authoring (Doc-To-Help Enterprise only), creating Modular Help systems, generating XHTML output, and documenting class libraries with Microsoft® Sandcastle.

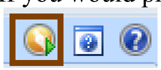
For more information, see:

- Working on a Team on page 241
- Creating a Modular Help System on page 257
- Generating XHTML Output on page 25
- Documenting Your Class Library with Microsoft Sandcastle on page 265

Doc-To-Help Quick Tour

If you haven't used Doc-To-Help before (welcome!), you may want to take a minute and review [Doc-To-Help Terminology]() on page 49. This Quick Tour uses Microsoft® Word as the document editor, but the same principles apply if you are using Doc-To-Help's built-in XHTML editor or an HTML editor.

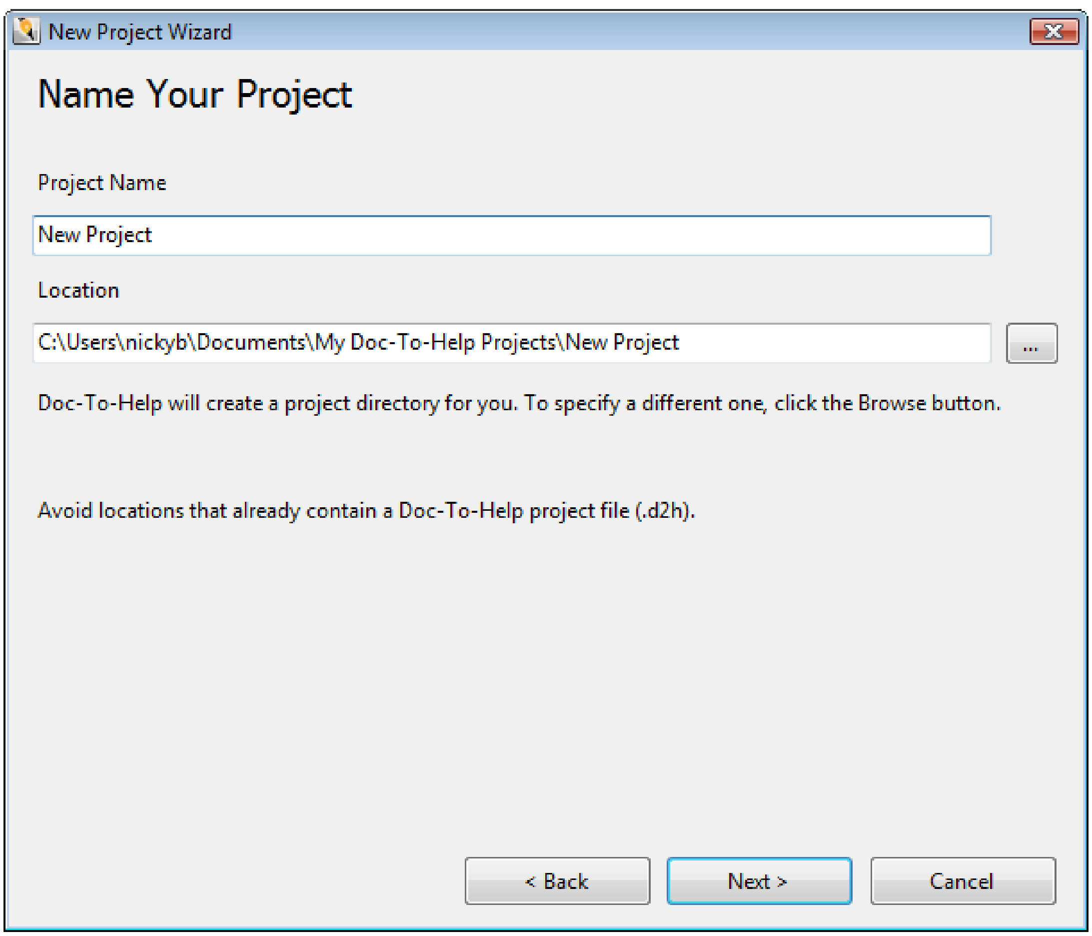
As you are working through this Quick Tour, you may want to learn more about the features discussed — if so, please refer to the [Dynamic Help window]() on page 114.

Note: If you would prefer to begin the Quick Tour from the **Getting Started Wizard**, click the **Getting Started Wizard** button  (on the upper right) to open it, and choose the **Start a New Project** option.

 **Watch the video:** [Starting a New Project in Doc-To-Help]() (3:21) [http://helpcentral.componentone.com/videos/D2H2009_Tour/StartingANewProject.html]

Let's get started …

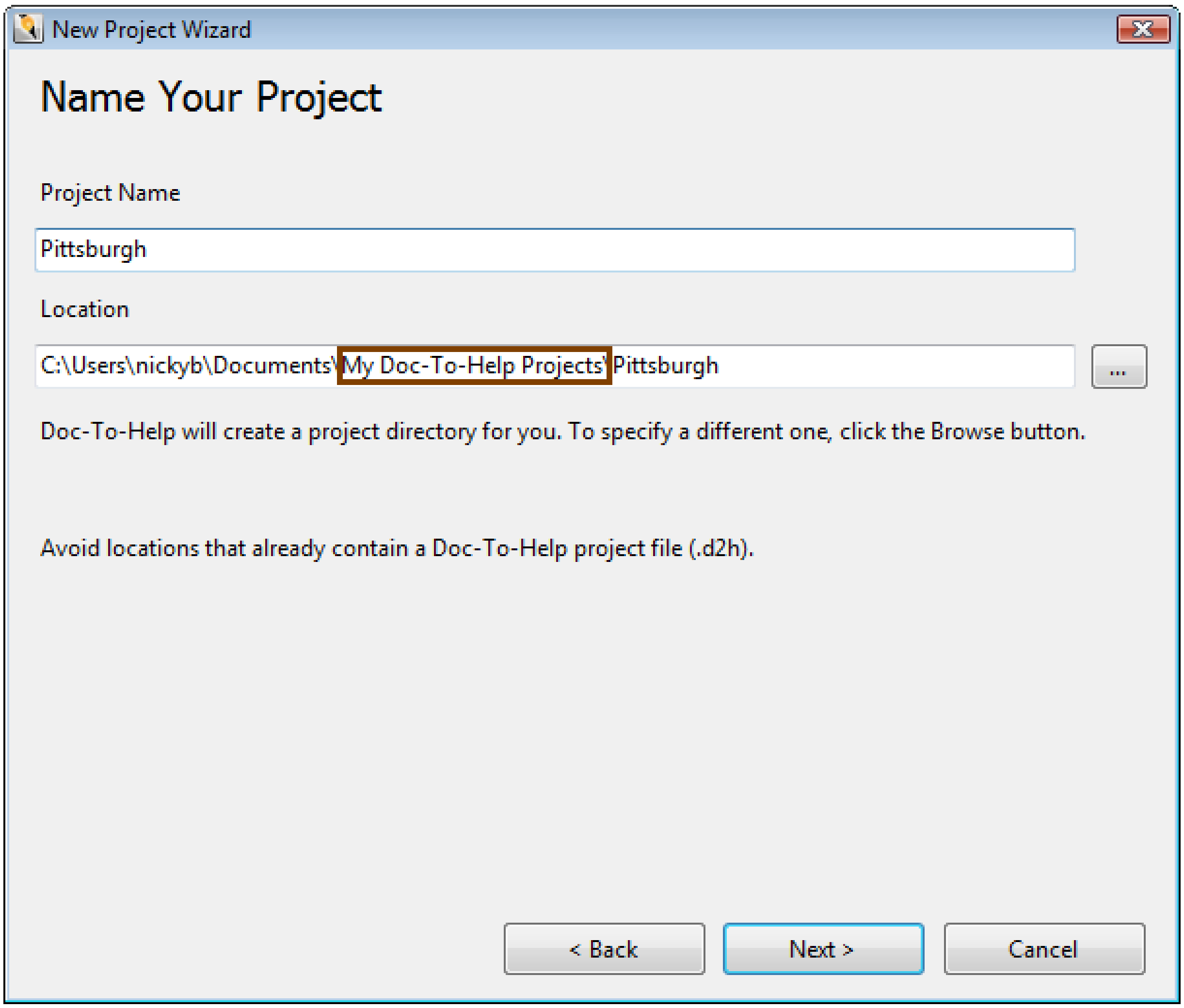
Click the **Doc-To-Help button**  > **New Project**. The **New Project Wizard** will open.



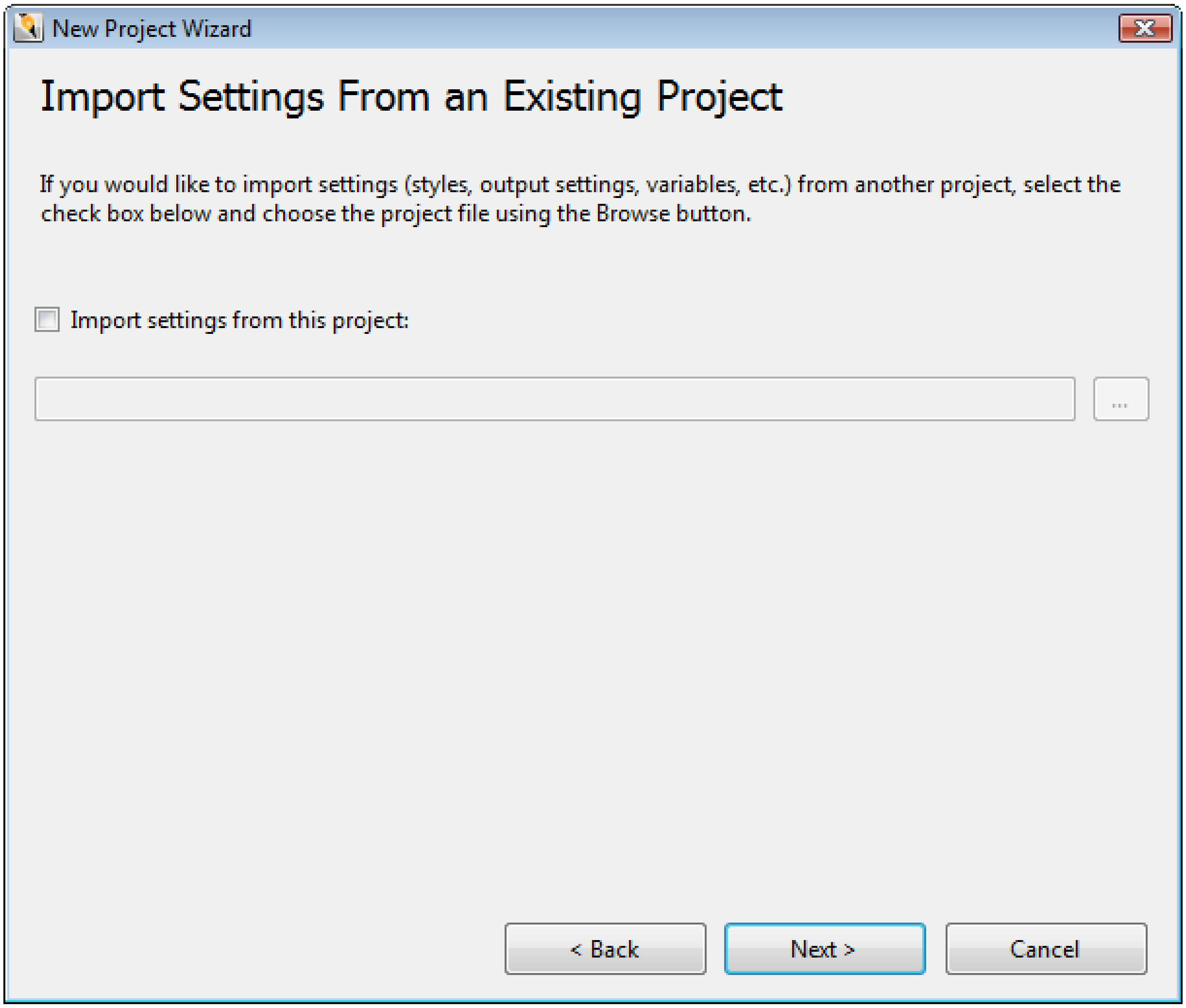
Name your project **Pittsburgh**. Doc-To-Help will automatically name the project location folder with the same name.

Note that Doc-To-Help saves your project in the default **My Doc-To-Help Projects folder**. This makes it easy to keep your projects organized.

(You can change the default folder if you wish, just click the **Doc-To-Help button**  > **Doc-To-Help Options** button. Choose the **File Locations** tab.)

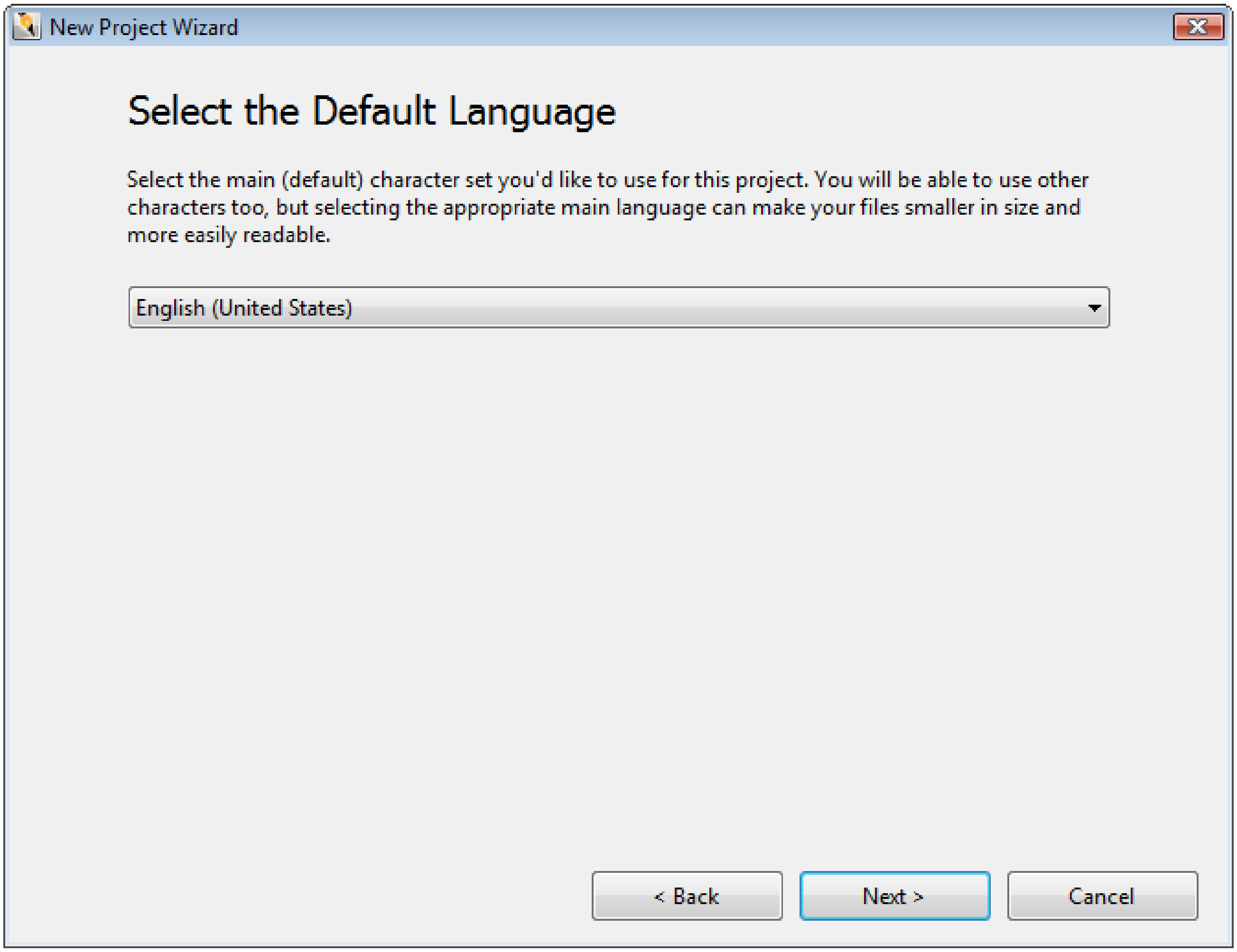


Click **Next**. The next screen will give you the opportunity to import settings from an existing project. Since we are going to work through a few settings later, we will skip this screen.



Click **Next**. Now, you can select the default language for your project. By default, it is U.S. English, but you can choose another from the drop-down.

Doc-To-Help will choose the correct character set based on the language chosen.



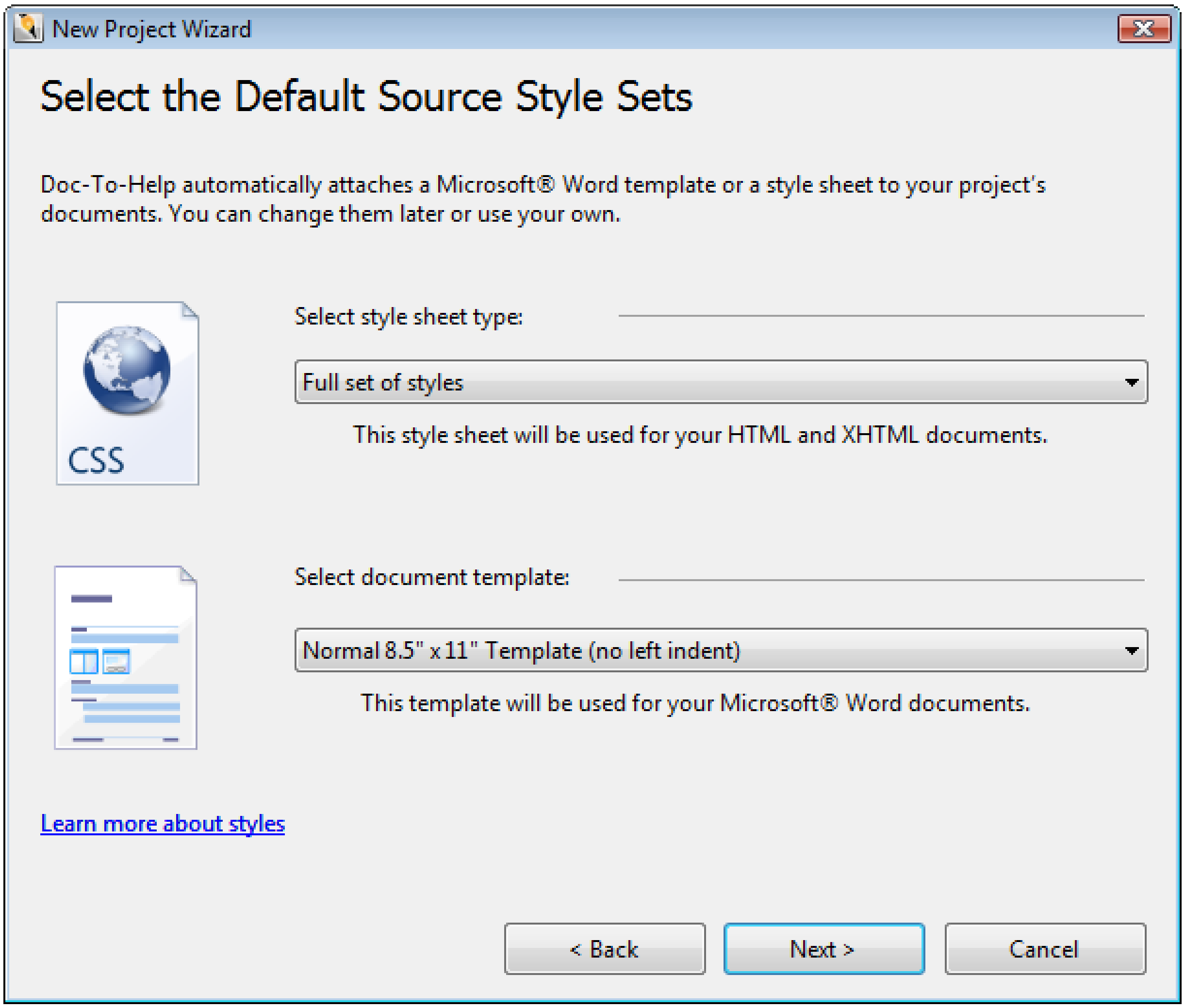
Click **Next**.

Doc-To-Help uses templates and style sheets to determine how content will look and behave in both your Source documents (when authoring in Word, Doc-To-Help's built-in XHTML editor, or an HTML editor) **and** your final Targets — the online Help and printed manual outputs.

Now you can use the wizard to select the style sheet (.css) for your HTML or XHTML Source documents and the template (.dot) file for your Microsoft® Word Source documents.

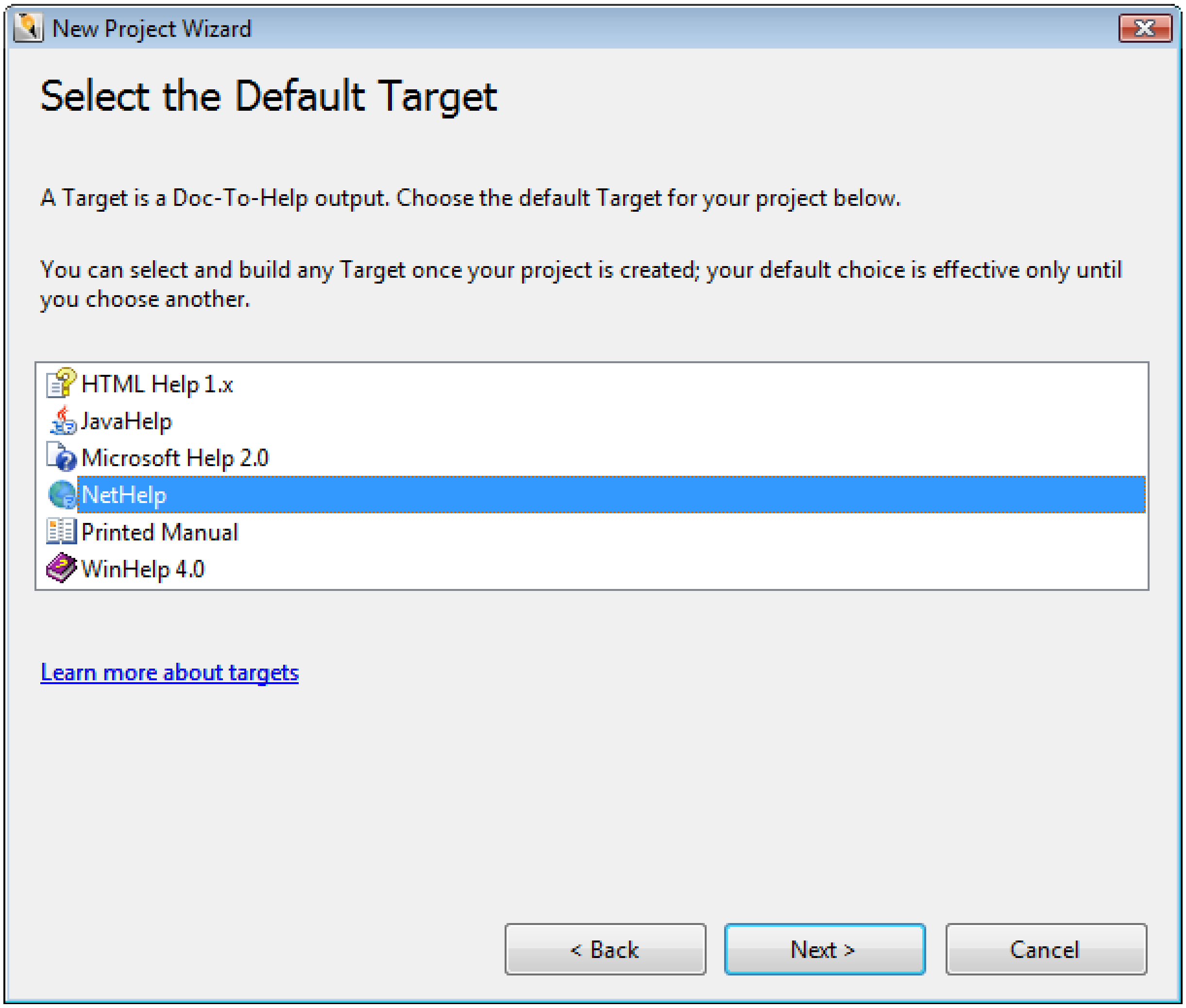
Doc-To-Help will automatically choose your Target style sheet and template based on your selections. For more information see the Guide to Templates and Styles on page 5.

- Leave the default setting of **Full set of styles**.
- Leave the default setting of **Normal 8-1/2" x 11" Template (no left indent)**.



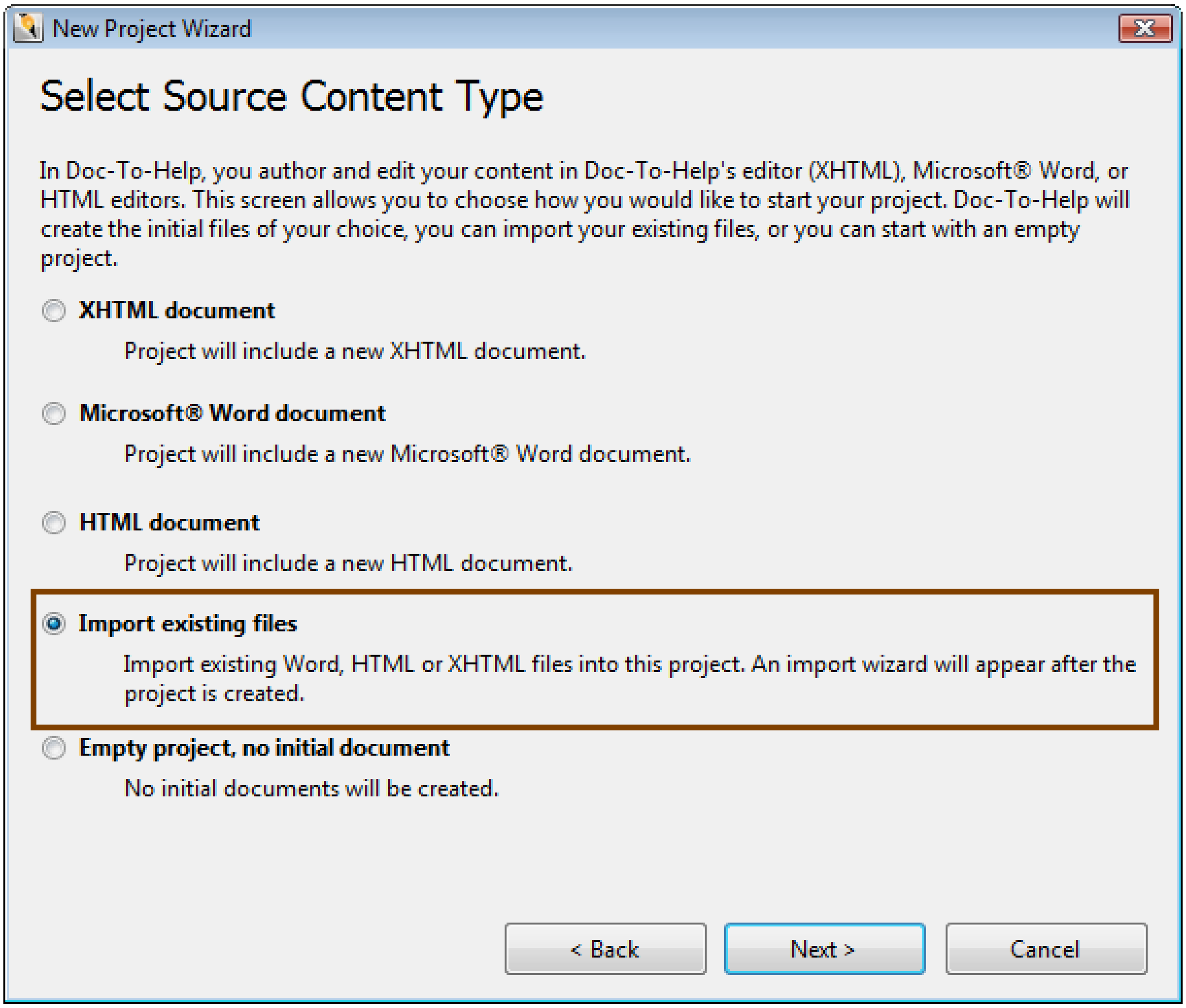
Click **Next**. Now, select your default Help target. You can generate any of these targets from your project at any time, but the default chosen here will be the one displayed when you open your project. Choose **NetHelp** (Doc-To-Help's browser-based uncompiled HTML format). This output can be used on the web, within a software application, or on a file server.

See Doc-To-Help Outputs and Deliverables (on page 30) for an explanation of all of the Targets listed on this screen.

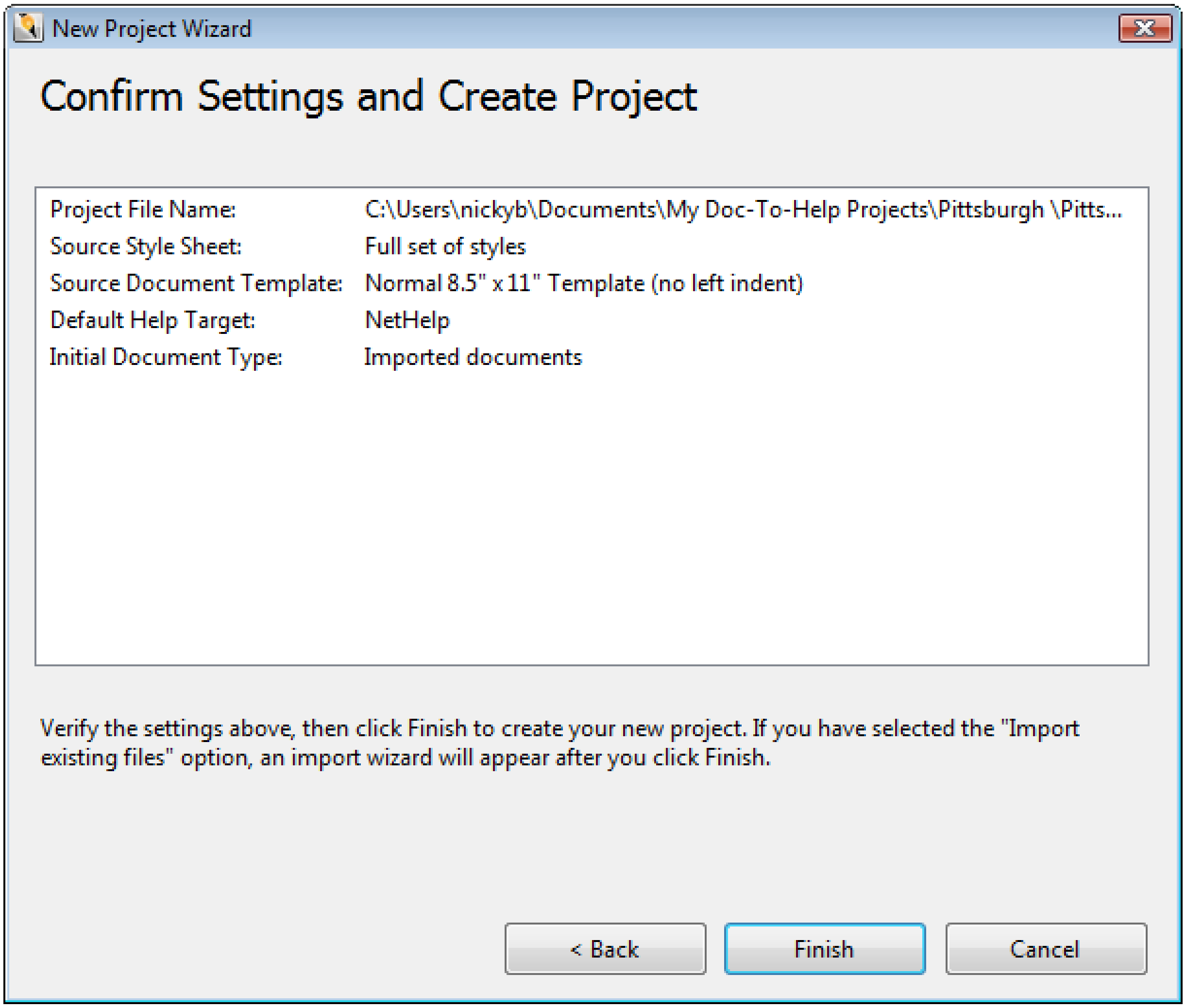


Click **Next**. Now you select the type of document you'd like your first document to be. Choose **Import existing files** — because we are going to use an existing document to start our project.

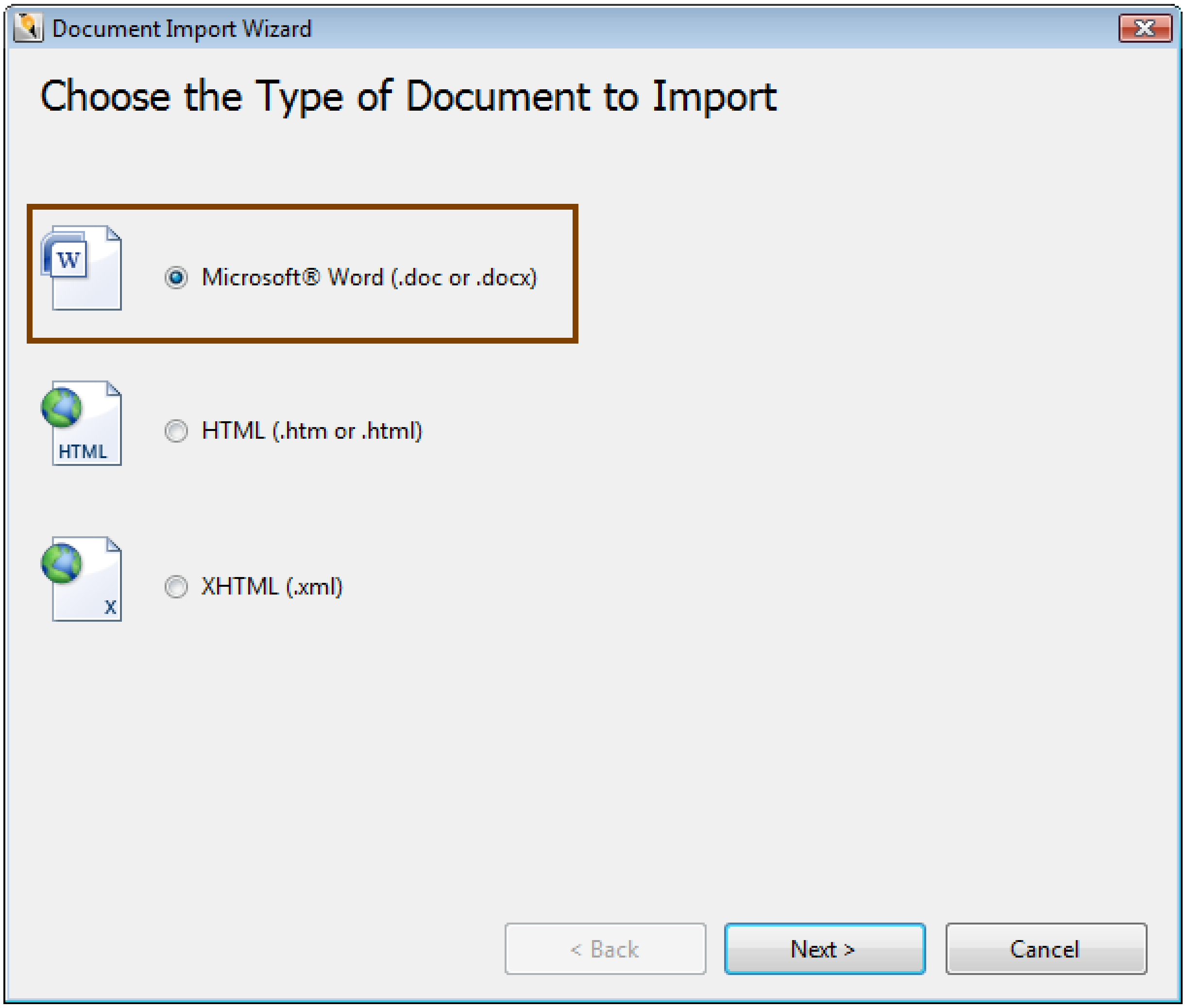
Doc-To-Help will launch another Wizard after the project is created that makes it easy to import an existing document.



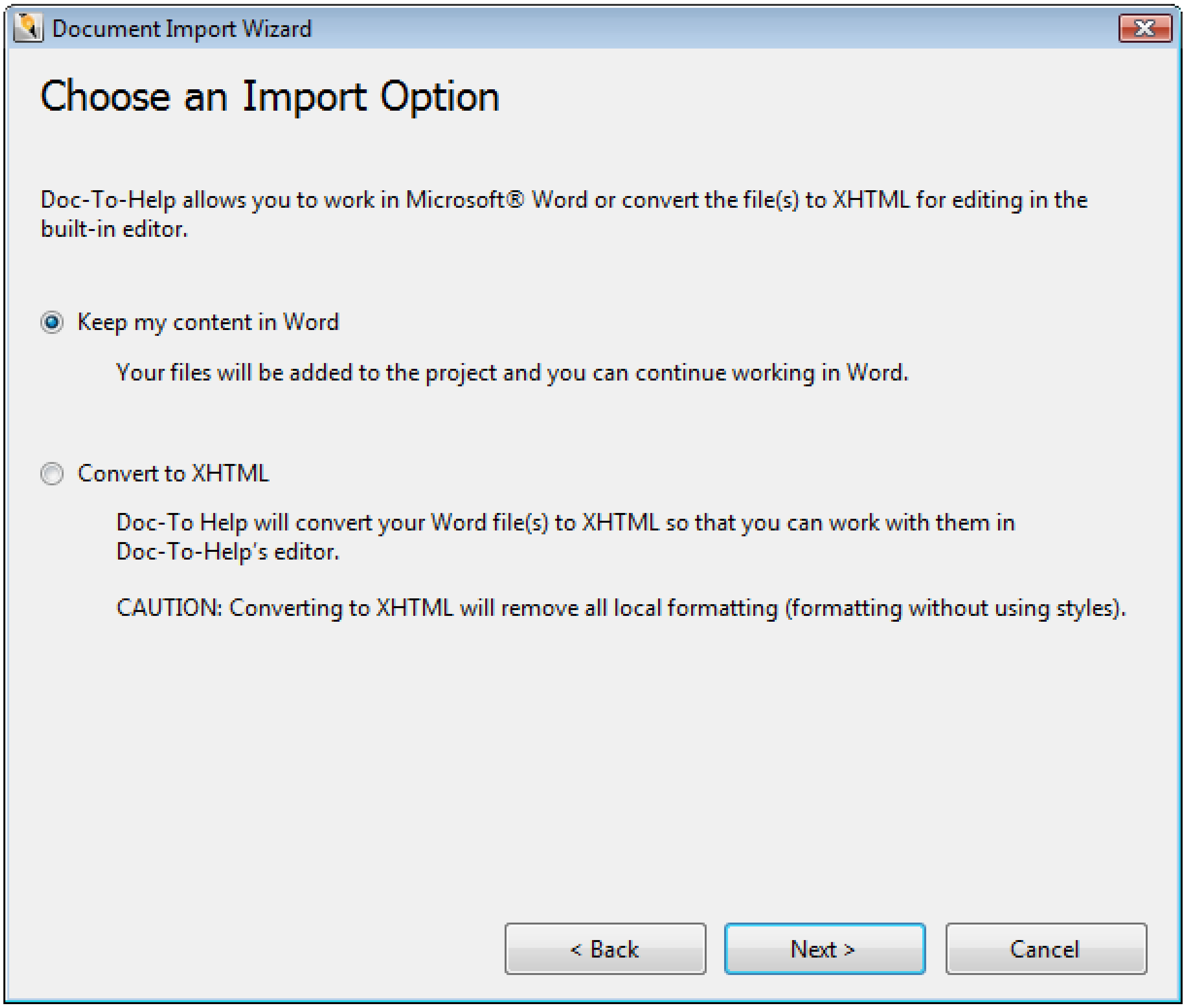
Click **Next**. That's it! Review the new project information and click **Finish**.



Our project is created, but since we chose the **Import existing files** option, the **Document Import Wizard** will open.



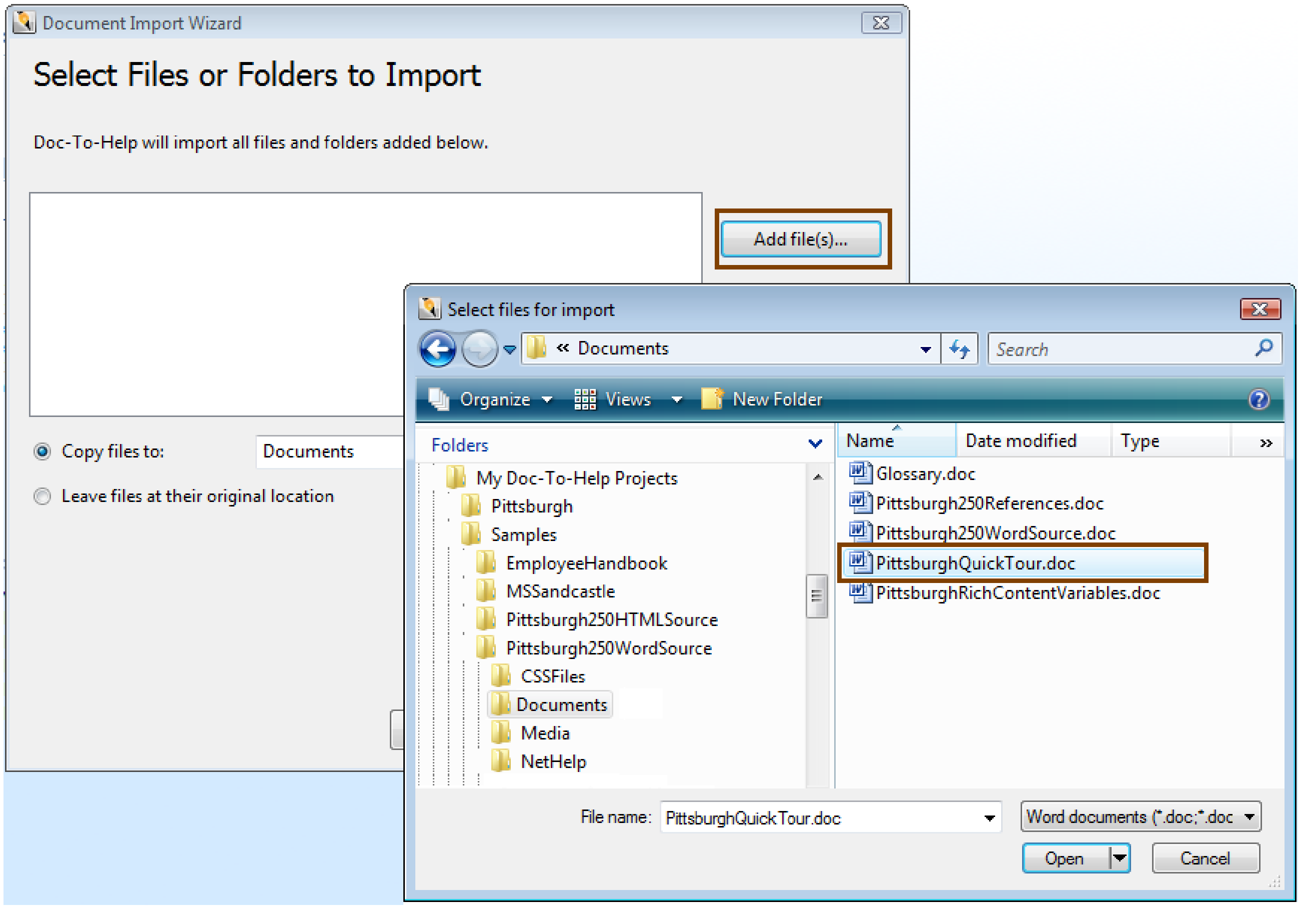
Choose **Microsoft® Word** and click **Next**.



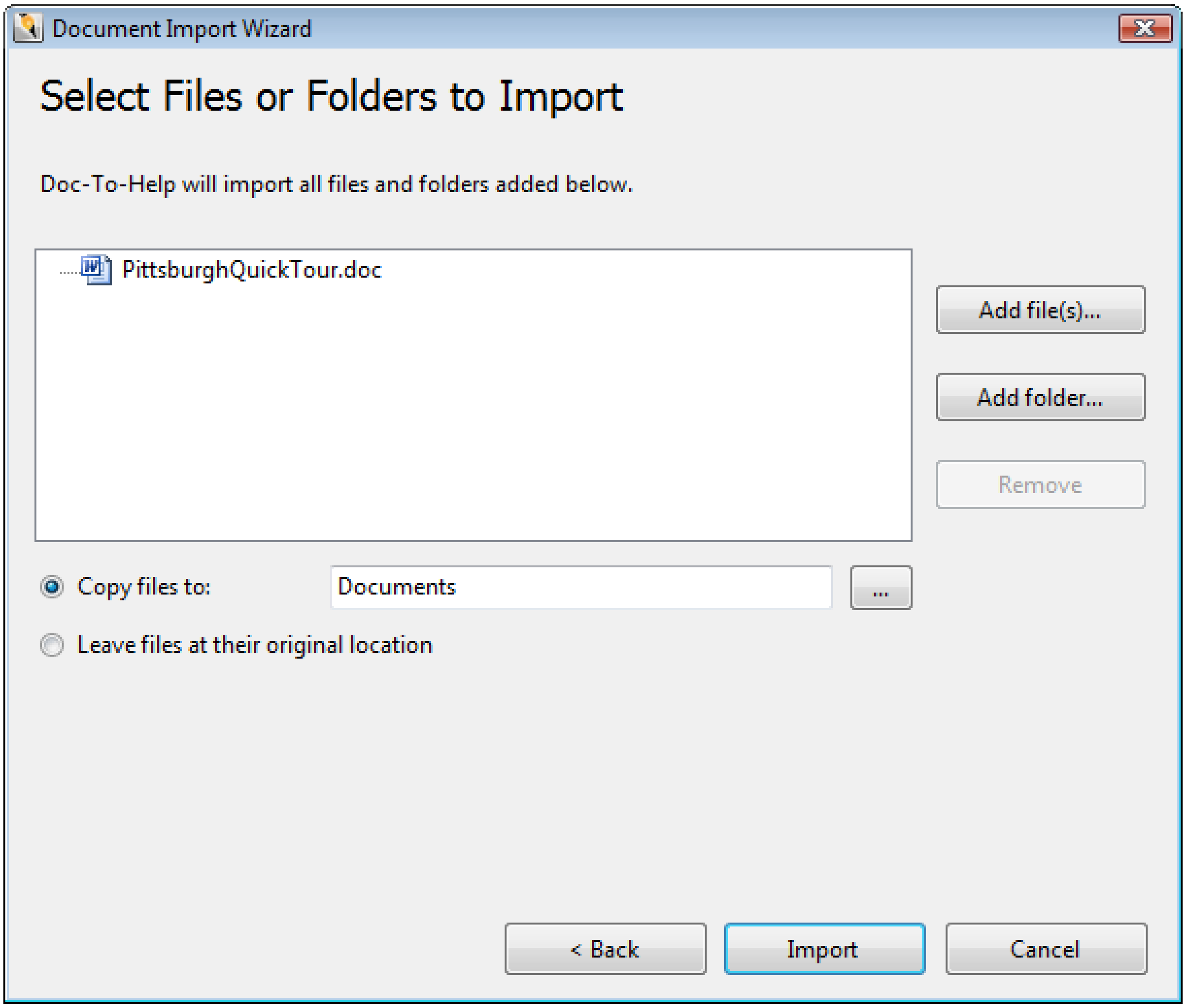
Choose **Keep my content in Word** and click **Next**.

Now click the **Add file(s)** button to select the file. The file you need is named **PittsburghQuickTour.doc**.

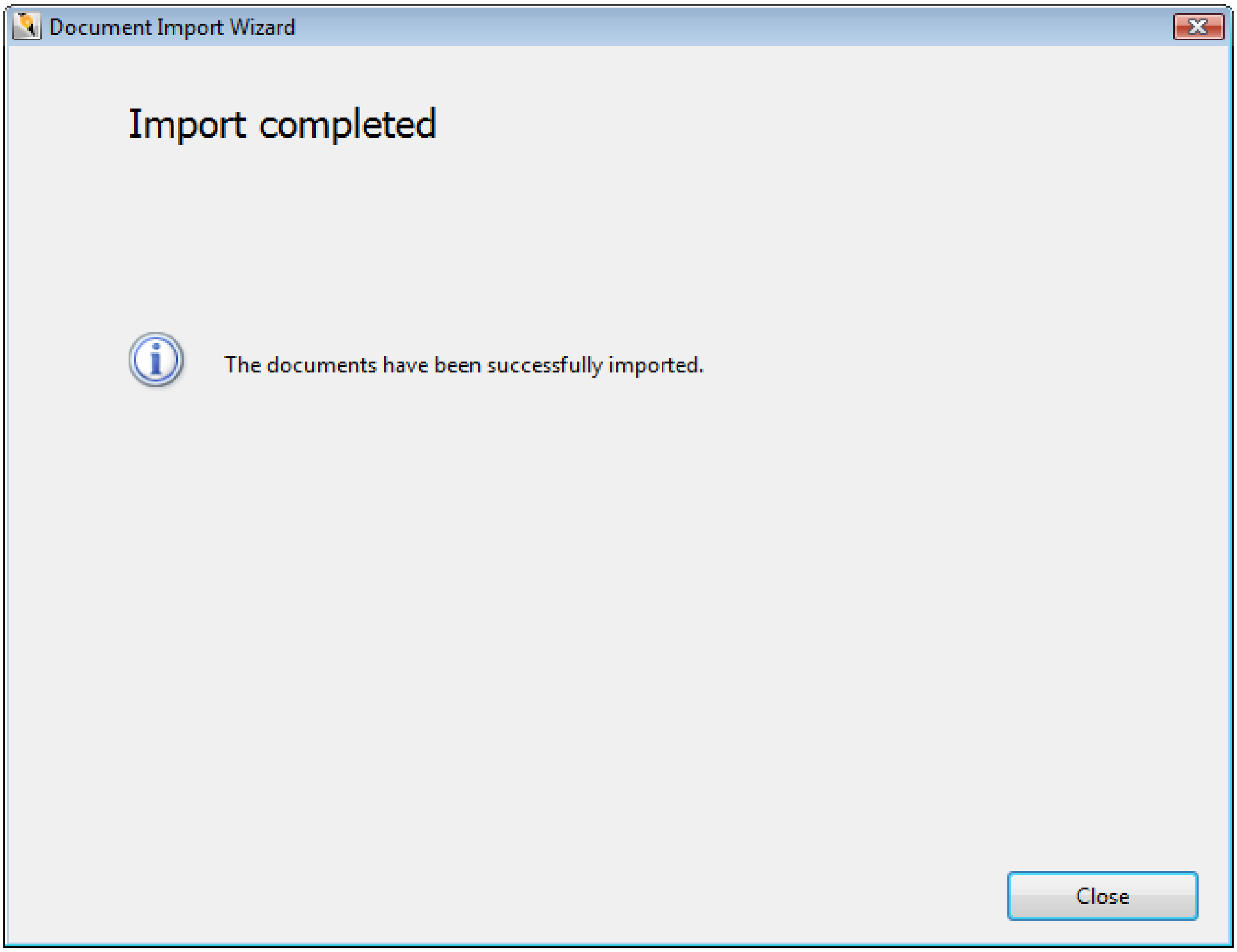
The **Select Files for Import** dialog box will automatically open to the **My Doc-To-Help Projects** folder. **PittsburghQuickTour.doc** can be found in **Samples\Pittsburgh250WordSource\Documents**.



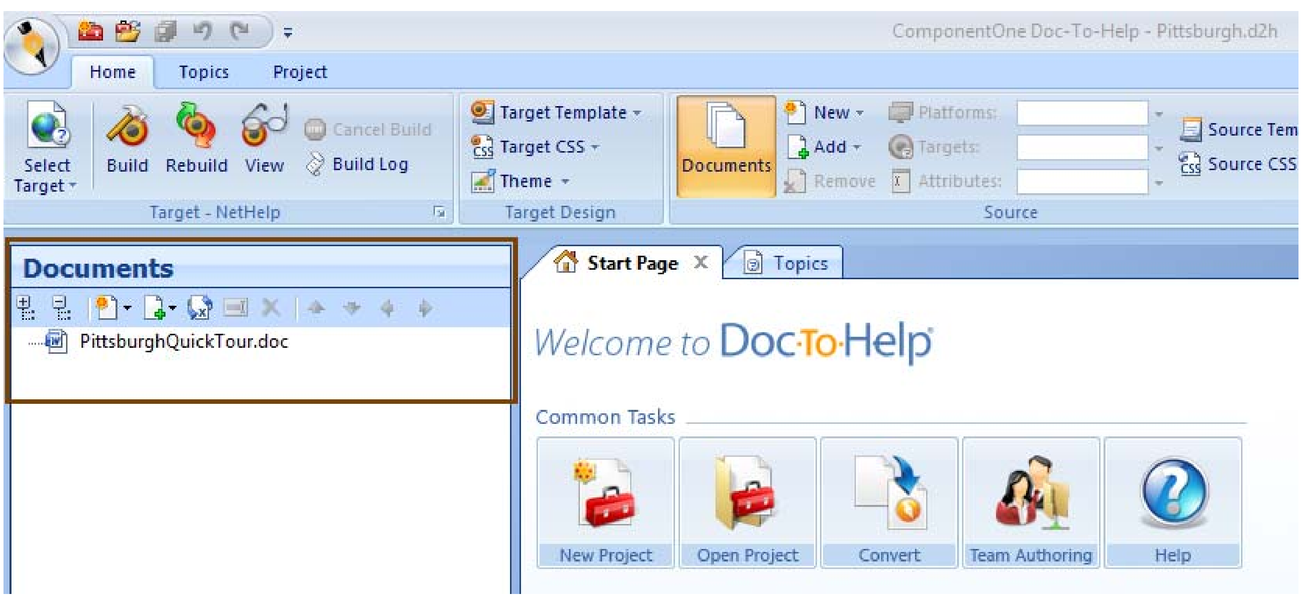
Click the **Open** button in the **Select files for import** dialog box. **PittsburghQuickTour.doc** will display in the Wizard. Please note that Doc-To-Help will automatically copy this file into the appropriate folder; your project's **Documents** folder.



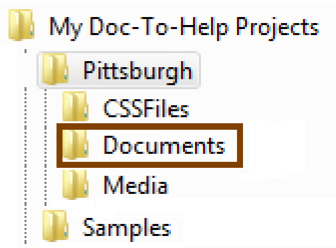
Click **Import** and the document will be imported.



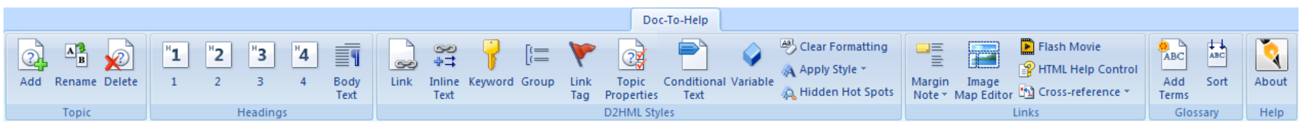
Click **Close**. The imported document displays in the **Documents** pane. The **Documents** pane is "Home base" in Doc-To-Help.



As noted earlier, our document was automatically added to the appropriate folder in the Doc-To-Help **Pittsburgh** project folder. Word source documents are stored in the **Documents** folder of the project.



From the **Documents** pane, double-click on **PittsburghQuickTour.doc** to open it. In Microsoft® Word, take a look at the **Doc-To-Help** ribbon (it will be a toolbar in pre-2007 versions of Word). With it, you can create links, add index entries, and more. (See Using D2HML on page 210 for an overview.) Doc-To-Help uses this information to create the final Targets.



Now it's time to learn more about specific Doc-To-Help features. Although the lessons below are divided up to make them more manageable, it is recommended that you do them in order so that you can see the progression of your project.

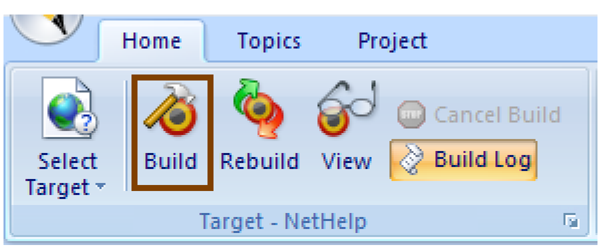
Building and Examining a Target

So far, we've done very little, but we've given Doc-To-Help enough to work with. If we build right now, we can take a look at what Doc-To-Help will do out-of-the-box — but first, take a quick look at our Word document. If it is not already open, double-click on **PittsburghQuickTour.doc** to open it.

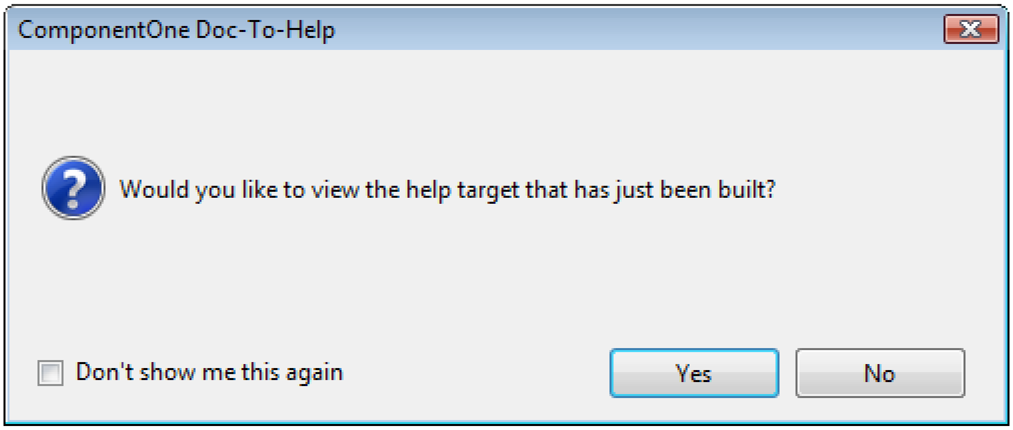
It is structured like this:
Heading 1 — Pittsburgh's 250th Anniversary
Heading 1 — Pittsburgh Sports
Heading 2 — Football
Heading 2 — Baseball
Heading 2 — Hockey
Heading 1 — Pittsburgh Sites

Doc-To-Help will use this structure to create topics, the table of contents, and automatic links.

Close **PittsburghQuickTour.doc** and click the **Build** button in Doc-To-Help. The default **NetHelp Target** (chosen earlier) will be built.

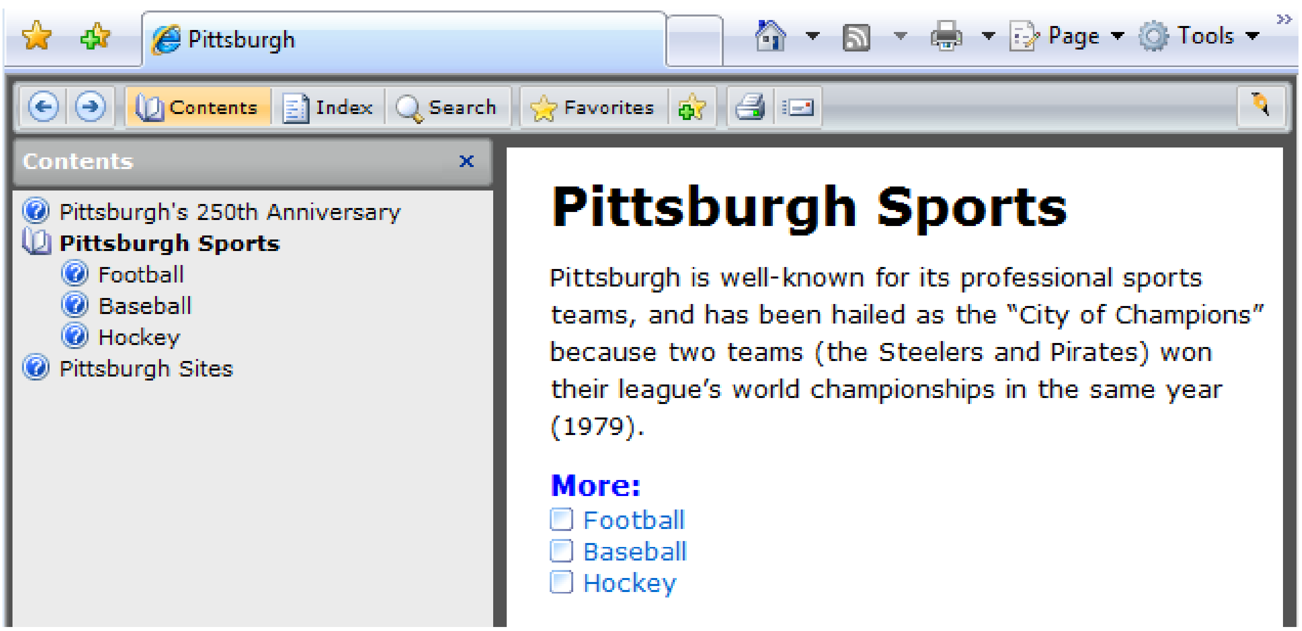


Doc-To-Help will prompt you to view the Target. Click **Yes**.



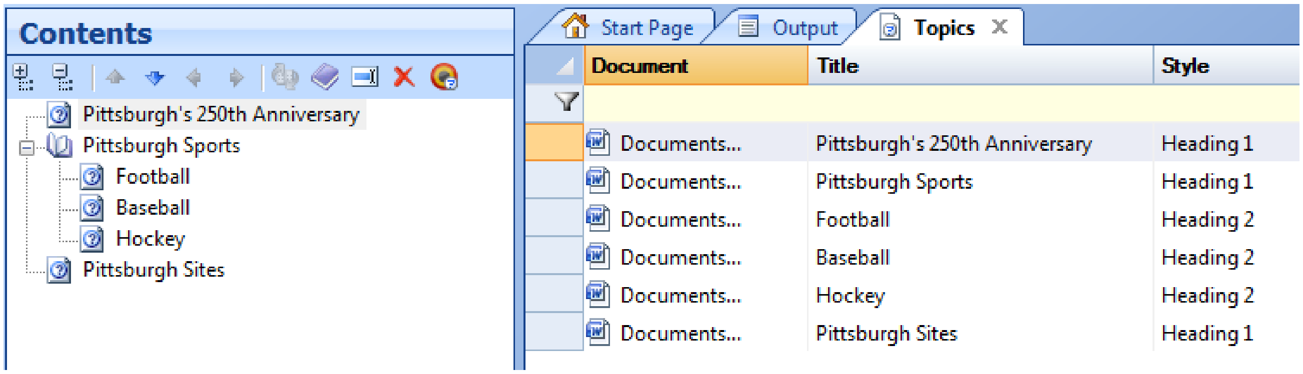
Take a look at the Target first:

- The Table of Contents structure is based on the Heading styles (since "Football", "Baseball", and "Hockey" were Heading 2s under the Heading 1 of "Pittsburgh Sports", they were converted to subtopics).
- Click on the "Pittsburgh Sports" topic in the Table of Contents. The Topic has three automatic subtopic links — "Football", "Baseball", and "Hockey" — which were created by Doc-To-Help based on the document structure. We'll learn later how to add custom subtopics, but these are created automatically.



Now a quick look at Doc-To-Help:

- Click the **Contents** pane button. The Table of Contents is displayed. In this pane, you can rearrange the TOC any way you want, and add/delete topics, but the automatically generated Table of Contents is logical, and can be used as-is if you like. Note that the Table of Contents in the **Contents** pane looks exactly like the Table of Contents in the NetHelp Target.
- Click the **Topics** window tab. Each Heading 1 and Heading 2 is a separate Topic in the project. This will make it possible to create links between topics and more.



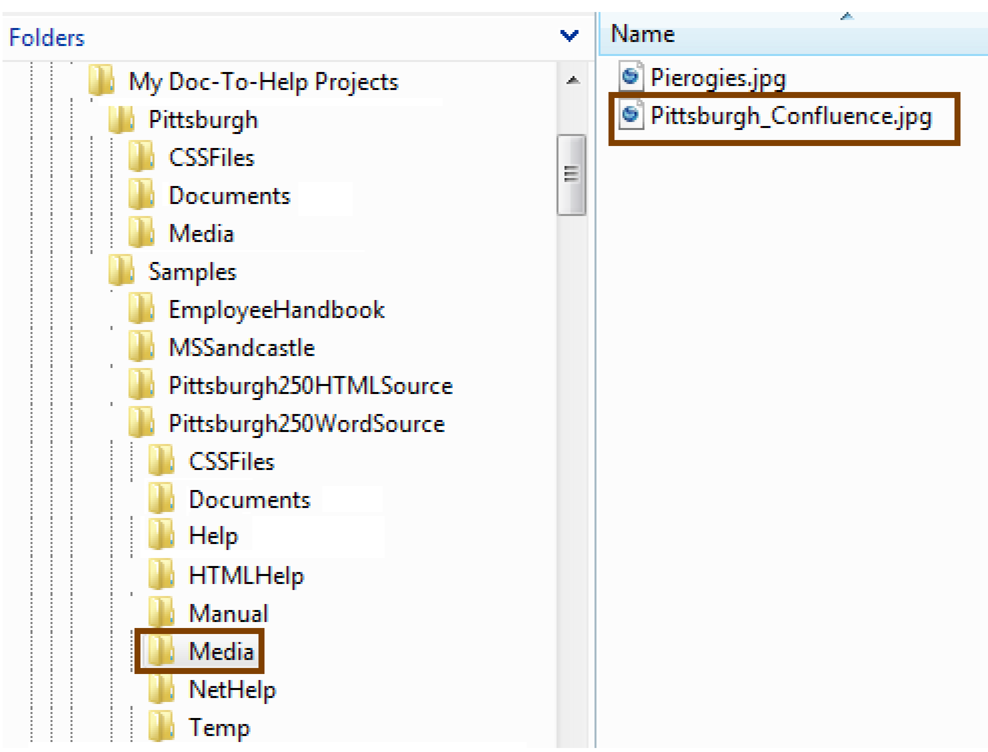
Now that we've seen our baseline, let's add more to this project.

Inserting a Graphic

First, the graphic we need must be moved into our project. Using Windows Explorer, navigate to \\My Doc-To-Help Projects\Samples\Pittsburgh250WordSource\Media. Copy **Pittsburgh_Confluence.jpg**. Drop it into the \\My Doc-To-Help Projects\Pittsburgh\Media folder. (Although you can use images from any folder, it is a good best practice to store all of your images in your project's Media folder.)

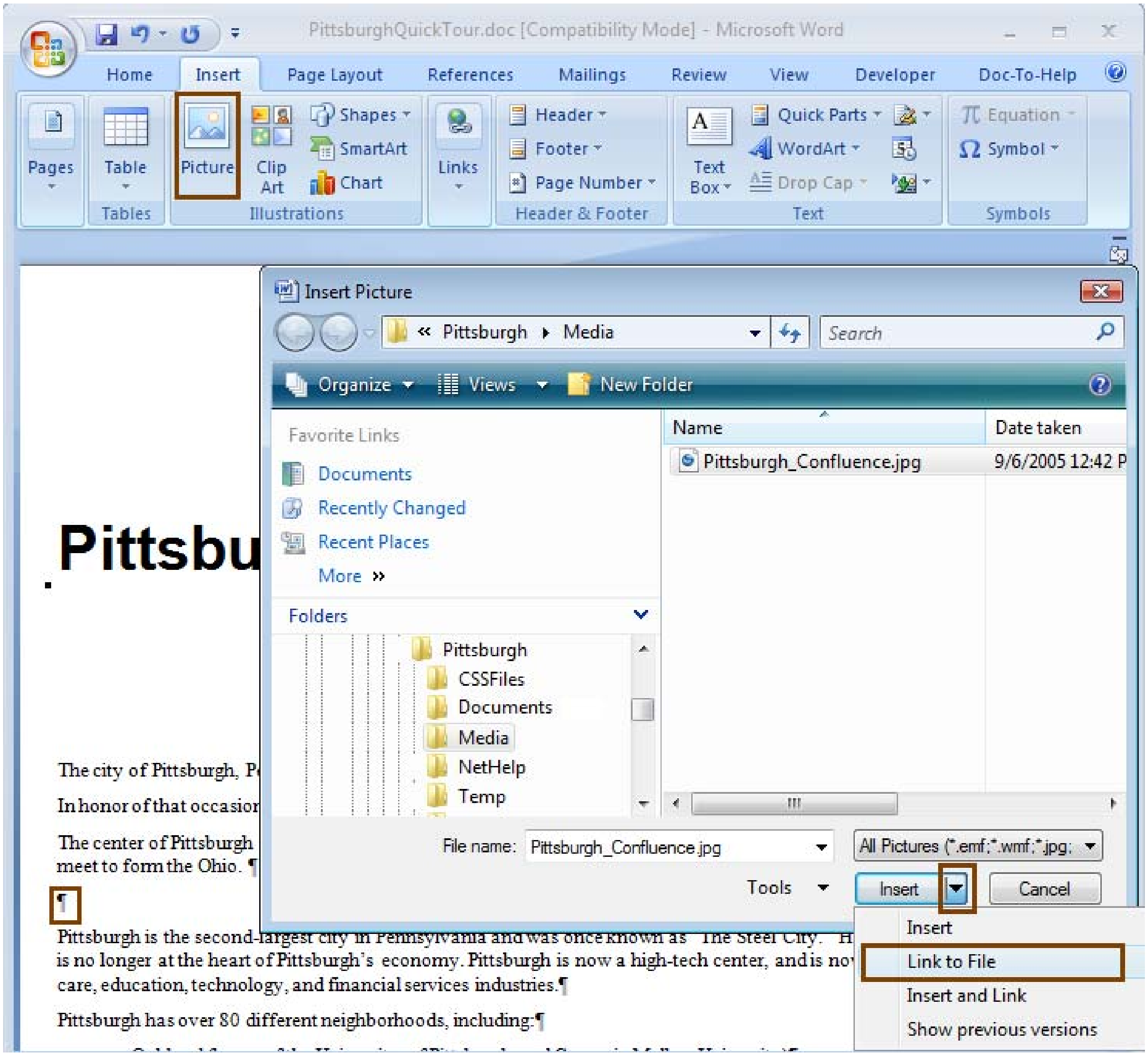
Just a reminder, the **My Doc-To-Help Projects** folder can be found here:

- **Windows® Vista:** \\Users\<username>\Documents\My Doc-To-Help Projects
- **Windows XP:** \\Documents and Settings\<username>\My Documents\My Doc-To-Help Projects

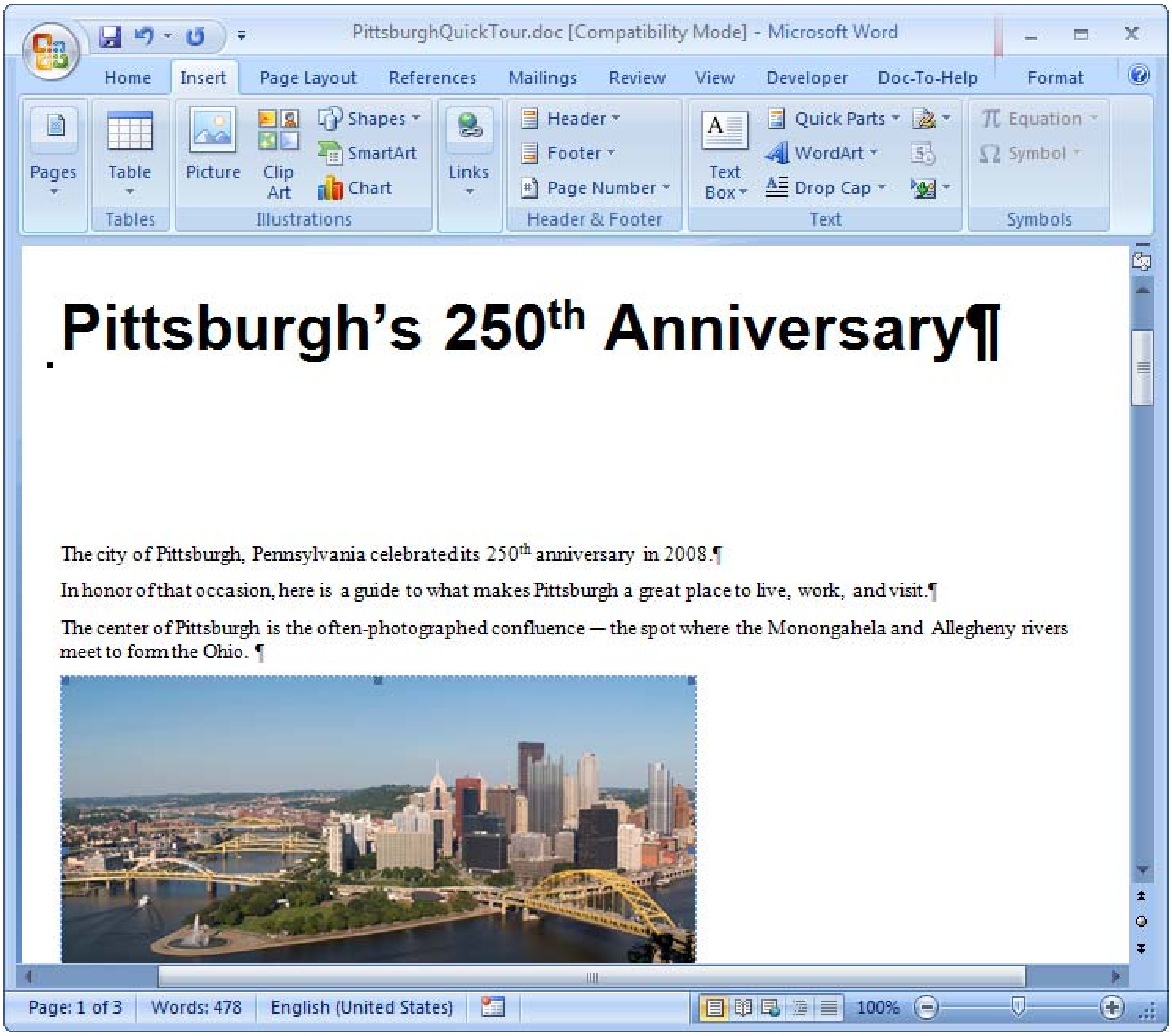


Now we are ready to insert our graphic. If it is not already open, double-click on **PittsburghQuickTour.doc** in the **Documents** pane to open it.

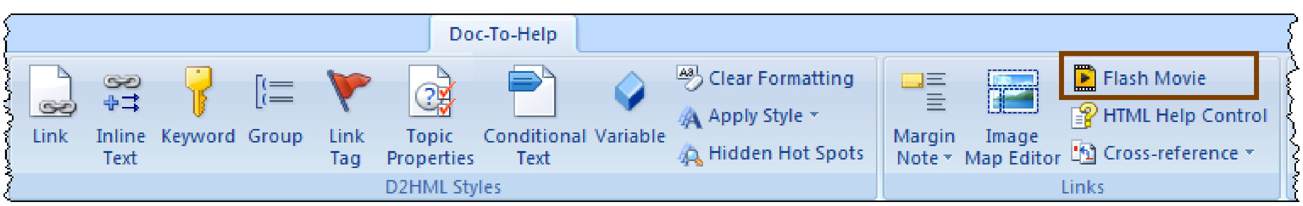
Add a new line after the third paragraph (it begins with "The Center of Pittsburgh…"). Click the **Picture** button in the **Insert** ribbon tab to open the **Insert Picture** dialog box. Choose **Pittsburgh_Confluence.jpg** and click the **Insert** drop-down. Choose **Link to File**. This is the recommended option for Doc-To-Help projects.



The graphic is now inserted in the document.



By the way, you can quickly add a Flash movie to your project using the **Flash Movie** button on the **Doc-To-Help** ribbon tab. See Inserting Flash Movies on page 133 for more information. Check out DemoWorks if you need a software tool to create movies.



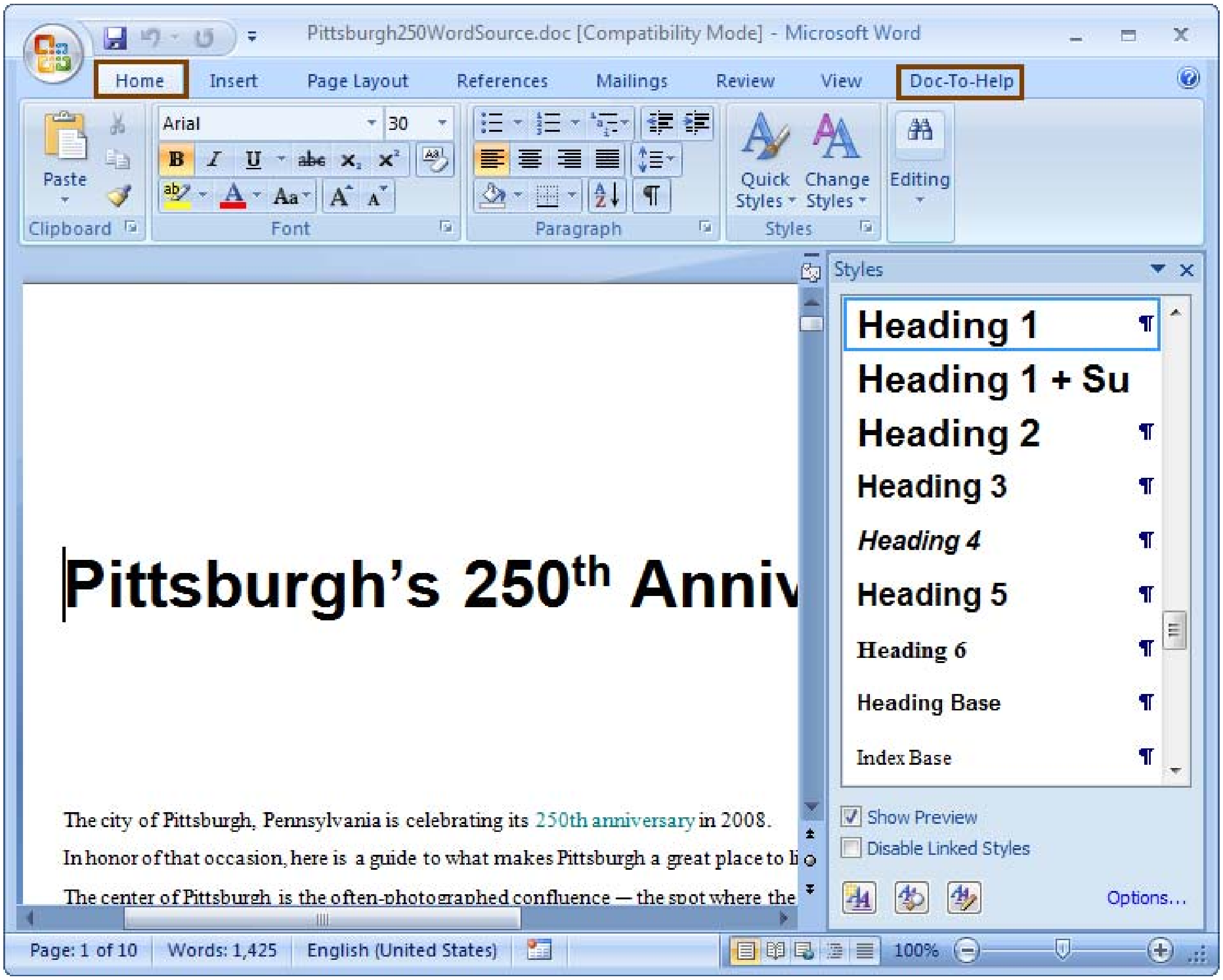
Applying a Style

Doc-To-Help uses styles to markup your source documents. In Word, these styles are in stored in a template (.dot or .dotm) file. The source styles are interpreted by the Target template to make the target files **look and behave** the way you desire. For example, as you saw earlier, Heading 1 and Heading 2 styles will determine your project's topic list, the structure of your Table of Contents and the automatic subtopic links.

In this section, you will learn how to apply a Doc-To-Help style in Word. Although the style we will apply in this exercise is one that only controls formatting, later we will apply styles that control both formatting and behavior (see Adding Glossary Entries on page 87).

If you were using Doc-To-Help's XHTML editor, you would apply styles using the **Style List** window, which is also very easy to use.

When using Microsoft Word as your editor, you can apply styles quickly using the **Home** and **Doc-To-Help** ribbon tabs.



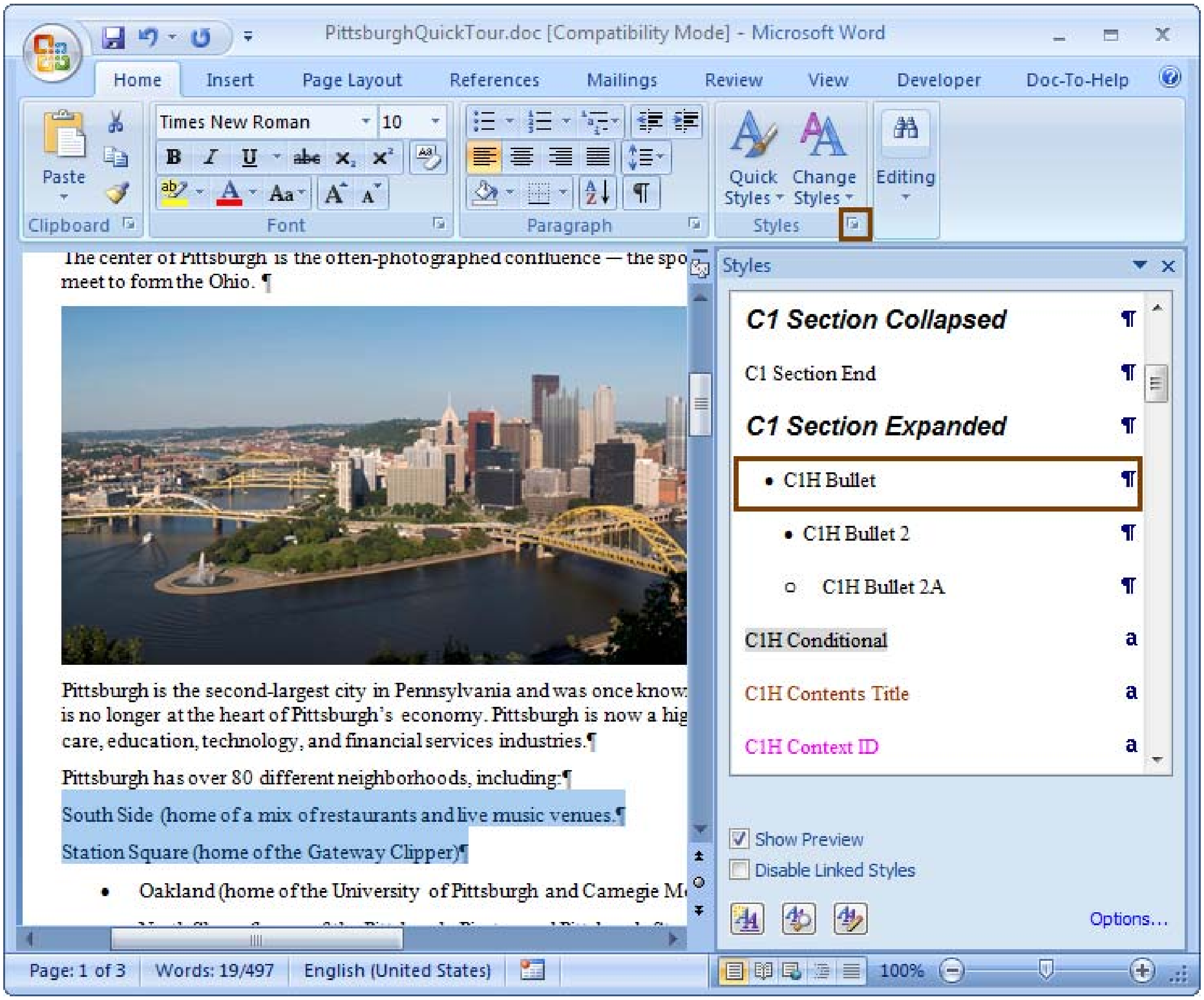
Let's add some new text to **PittsburghQuickTour.doc**. After “Pittsburgh has over 80 different neighborhoods,” add:

South Side (home of a mix of restaurants and live music venues)

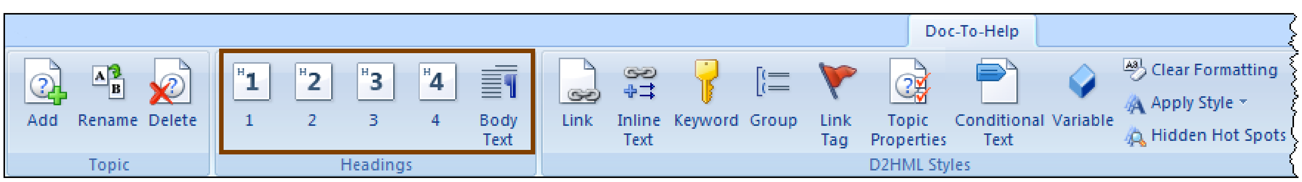
Station Square (home of the Gateway Clipper)

Select the text and click the **Styles** dialog box launcher on the **Home** ribbon tab. The **Styles** box will open. (In Word 2003, from the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** box will open.)

Choose the **C1H Bullet** style. The style is applied.

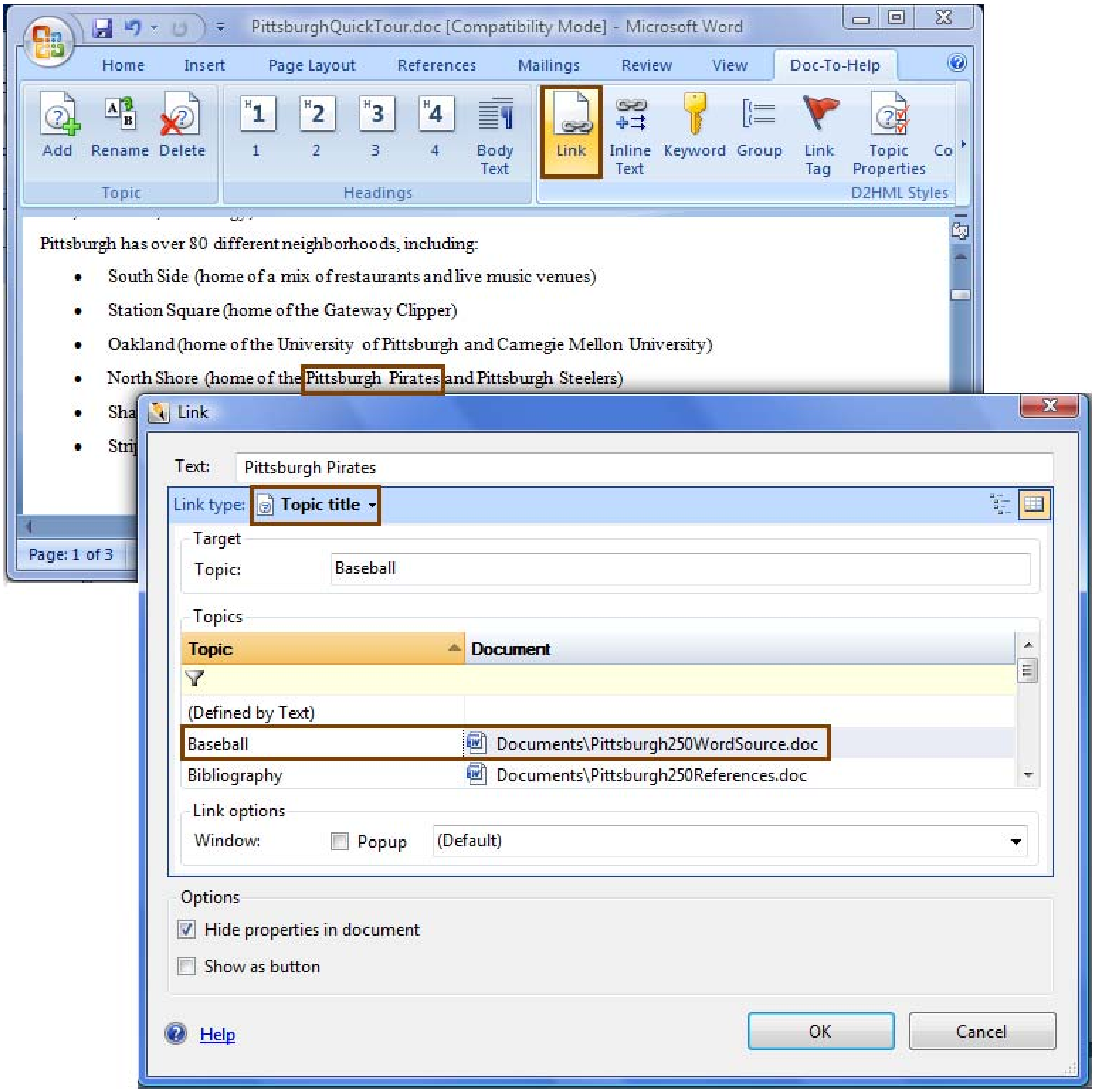


Open the **Doc-To-Help** ribbon tab, you'll see that it includes the **Heading 1, 2, 3, 4** and **Body Text** buttons so you can apply those styles quickly.



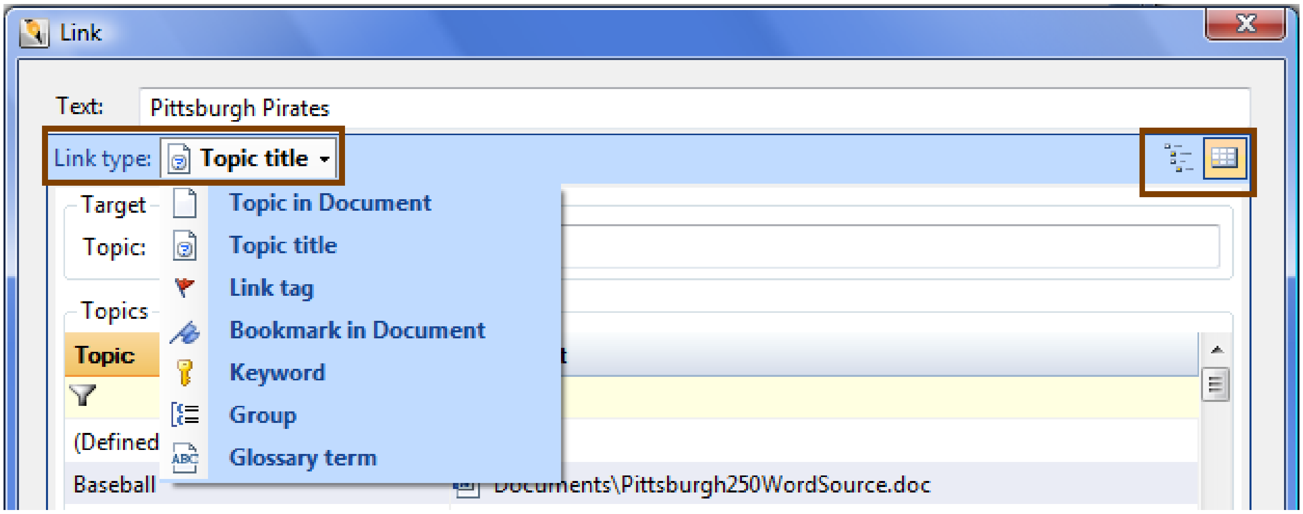
Creating a Topic Link

In **PittsburghQuickTour.doc**, select the phrase "Pittsburgh Pirates" in the text (it is in the “North Shore” bullet point) and click the **Link** button on the **Doc-To-Help** ribbon tab. The **Link** dialog box will open.



Take a quick look at the **Link** dialog box. Note that you can also create links to **Keywords** (AKA index entries), **Groups**, **Bookmarks**, **Link tags**, and **Glossary** entries. Simply click on the **Link type** drop-down. Also note that in the **Options** area, you can select the **Show as Button** check box. This will change your link text to a button. Selecting the **Popup** check box in the **Link options** will make the topic display in a popup window. By the way, linking will work in the PDF version of your manual target, as well as your Help targets.

Handy Hint: You can select a **Grid** or **Tree view** of the "link to" topics. Click the button on the right to choose.



Now that you have learned more about the **Link** dialog box — in the **Topics** area, choose **Baseball** from the list. When you click **OK**, you will create a topic link from the phrase "Pittsburgh Pirates" to the topic about baseball.

For more on links, see Creating Links on page 212.

If you have hidden text turned "on" in your document, you will see the information for this link tag displayed. As you can see, the "tag" is the Topic you selected ("baseball").

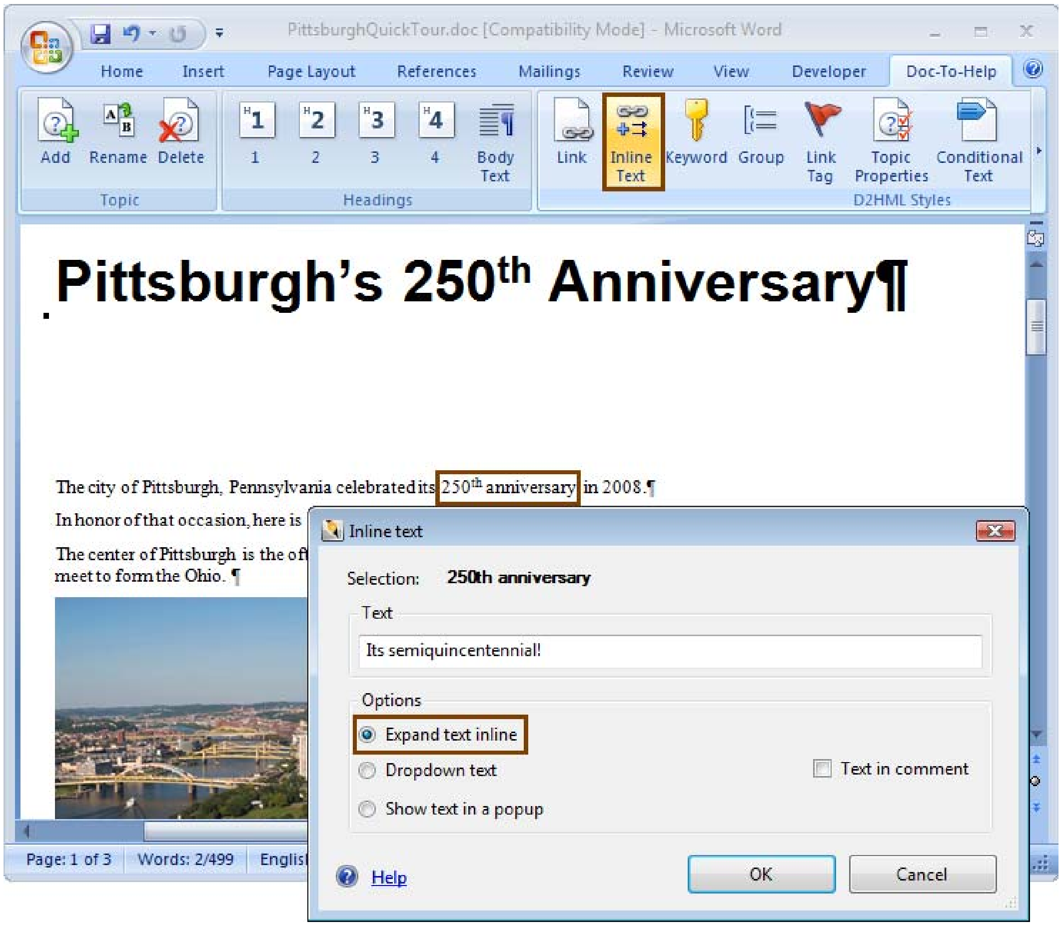
- Oakland (home of the University of Pittsburgh and Carnegie Mellon University)¶
- North Shore (home of the Pittsburgh Pirates|tag=Baseball;document=Documents\PittsburghQuickTour.doc and Pittsburgh Steelers)¶

If you would like to hide hidden text — in Word 2003 and earlier, choose **Tools** > **Options**. In the **View** tab, **Formatting Marks** section, clear the **Hidden Text** check box and click **OK**. In Word 2007, choose the **Office button** > **Word Options**, and click **Display** on the left. Clear the **Hidden Text** check box on the right.

Creating Inline Text

Inline text is expanding, drop-down, or popup text. It's a great way to add dynamic information to your help project.

In **PittsburghQuickTour.doc**, select the phrase "250th Anniversary" from the first sentence and click the **Inline Text** button. Choose the **Expand text inline** radio button, and add text (Its semiquincentennial!) in the text box.



Click **OK**. Doc-To-Help will inform you that this is an invisible style (this means the inline text we added will be invisible in this document (our Source document) — as well as in our Target until clicked). Since that is OK, click **No** to close the message box. When we build the Target, "250th Anniversary" will be a hyperlink that will display "Its semiquincentennial!" when clicked. Save and close the document, build, and take a look if you would like.

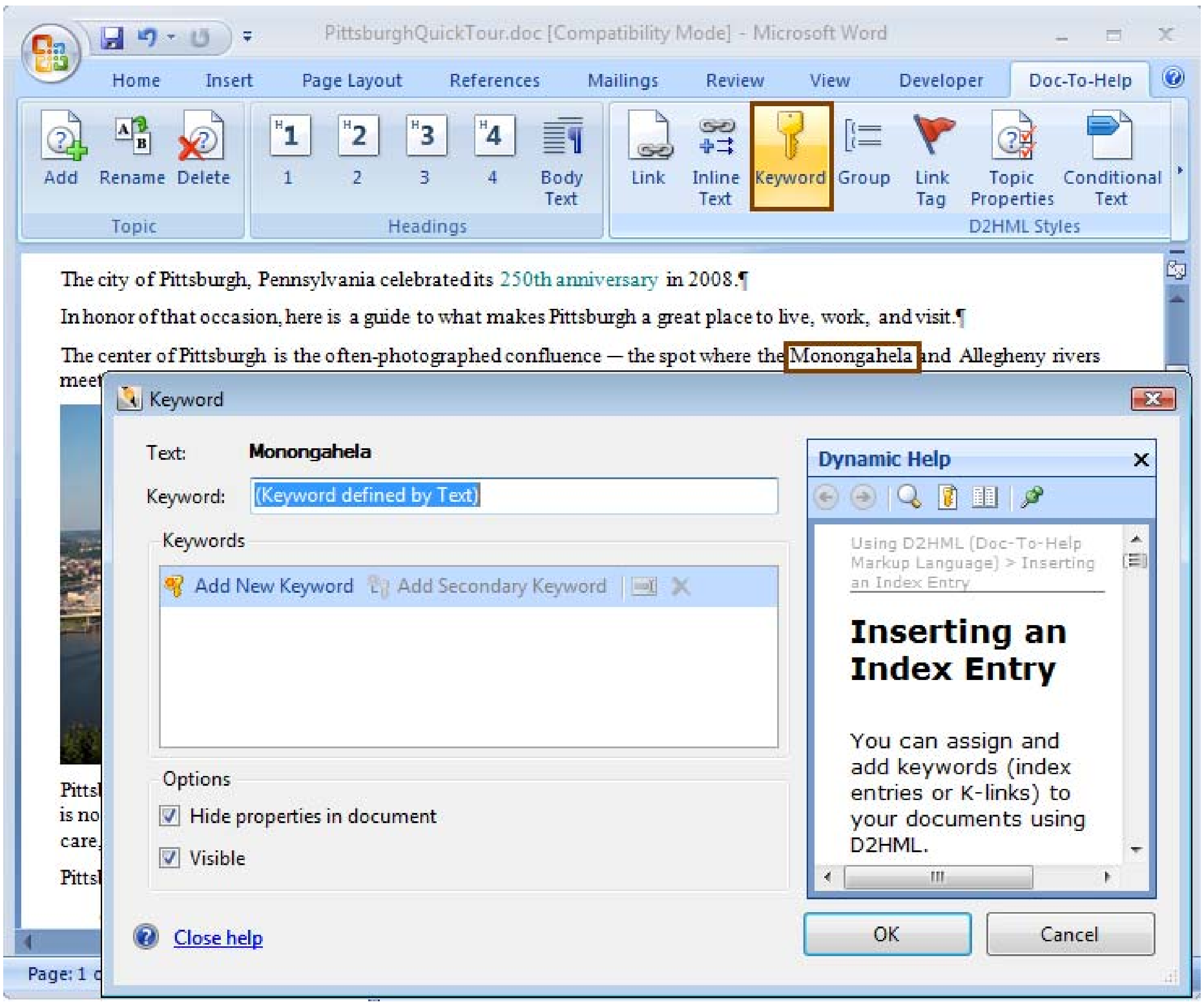
For more on Inline text (including working examples of each type) see Creating Inline, Dropdown, and Popup Text on page 215.

Creating an Index Entry

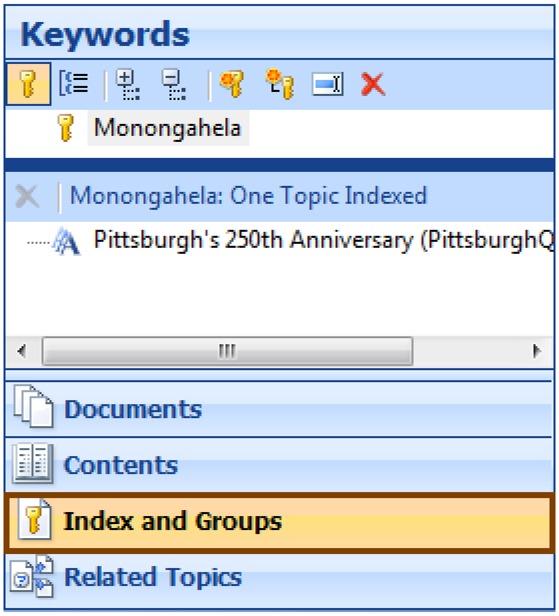
You can add index keywords (also known as K-Links) right from your document. You can also do this in Doc-To-Help, which will be demonstrated later.

Select the word "Monongahela" in **PittsburghQuickTour.doc** (it's in the 3rd paragraph) and click the **Keyword** button. Click **OK** to add “Monongahela” to your index.

“Monongahela” will display in your source document with a different text color, but that color change will not be visible in your final Targets. It is there in your source so that you can see it.



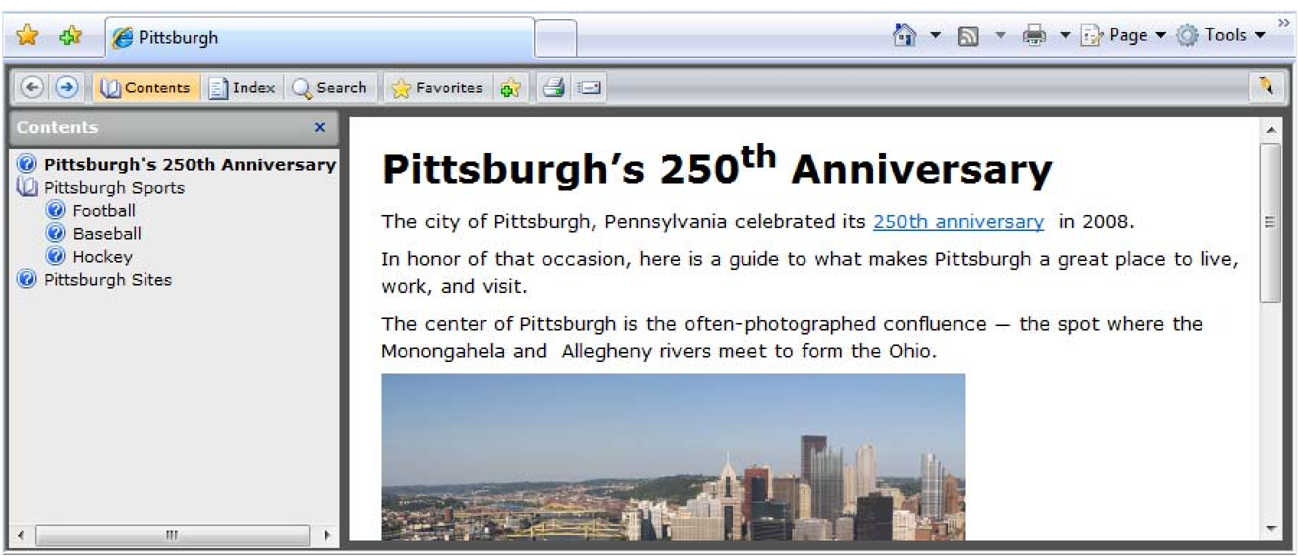
When you look in the **Index and Groups pane** of Doc-To-Help, Monongahela will be there.



You can add additional keyword and secondary keywords from this dialog box. For more on indexing, see Inserting an Index Entry on page 215.

Let's build our target now and take a look at it. First, save and close **PittsburghQuickTour.doc**. Click the **Build Target** button in Doc-To-Help. After the build is complete, view the Target. See Building a Target on page 235 for more information.

The project will open in your browser.



Click on "Pittsburgh's 250th Anniversary" in the table of contents. You'll see that:

- **"250th Anniversary"** is a hyperlink that will display text (Its semiquincentennial!) when clicked.
- **"Monongahela"** is not highlighted in the Target (but it is included in the Index — click the **Index** button above the Table of Contents).
- The first two bullet points (about the **South Side** and **Station Square**) are properly formatted.

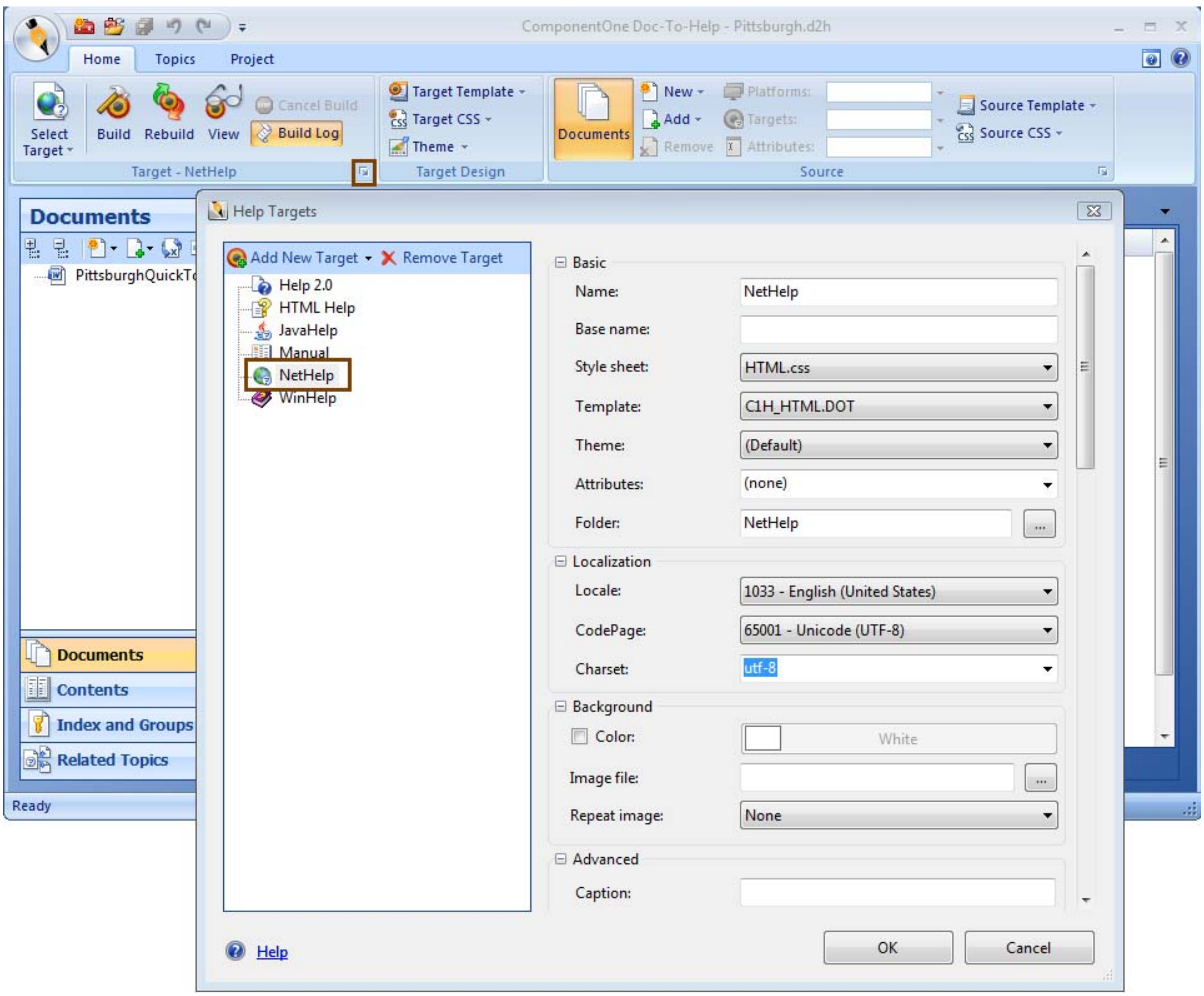
- **"Pittsburgh Pirates"** is a hyperlink that opens the "Baseball" topic.

You will also notice that the project name appears in the browser as **"Pittsburgh"** — the name we gave the .D2H project in the wizard. We will change that in the next step using the **Help Targets** dialog box.

Defining Help Targets

By default, our sample has six default Help Targets. Using the **Help Targets** dialog box in Doc-To-Help, we can rename our targets and delete the ones we don't need.

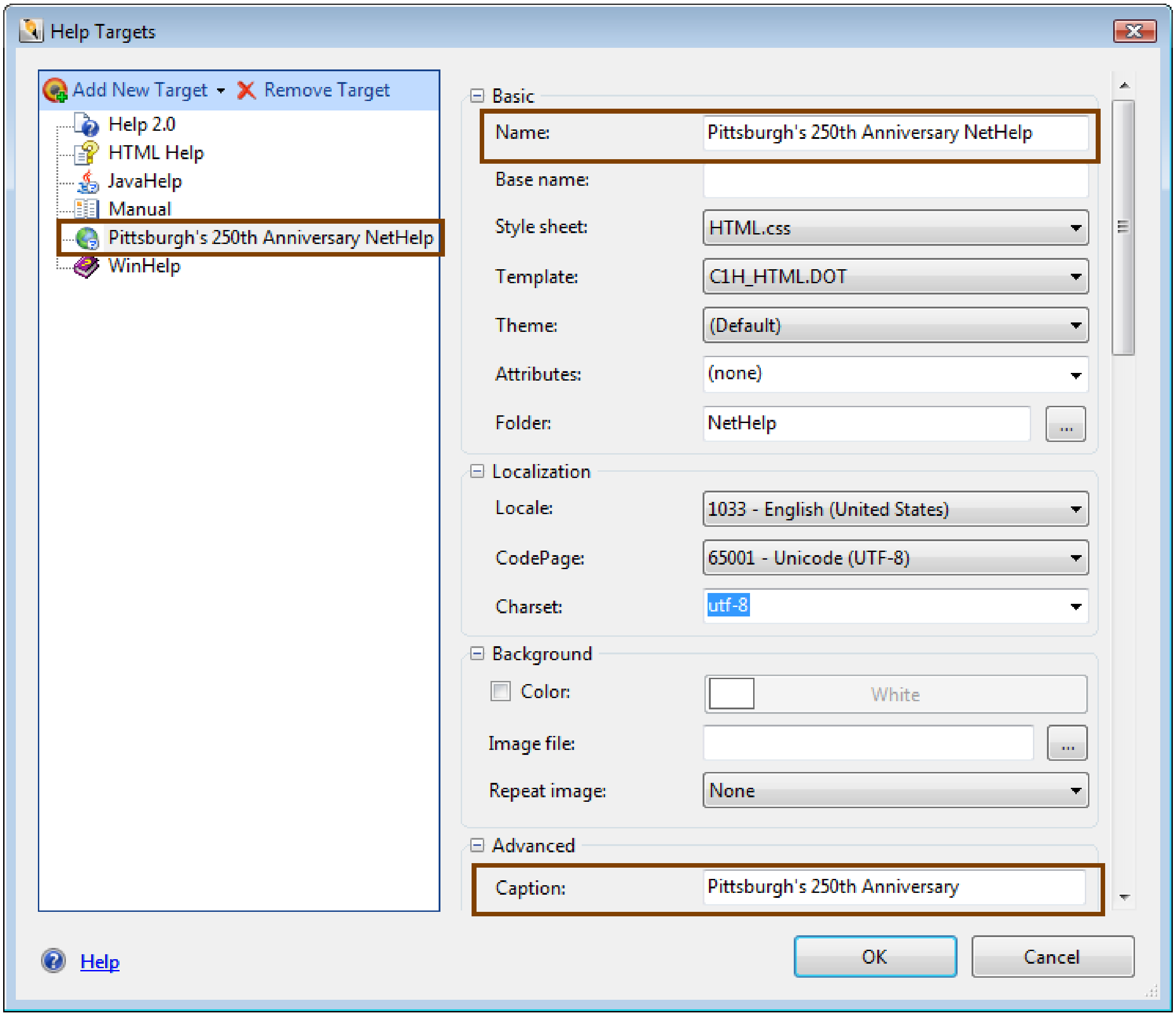
First, click the **Targets ribbon group** dialog box launcher to open the **Help Targets** dialog box.



Since NetHelp is our default Target, that is the option displayed first. Let's change the **Name** field from "NetHelp" to "Pittsburgh's 250th Anniversary NetHelp."

As soon as you click in the window on the left, the name of the project will change in that list.

Also enter "Pittsburgh's 250th Anniversary" in the **Caption** field. This will change the name displayed in the browser from "Pittsburgh" to the name we prefer.



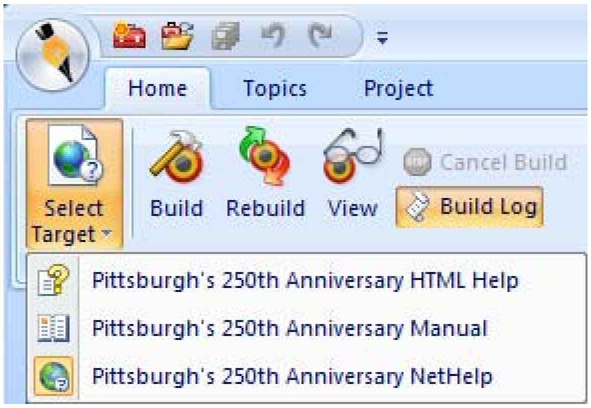
Click on the **HTML Help** (compiled HTML Help) **Target** on the left. Rename it also.

Do the same with the **Manual Target**. While you are there, scroll down and change the "Title" to "Pittsburgh @ 250," the "SuperTitle" to "Visitors Guide," and the "By Line" to "Pittsburgh Productions."

Since we don't need to create Help 2.0, JavaHelp, or WinHelp for this project, click on those **Targets** in the window on the left and click the **Remove Target** button for each. (You can always add them back later.) Click **OK**.

See Creating Help Targets on page 155 for more information.

Now, click the **Select Target** button. The newly named Targets are displayed, instead of the defaults.



If you'd like, save and close your documents. Select each Target, then build and view the Targets. You will see the changes we made displayed in each Target.

Please note: you can have several version of the same Target. For example, you could have multiple NetHelp Targets directed at different audiences. Just add them using the **Add New Targets** button in the **Help Targets** dialog box.

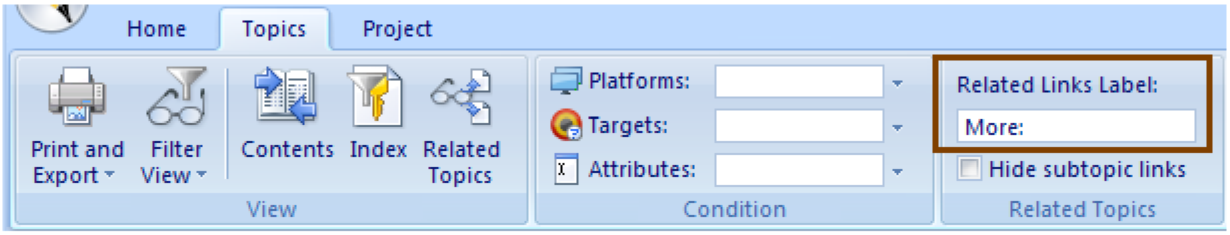
Creating Custom Topic Relations

As we discussed earlier, when you open your NetHelp Target and click on the "Pittsburgh Sports" topic in the Table of Contents, you will notice that at the bottom of the topic, there is a section that begins with "**More:**," followed by links to "Football," "Baseball," and "Hockey." These are called **Subtopic Links** and they are created automatically by Doc-To-Help, based on the structure of your document.

"Pittsburgh Sports" is a Heading 1, and "Football," "Baseball," and "Hockey" are Heading 2's immediately following it in **PittsburghQuickTour.doc**. If there were Heading 3's under "Football" they would appear as **Subtopic Links** in that topic.

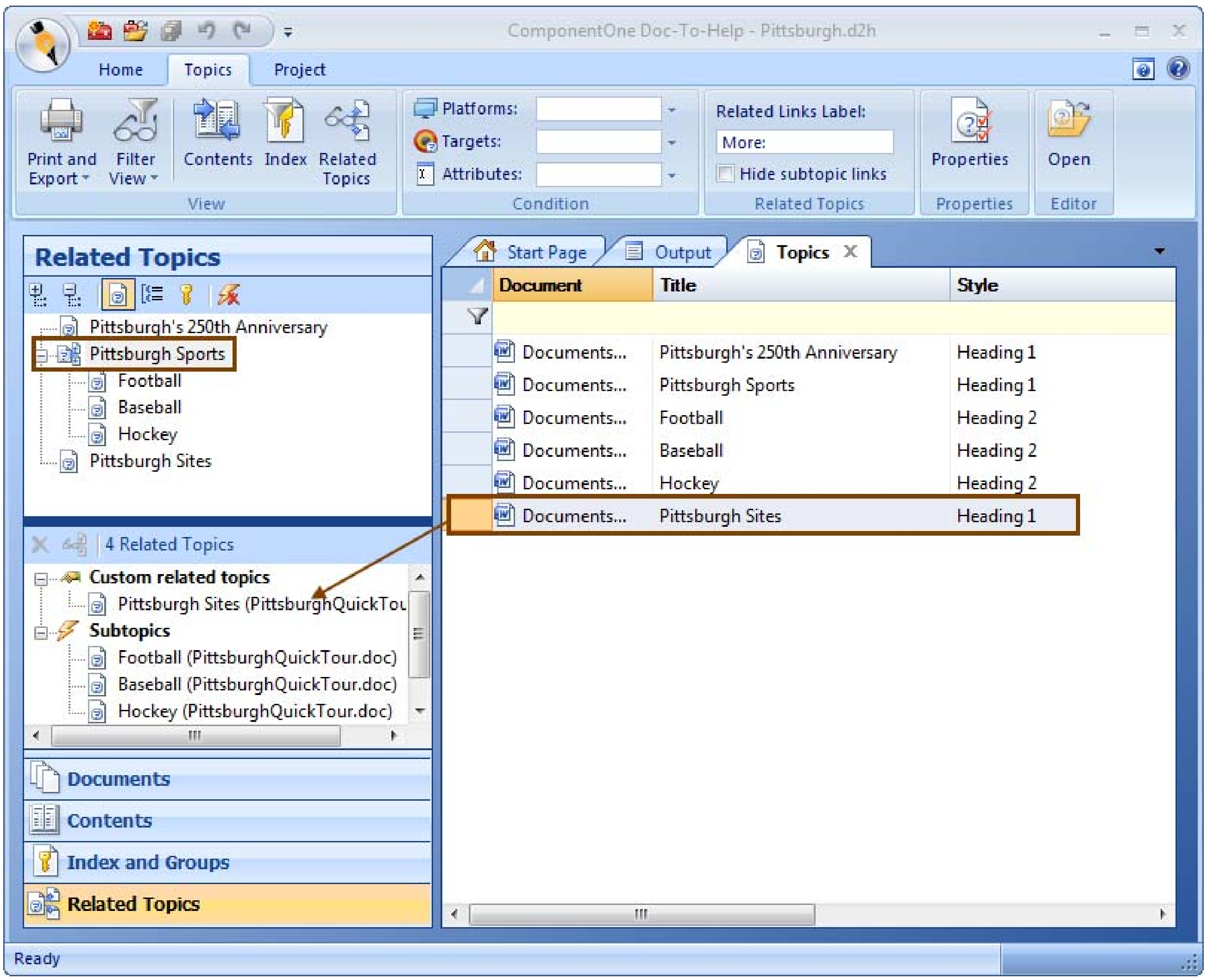
By default, the word "More" appears before the Subtopic links, but you can change that. Open the **Help Targets** dialog box for your Help Target and change it in the **Label** field.

You can change it for individual topics by selecting a **Topic** in the **Topics window** and changing the **Related Links Label** in the **Topics** ribbon.

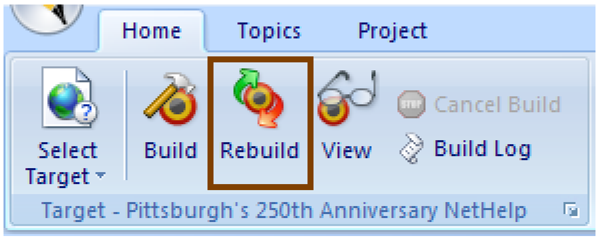


It is easy to add your own topic relations using the **Related Topics** pane.

Open the **Related Topics** pane and choose **Pittsburgh Sports** from the list. You'll see that "Football" "Baseball" and "Hockey" are displayed in the pane below automatically. Select **Pittsburgh Sites** from the **Topics window** and drag it into the lower **Related Topics** pane. It is now related (it is marked as a **Custom related topic**).

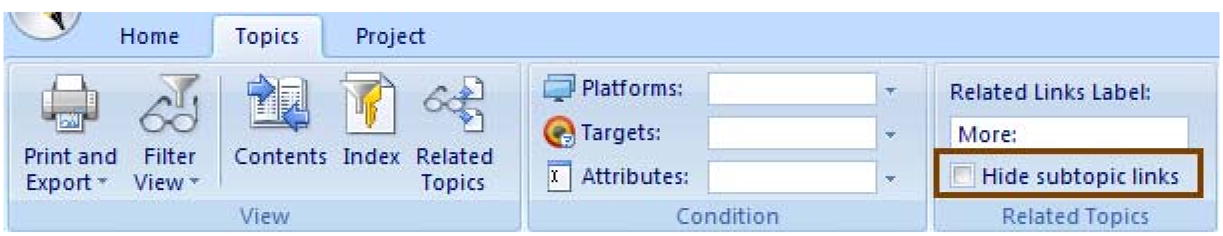


Save and close your documents, then click the **Rebuild Target** button and build your project.



Look at the new list of subtopic links for **Pittsburgh Sports**. "Pittsburgh Sites" has been added to the list. See Related Topics pane on page 111 for more information about topic relations.

If you would prefer the subtopic links are not displayed in a specific topic, choose that topic in the **Topics** window and select the **Hide Subtopic Links** check box from the **Topics** tab.



As you can see, automatic and custom topic relations give you a lot of options for navigating your Help project.

For example:

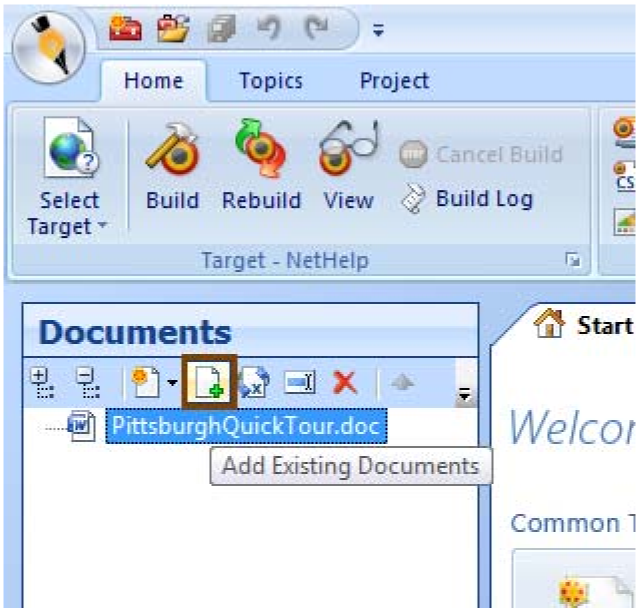
- You can create custom links to other topics that users may find useful. And the links are very clear and noticeable.
- You can create custom links from subtopics to their "parent" topics — this directs the user back to the parent, which may lead them to other topics of interest.
- Relating topics eliminates information "deadends" — the user has links to other information and could be saved the trouble of using search, or checking the TOC or index.

Adding a Document

On occasion, you might need to add an existing document to your project after you've created it, so let's do that now.

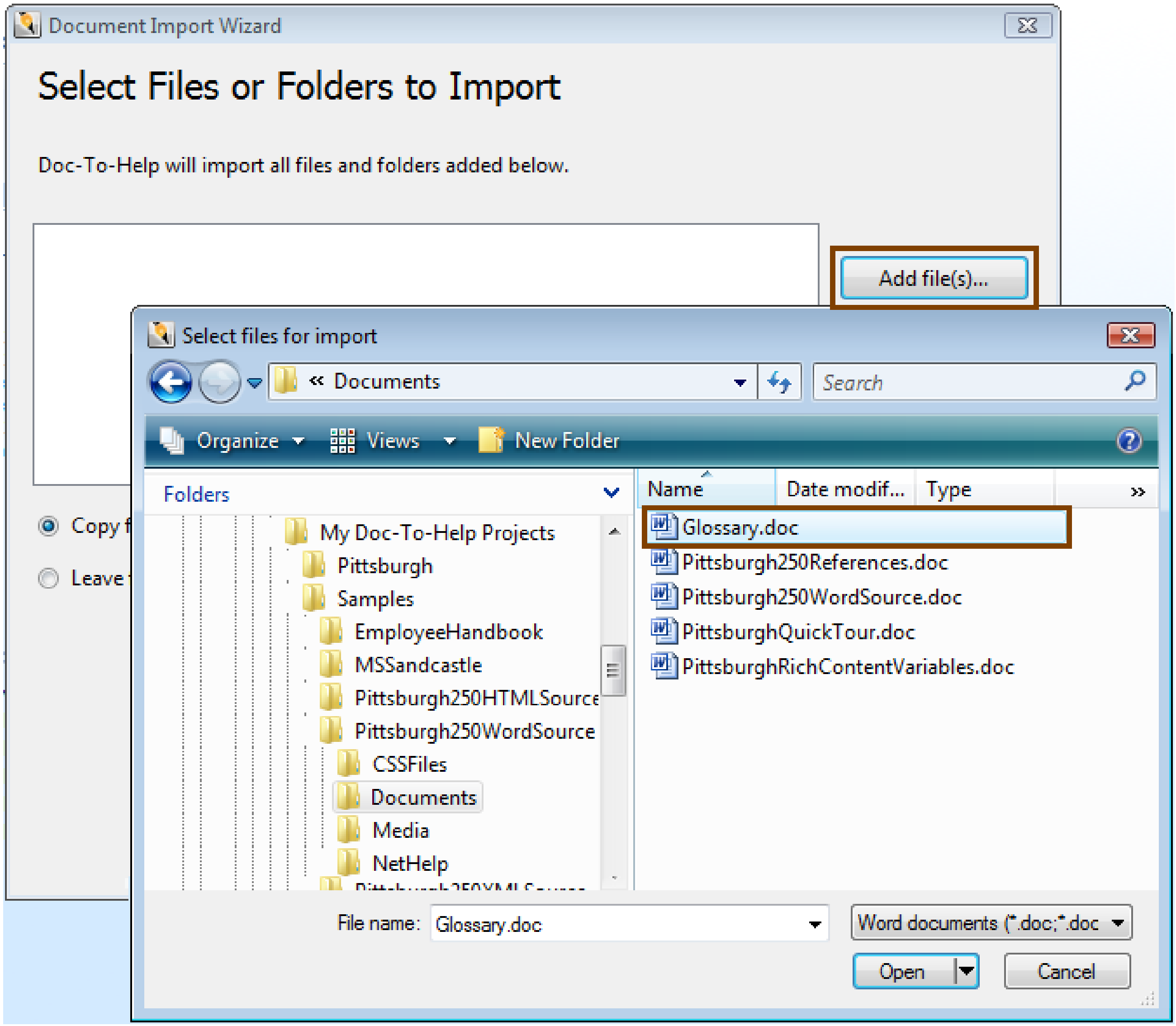
We'll take this opportunity to add a Glossary to our project.

Click the **Add Existing Documents** button in the **Documents pane**.

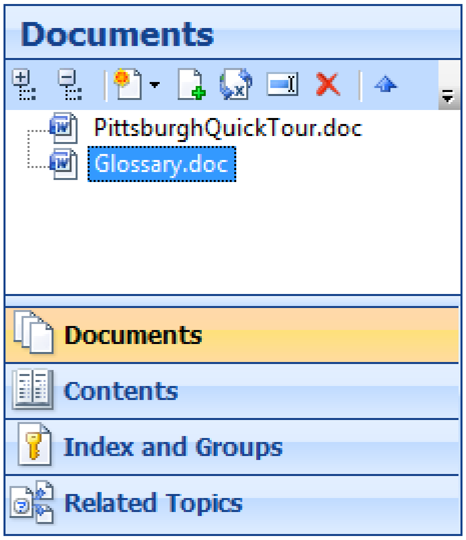


The **Document Import Wizard** will open. Follow the same steps we used when importing **PittsburghQuickTour.doc**, but this time, choose **Glossary.doc** from the \\My Doc-To-Help Projects\Samples\Pittsburgh250WordSource\Documents folder.

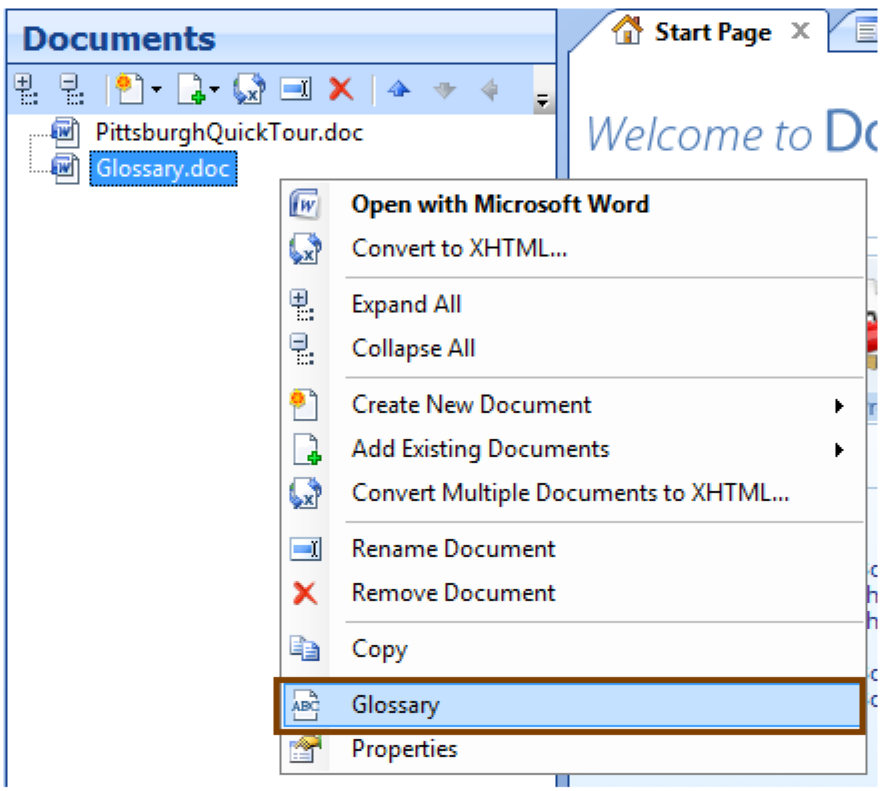
We could have imported this originally, but now you can see how easy it is to do after you have already created your project …



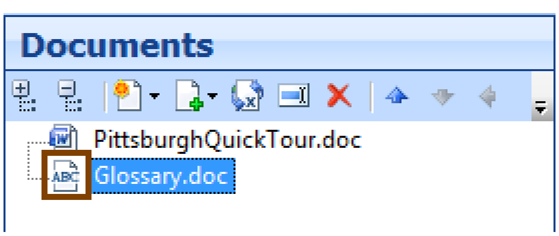
After the import is complete, the new document will appear in the **Documents pane**.



Since a Glossary is a special type of document in Doc-To-Help, right-click on it and choose **Glossary** from the menu.

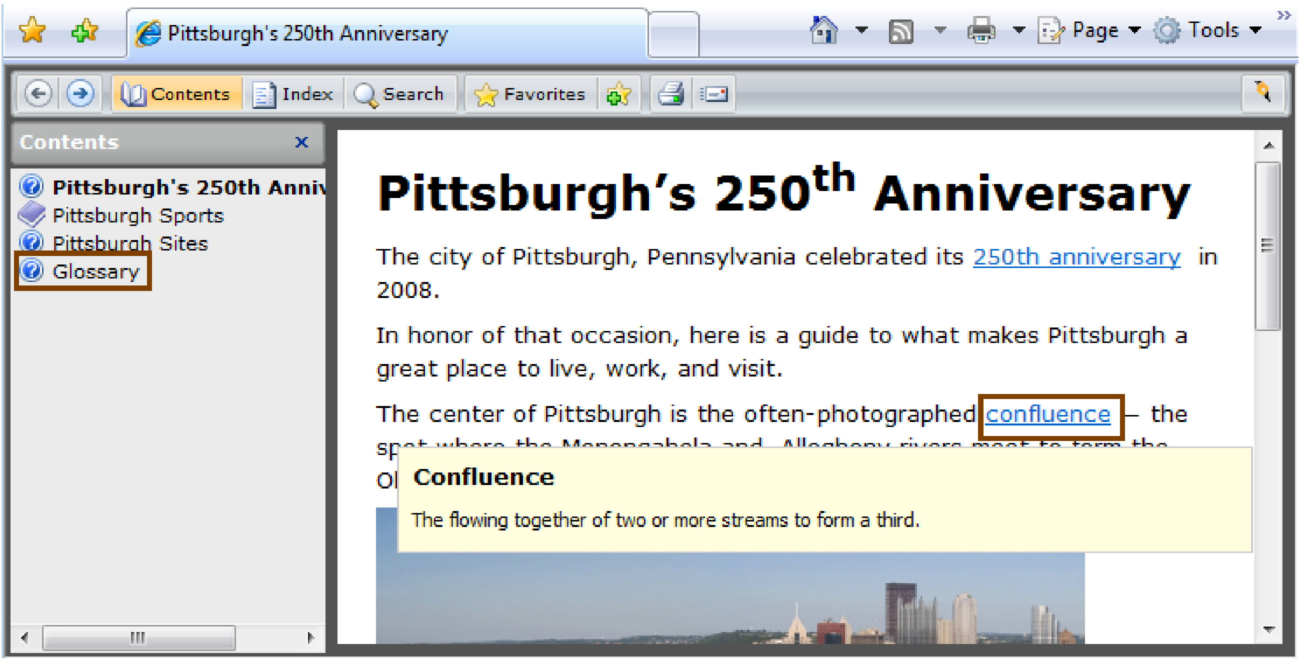


Our document is now flagged as a Glossary with a special icon.



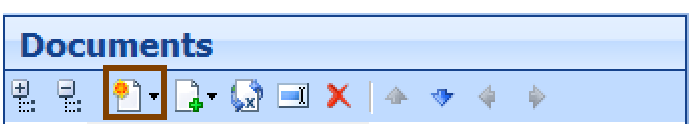
Double-click on **Glossary.doc** to open it. You'll see that two index entries are in this document: Confluence and Primanti Brothers Sandwich. If you put your cursor in either heading, you'll notice that they are formatted with the **Glossary Heading** style. This style has a special behavior in online Help Targets.

Close Glossary.doc and build your Target. You'll notice that **Glossary** has been added to the Table of Contents, plus the word "confluence" is now a hyperlink. When you click on it, the glossary entry is displayed as a popup.



This popup is a Doc-To-Help feature — Doc-To-Help automatically creates links from topic text to identical glossary entries – but only once in each topic. This can be turned off if you like, but it is a great way to give your readers access to useful definitions that they might normally miss, with no extra work on your part. If you would like to create a manual link to a glossary entry, simply create a link using the **Glossary** tab in the **Link** dialog box.

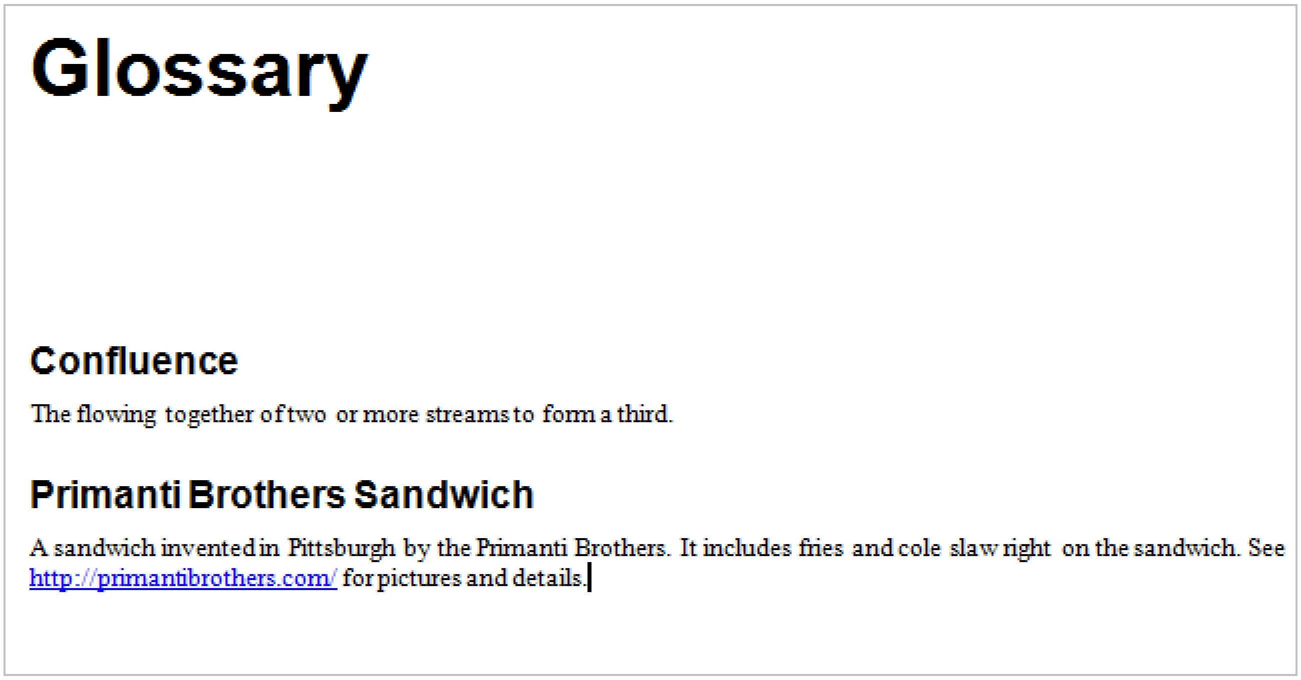
Note: To add a new, empty document to your project, click the **Add New Document** button in the **Documents** pane toolbar.



Adding Glossary Entries

It is easy to add additional entries to your glossary — and you don't need to worry about alphabetizing them because Doc-To-Help will do it for you.

Open **Glossary.doc**. It contains two entries.



Glossary

Confluence

The flowing together of two or more streams to form a third.

Primanti Brothers Sandwich

A sandwich invented in Pittsburgh by the Primanti Brothers. It includes fries and cole slaw right on the sandwich. See http://primantibrothers.com/ for pictures and details.

It's easy to add more entries and alphabetize the list.

After the first two entries, add the following:

Ohio River
A river that begins in Pittsburgh and joins the mighty Mississippi in Illinois.

Now, we need to apply styles to the new entry. These styles not only control the look of the entries, but also the behavior.

Click on **Confluence**, you'll notice the style is **Glossary Heading**. The definition is **C1H Popup Topic Text**. Apply those styles to **Ohio River** and its definition. (Use the same method we used when we applied the **C1H Bullet** style earlier.)

It will look like this:

Glossary

Confluence

The flowing together of two or more streams to form a third.

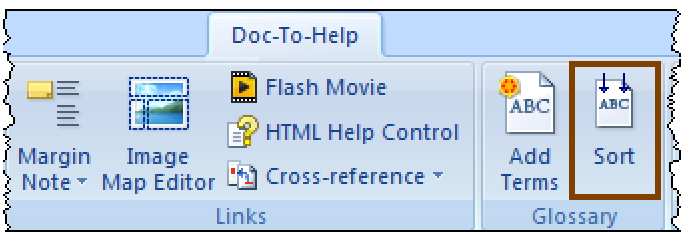
Primanti Brothers Sandwich

A sandwich invented in Pittsburgh by the Primanti Brothers. It includes fries and cole slaw right on the sandwich. See http://primantibrothers.com/ for pictures and details.

Ohio River

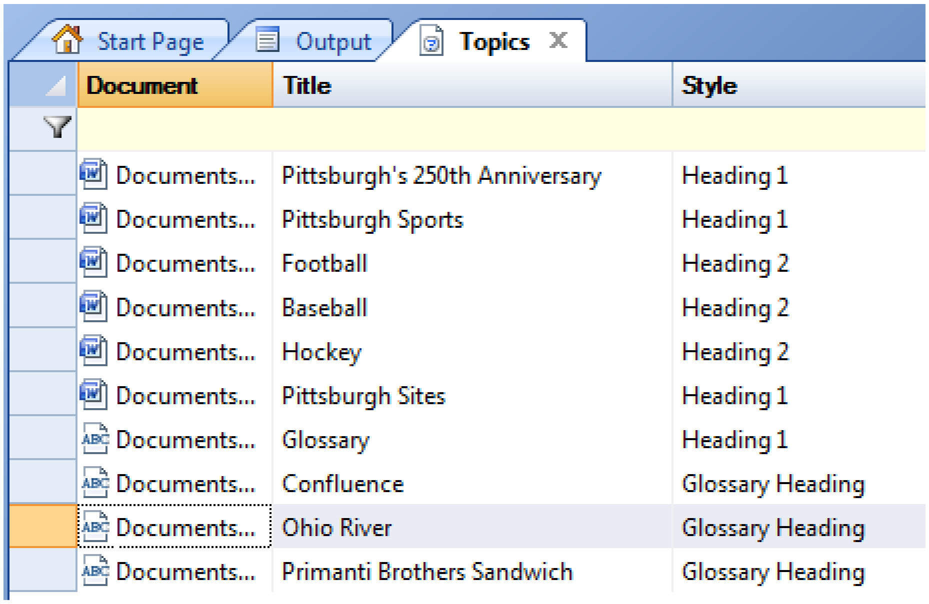
A river that begins in Pittsburgh and joins the mighty Mississippi in Illinois.

Ohio River is in the Glossary, but now the Glossary is not in alphabetical order. To fix that, click the **Sort** button in the **Doc-To-Help** ribbon. The Glossary will now be alphabetized.



Save and close **Glossary.doc**. Build the project.

"Ohio River" is now in the list of topics in the **Topics** window.

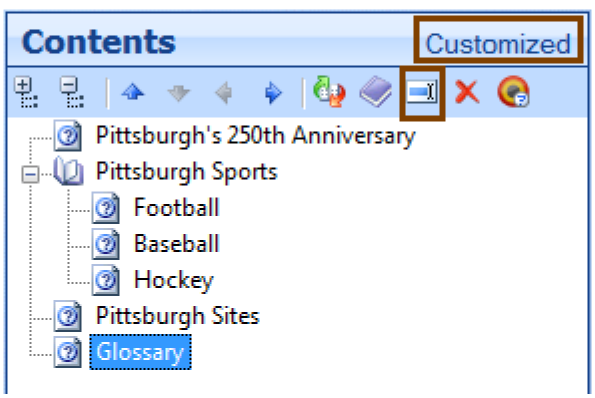


See Creating a Glossary on page 138 for more information.

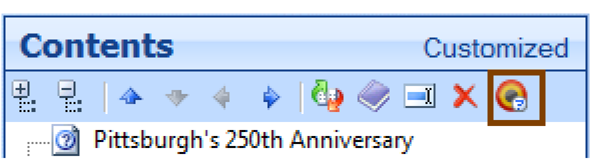
Creating a Customized Table of Contents

As you've already seen, Doc-To-Help automatically creates a Table of Contents for you based on the structure of your documents. But you can customize your Table of Contents if you wish.

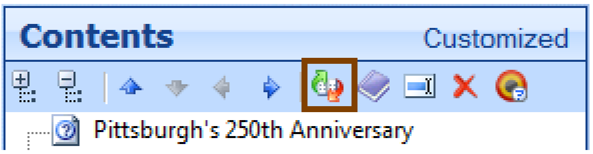
Open the **Contents pane**. If you select an item in the Table of Contents and click **Remove Topic** to remove it — or drag a topic from the **Topics window** into the **Contents pane** to add it — the table of contents will be flagged as **Customized**. Changing the name of an item in the TOC (using the **Rename** button) will also customize it.



If you would like this table of contents to be exclusively used for the Target selected, click the **Target-Specific Table of Contents** button.



If you decide that you would prefer to return to the original table of contents (based on the document structure) click the **Rebuild Table of Contents** button.

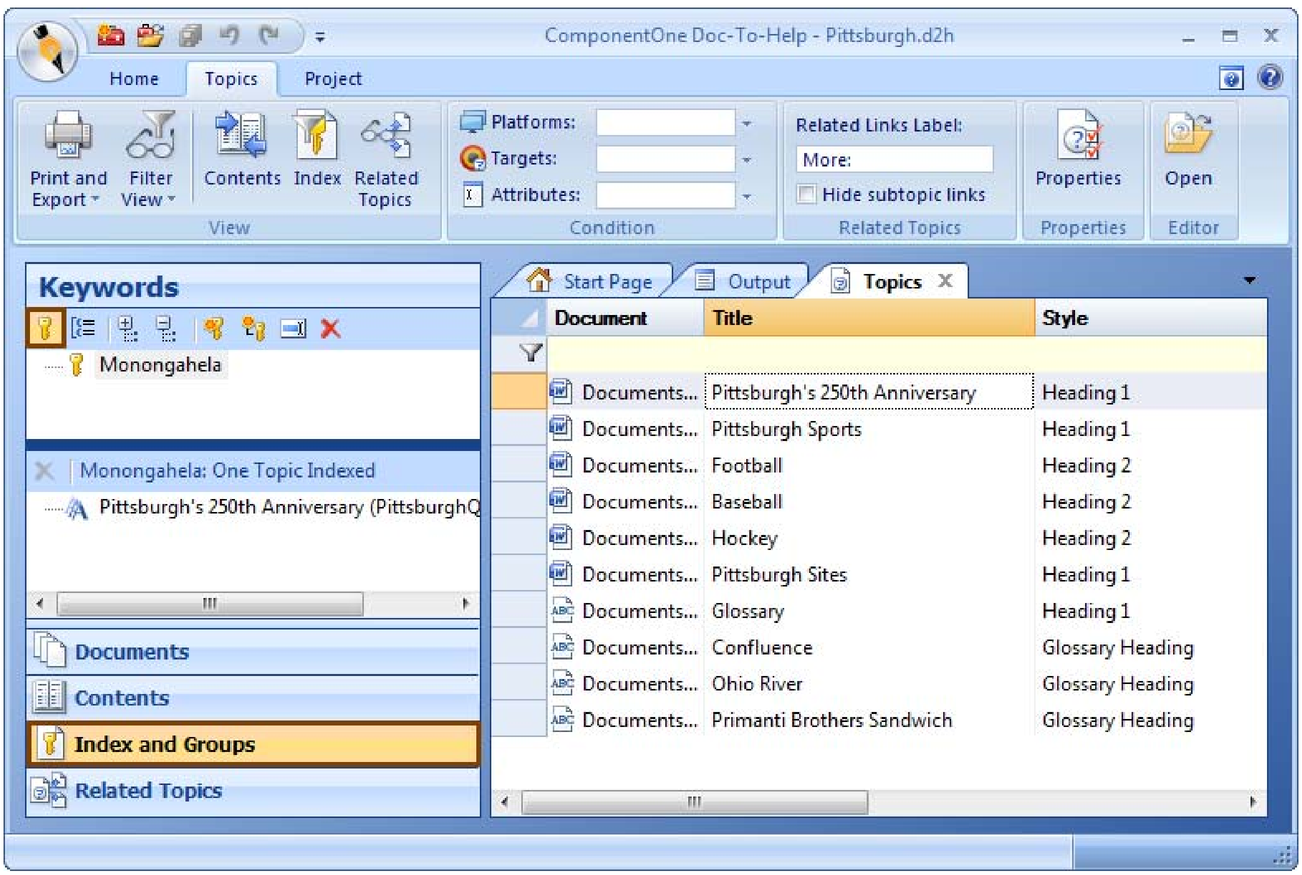


See Contents pane on page 109 for more information on TOCs.

Adding Items to the Index and Creating Groups

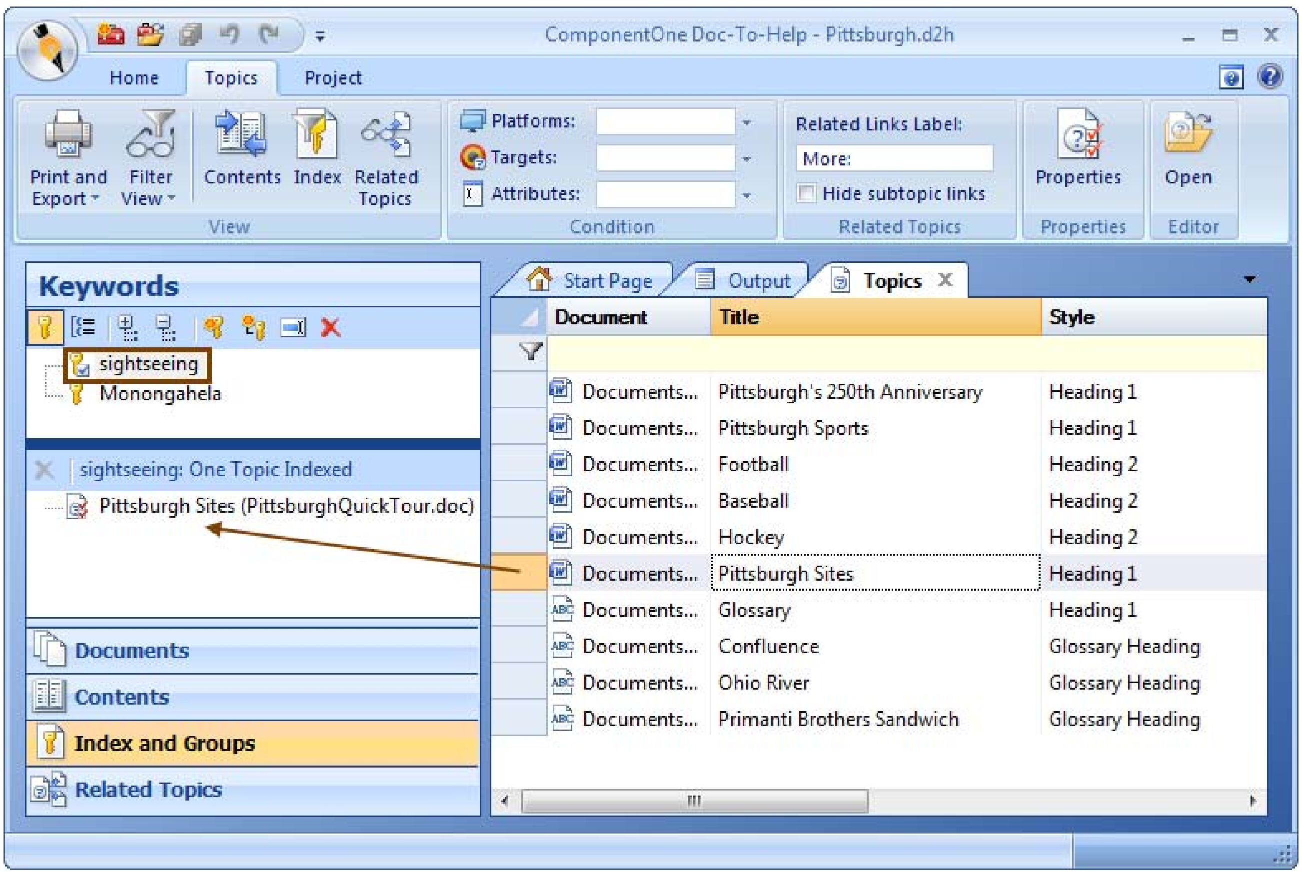
There is already one item in our index, which we added directly from one of our documents. We can also add items using the **Index and Groups pane**, along with the **Topics window**.

Click on the **Index and Groups** pane, and click the **Keyword** button.



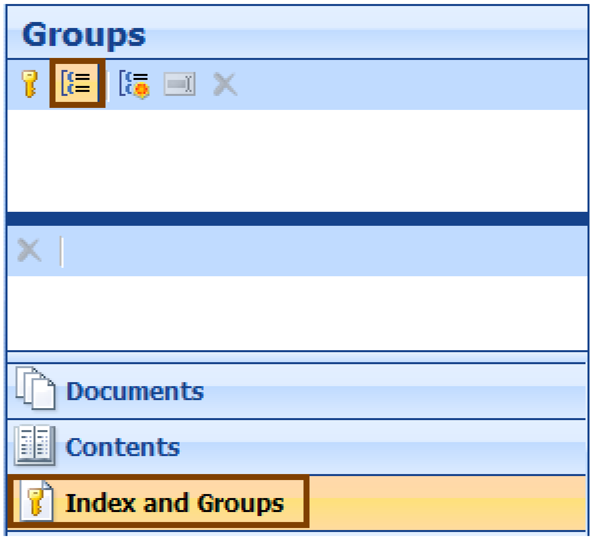
Click the **Add New Keyword** button.

Name the keyword "sightseeing." Then select the topic "Pittsburgh Sites" from the **Topics window** and drag it into the lower **Keywords** pane. You can link multiple topics to a single keyword; just drag them into the lower pane also.



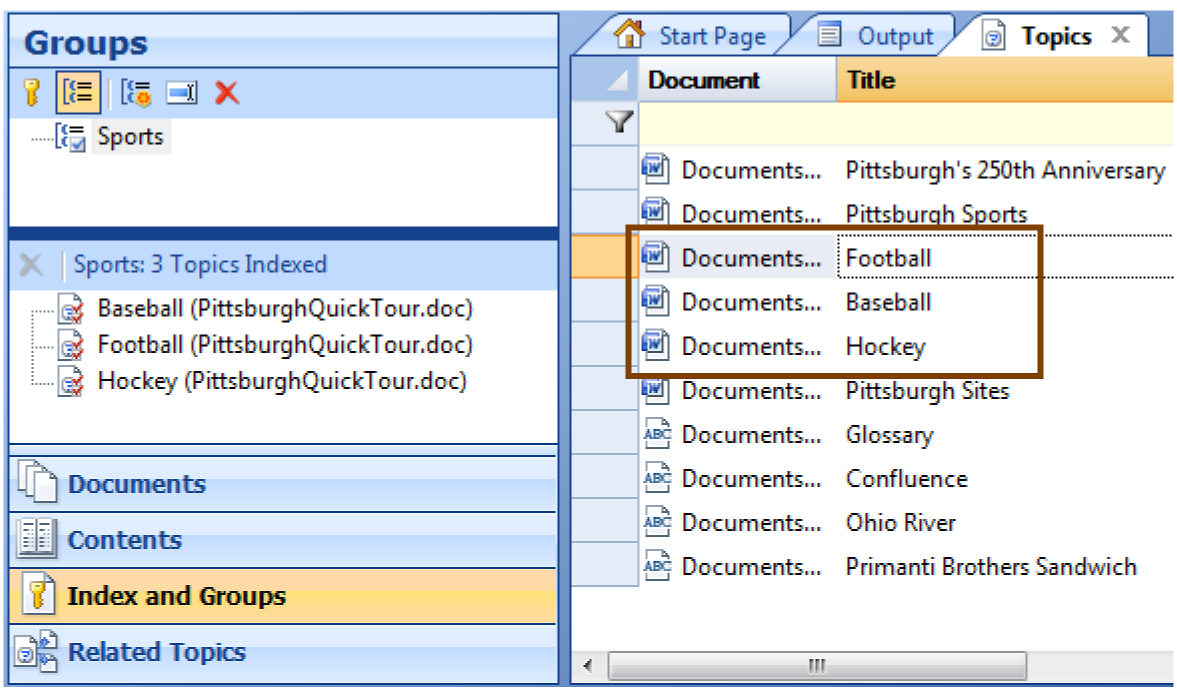
You can create topic **Groups** using this same pane. Topic groups are collections of topics you can create links to. When the user clicks the link, the topics in the group will display in a popup window or dialog box.

First, click the **Groups** button in the **Index and Groups** pane.



Click the **Add Group** button and name the new group "Sports."

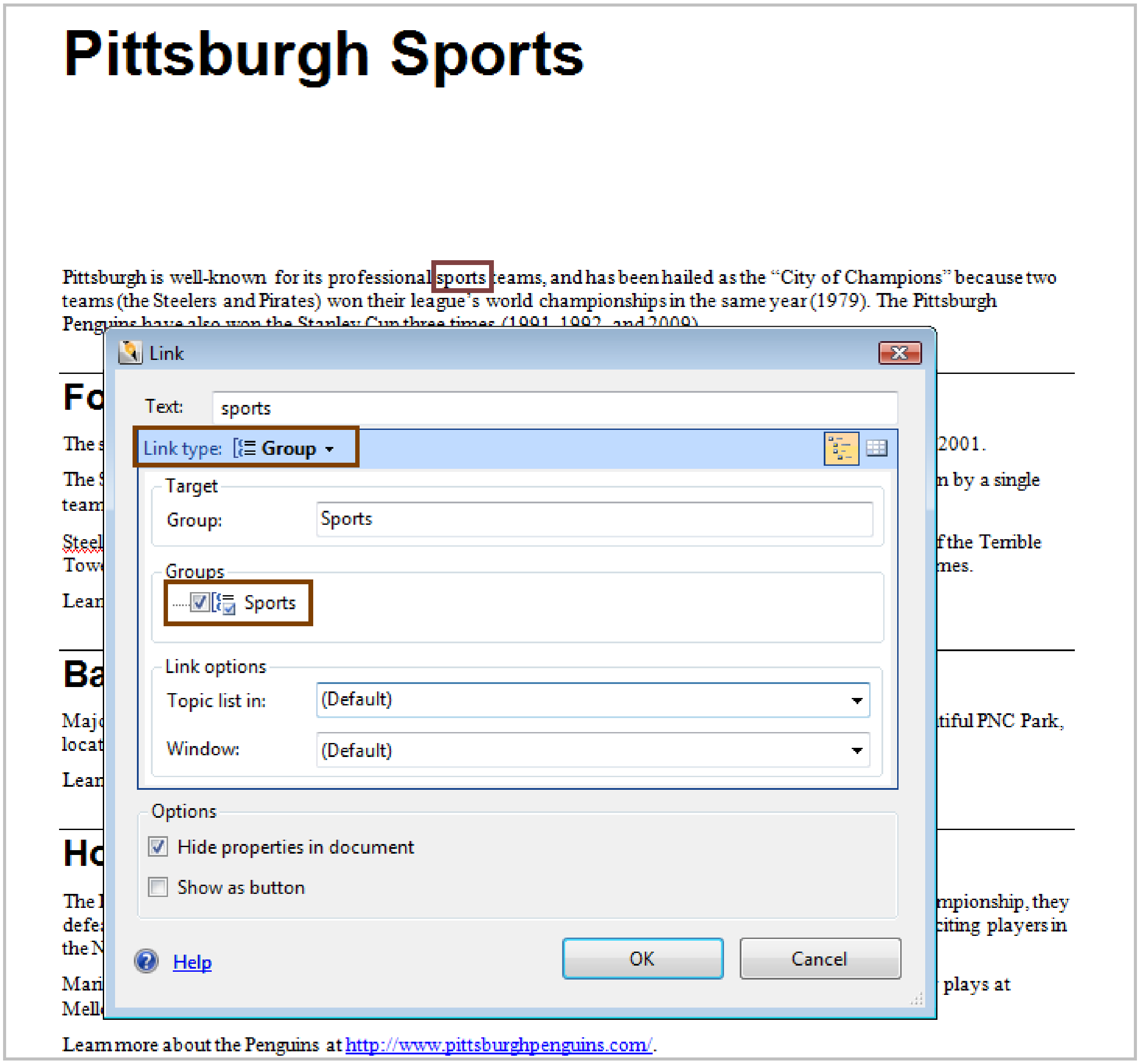
Select the topics "Football", "Baseball", and "Hockey" in the **Topics window** and drag them into the lower **Groups** pane.



These topics are now a group named "Sports" and you can create a link to them using the **Link** dialog box.

Open **PittsburghQuickTour.doc** and in the "Pittsburgh Sports" topic, select the word "sports."

Click the **Link** button on the **Doc-To-Help** ribbon tab and choose **Group** from the **Link type** drop-down. Select the **Sports** check box and click **OK**.

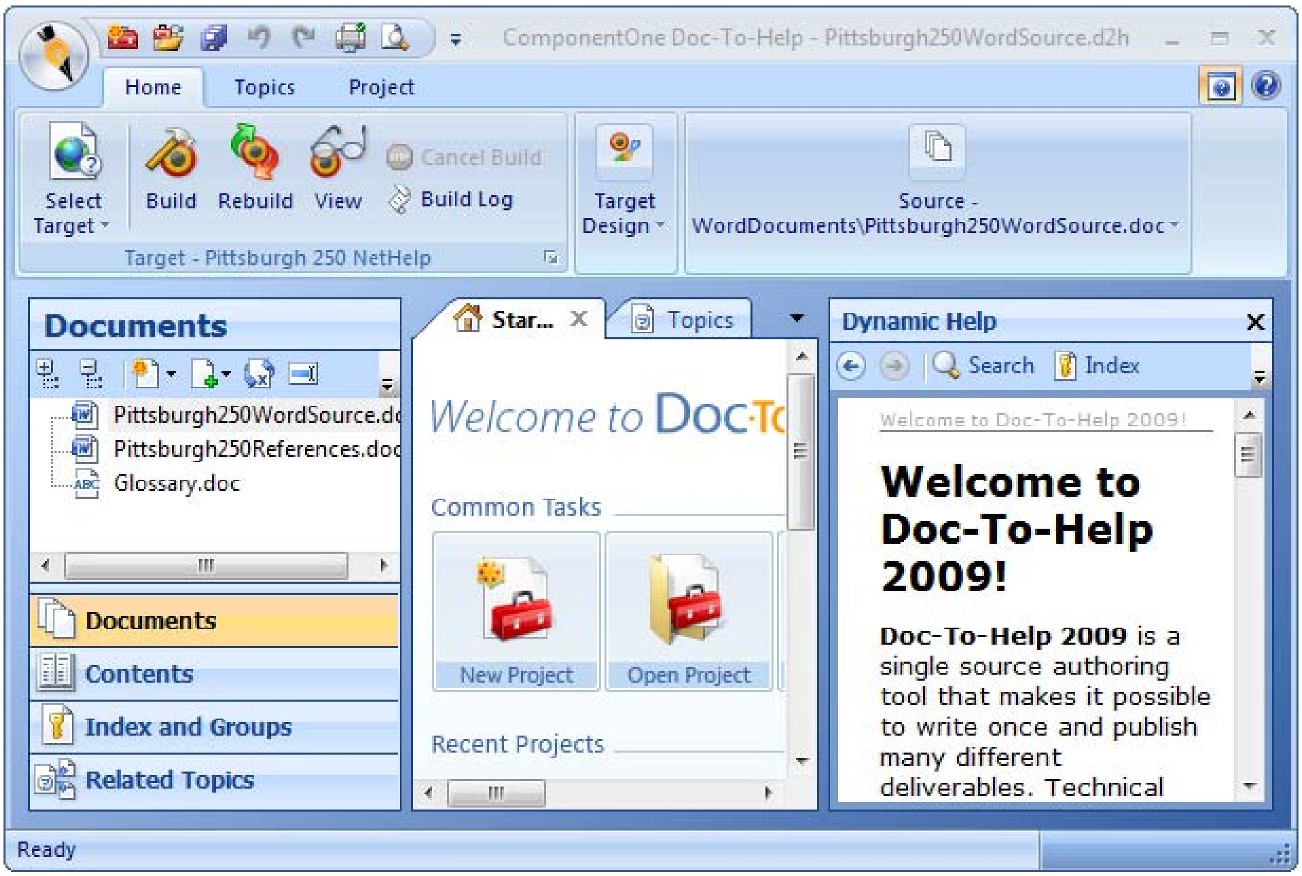


Save and close your document. Build your Target.

Now, "sports" is a hyperlink, and when you click on it, a menu with all of the topics in that group displays. Select any one of the menu items, and the topic opens.

We hope you enjoyed the Doc-To-Help Guided Tour! Feel free to use this project to test-drive Doc-To-Help further.

Navigating Doc-To-Help



Doc-To-Help 2009 features a crisp, clean interface that incorporates the new (and easy-to-use) Ribbons, Navigation Panes, Tabbed Windows, Super ToolTips, and Dynamic Help.

Doc-To-Help button

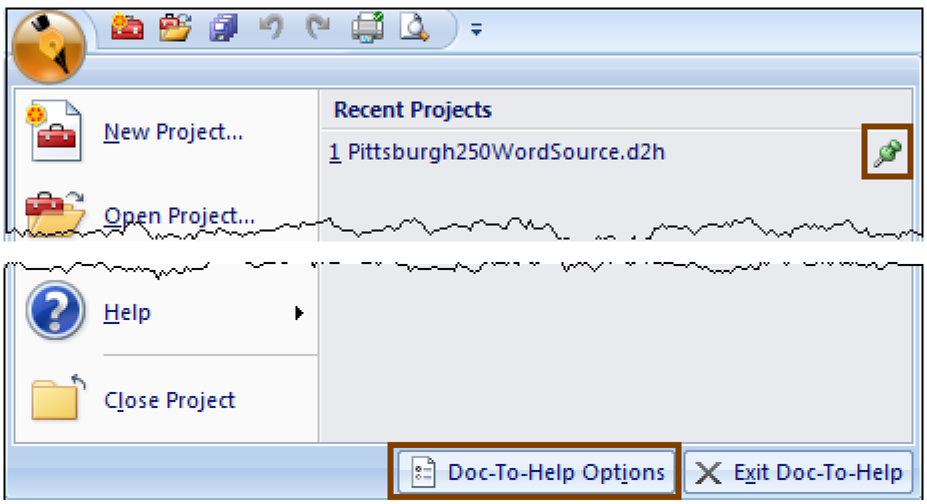
The **Doc-To-Help button** is located in the upper left corner of Doc-To-Help 2009.

Click on it to:

- Create or open a project (on page 139)
- Save a copy of your project (on page 139)

- Save XHTML documents (on page 127)
- Print open XHTML documents. You have the option to **Print**, **Quick Print**, or **Print Preview**.
- Convert an existing legacy project (on page 139) (RoboHelp® HTML, HTML Help, RoboHelp® Word, WinHelp, and Doc-To-Help 2000)
- Run reports (on page 239)
- Access the **Tools** menu, where you can set options (on page 45) (Startup options, Default Editors, and more), deactivate (on page 45) Doc-To-Help, check for updates (on page 45), compact (on page 198) your project, or import/export project settings (on page 199)
- Create a Team Authoring project (on page 241)
- Access the Doc-To-Help online Help and manual, links to online resources, and version information

You can pin projects you open often to the **Recent Projects** list. (Click the green pushpin.) Another way to set options: click the **Doc-To-Help Options** button.



Quick Access toolbar

The **Quick Access toolbar** provides easy access to commonly-used functions.

New Project — Create a new Doc-To-Help project using the **New Project Wizard**.

Open Project — Open an existing Doc-To-Help project.

Save — Save changes in one click to all your open XML project documents. Please note that project documents open in other applications (such as Microsoft® Word) must be saved separately.

Undo — Undo your last action.

Redo — Redo your last undo.

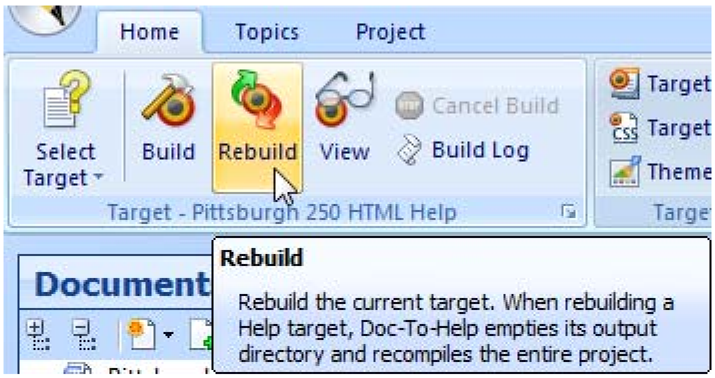
Quick Print — Print the displayed XHTML document.

Print Preview — Preview the displayed XHTML document.

Customize Quick Access button — This button (the arrow on the right) can be used to move the Quick Access toolbar (choose **Show Above/Show Below the Ribbon**), or minimize/maximize the Ribbons (select **Minimize the Ribbon**).

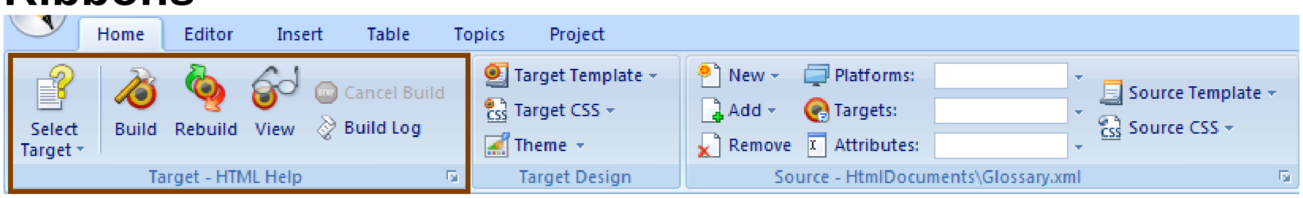
The **Quick Access toolbar** is generally found next to the Doc-To-Help button (on page 95), unless it is moved below the Ribbon using the **Customize Quick Access button**.

Super ToolTips



Super ToolTips are expanded tooltips that are displayed when you hover over a button. They display the button name and a description of what it does.

Ribbons



Doc-To-Help 2009 features Microsoft® Office 2007 style ribbons.

Ribbons replace menus and toolbars, and are easier-to-use because they group features by common tasks.

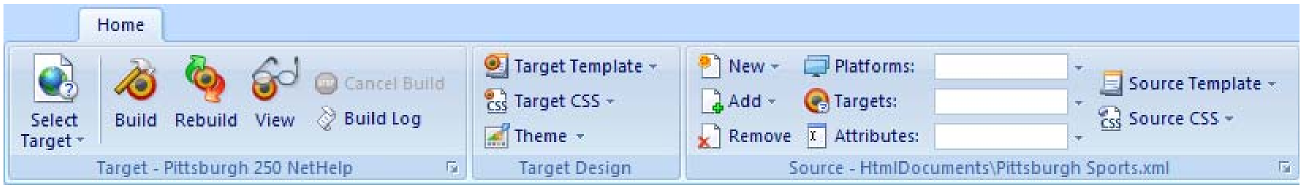
Doc-To-Help has seven ribbons: Home (on page 98), Editor (on page 99), Insert (on page 103), Table (on page 101), Topics (on page 104), Project (on page 106), and Team Authoring (on page 107). The **Home**, **Topics**, and **Project** tabs are displayed by default; the **Editor**, **Insert**, and **Table tabs** appear only when an XHTML file is opened for editing within Doc-To-Help. The **Team Authoring tab** opens for team projects only.

Each ribbon is divided into logical ribbon groups (the **Target ribbon group** is highlighted above). Many dialog boxes can be opened directly from ribbon groups using dialog box launchers. (The dialog box launchers are the small arrows on the bottom right of specific ribbon groups.)

You can minimize the Ribbons using the **Customize Quick Access** toolbar button. See Quick Access toolbar on page 96 for details.

See Navigating Doc-To-Help on page 95 for information on other navigation methods.

Home tab



The **Home tab** provides access to the most frequently used functions in Doc-To-Help, including:

- Selecting, building, and viewing targets (see Building a Target on page 235)
- Assigning templates and style sheets
- Setting document conditions
- Defining the look of your target(s)

See Ribbons on page 97 for more information on Doc-To-Help's ribbons.

Target ribbon group (Home tab)

Tools in this ribbon group:

See Building a Target on page 235 for more information.

Select Target — Select the target using the drop-down list. The icon above it will display the target type.

Build — Build the current target.

Rebuild — Rebuild the current target. When rebuilding a Help target, Doc-To-Help empties its output directory and recompiles the entire project.

View — View the current Help target output.

Cancel Build — Cancel the current build.

Build Log — View the Output window (on page 113), which displays the build log, as well as errors and unresolved links.

View PDF — View the PDF of the Manual target. (Only visible if the Manual target has been generated.)

Click the dialog box launcher to open the Help Targets dialog box on page 155.

Target Design ribbon group (Home tab)

Tools in this ribbon group:

Target Template — This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5.

Target CSS — This style-sheet controls the formatting of the Target for HTML source documents. See the Guide to Templates and Styles on page 5. Also see Editing a CSS on page 189.

Theme — Select and edit the theme (skin) for the current help target. Click this button to open the Theme Preview. See Customizing with the Theme Designer on page 11.

Source ribbon group (Home tab)

Tools in this ribbon group:

Create New Document — Add a new XHTML, HTML, or Word document to the project. See Adding a Document to a Project on page 116.

Add Existing Document — Add an existing XHTML, HTML, or Word document to the project.

Remove Document — Remove the selected document from the project.

Platforms — Set a platform-based condition for the selected document. The document will be included in all of the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets — Set a target-based condition for the selected document. The document will be included in all the target(s) selected.

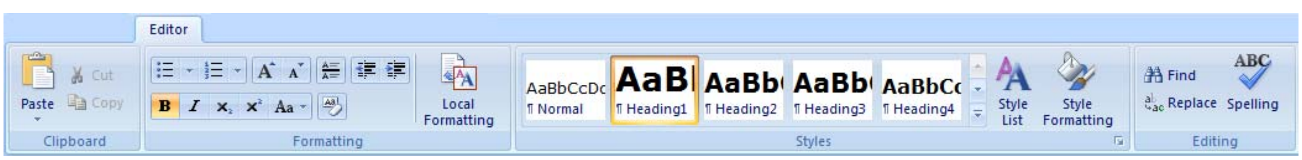
Attributes — Set an attribute-based condition for the selected document. The document will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > Attributes button) to create custom attributes.

Source Template — The default template for new Word source documents added to the project. To change the look of the Target, edit the Target template. Styles that exist only in the Source Template will pass-through to the Target Template. See the Guide to Templates and Styles on page 5.

Source CSS — The default style sheet for new HTML source documents added to the project. To change the look of the Target, edit the Target CSS. Styles that exist only in the Source CSS will pass-through to the Target CSS. See the Guide to Templates and Styles on page 5. Also see Editing a CSS on page 189.

Click the dialog box launcher to open the Document Properties dialog box on page 136.

Editor tab



The **Editor tab** contains handy tools to edit XHTML documents, including:

- Cut/Copy/Paste text
- Format text
- Apply and create styles
- Find and replace text
- Check spelling

The Editor tab is only available when editing an XHTML document within Doc-To-Help.

See XHTML Editor window on page 113 for more on importing/converting documents to XHTML.

See Ribbons on page 97 for more information on Doc-To-Help's ribbons.

Clipboard ribbon group (Editor tab)

Tools in this ribbon group:

Paste — Paste the contents of the clipboard.

Cut — Cut the selection and put it on the clipboard.

Copy — Copy the selection and put it on the clipboard.

Formatting ribbon group (Editor tab)

Tools in this ribbon group:

Bullets — Start a bulleted list.

Numbers — Start a numbered list.

Big Font — Increase the font size.

Small Font — Decrease the font size.

Horizontal Line — Insert a horizontal line.

Decrease Indent — Decreases the indent level of the paragraph.

Increase Indent — Increases the indent level of the paragraph.

Bold — Make the selected text bold.

Italic — Italicize the selected text.

Subscript — Create small letters below the text baseline.

Superscript — Create small letters above the line of text.

Change Case — Change selected text to UPPERCASE, lowercase, or other common capitalizations.

Clear Formatting — Clear all formatting from the selection, leaving only the plain text.

Local Formatting button — If this button is pressed, you will have several options for applying formatting without styles to the current document. Please note that applying formatting without styles is not a good best practice, because it makes your documents less standards-compliant and makes it harder to maintain consistency among documents.

> **Font** — Change the Font face.
>
> **Font Size** — Change the Font size.
>
> **Underline** — Underline the selected text.
>
> **Strikethrough** — Draw a line through the middle of the selected text.

Text highlight color — Make text look as if it was marked with a highlighter pen.

Font color — Change the text color.

Align Text Left — Align text to the left.

Align Text Center — Center text.

Align Text Right — Align text to the right.

Align Text Justify — Align text to both the left and right margins, adding extra space between words as necessary.

Styles ribbon group (Editor tab)

Tools in this ribbon group:

Style Gallery — Use these buttons to apply styles with a single click. The styles displayed can be changed with the Manage Styles dialog box (on page 122) (click the dialog box launcher to open it).

Style List — Opens the **Styles** dialog box, where you can modify and create styles, as well as select a style to apply.

Style Formatting — Opens the **Style Formatting** editor for quick formatting.

See Editing a CSS on page 189 for more information about adding and modifying styles.

Editing ribbon group (Editor tab)

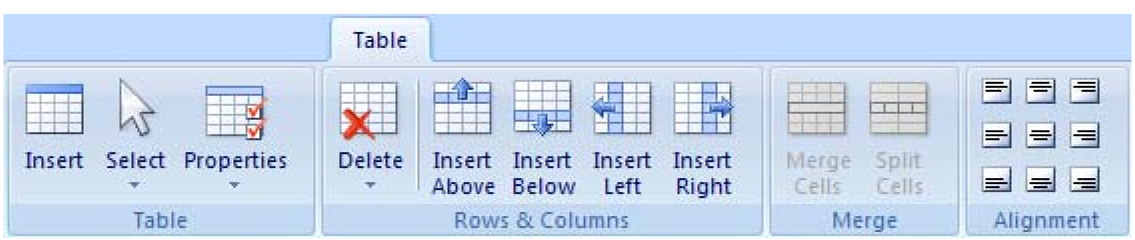
Tools in this ribbon group:

Find — Find text in the document, or the entire project.

Replace — Replace text in the document, or the entire project.

Spelling — Check the spelling in the document. (To set spelling options, use the **Doc-To-Help** button > **Doc-To-Help Options** button > **Spelling** tab.)

Table tab



The **Table tab** allows you to work with tables in XHTML documents:

- Insert tables
- Change table properties
- Delete tables

- Insert rows and columns
- Merge and split cells
- Align table text

The Table tab is only available when editing an XHTML document within Doc-To-Help.

See Ribbons on page 97 for more information on Doc-To-Help's ribbons.

Table ribbon group (Table tab)

Tools in this ribbon group:

Insert — Insert a table into the document.

Select — Select the current cell, row, column, or entire table using the drop-down list.

Table Properties — Opens the **Properties** dialog box to change the current cell, row, column, or table properties.

See Inserting Tables in the XHTML Editor (on page 125) and Viewing and Modifying Table Cells, Rows, and Columns in the XHTML Editor (on page 125) for more information.

Rows & Columns ribbon group (Table tab)

Tools in this ribbon group:

Delete — Delete selected cells, rows, columns, or the entire table.

Insert Rows Above — Insert a row above the selected one.

Insert Rows Below — Insert a row below the selected one.

Insert Columns to the Left — Insert a column to the left of the selected one.

Insert Columns to the Right — Insert a column to the right of the selected one.

Merge ribbon group (Table tab)

Tools in this ribbon group:

Merge Cells — Merge selected cells into one cell.

Split Cells — Split the selected cells into multiple new cells.

Alignment ribbon group (Table tab)

Tools in this ribbon group:

Align Top Left — Align text to the top left corner of the cell.

Align Top Center — Center text and align it to the top of the cell.

Align Top Right — Align text to the top right corner of the cell.

Align Center Left — Center text vertically and align it to the left side of the cell.

Align Center — Center text horizontally and vertically within the cell.

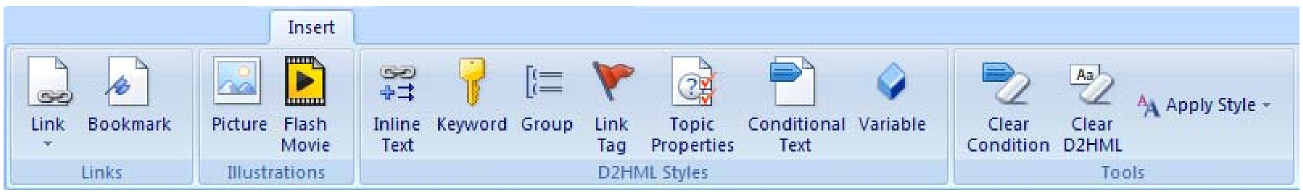
Align Center Right — Center text vertically and align it to the right side of the cell.

Align Bottom Left — Align text to the bottom left corner of the cell.

Align Bottom Center — Center text and align it to the bottom of the cell.

Align Bottom Right — Align text to the bottom right corner of the cell.

Insert tab



The **Insert tab** allows you to apply D2HML (Doc-To-Help Markup Language) styles to your XHTML documents, including:

- Links
- Inline Text
- Keywords and Groups
- Link Tags
- Topic Properties
- Conditional Text
- Variables

This tab is also used to create hyperlinks, and insert bookmarks, pictures and Flash movies.

The Insert tab is only available when editing an XHTML document within Doc-To-Help. See Using D2HML (Doc-To-Help Markup Language on page 210.

For more on Styles, see Defining Character/Paragraph Styles and Topic Types on page 181.

See Ribbons (on page 97) for more information on Doc-To-Help's ribbons.

Links ribbon group (Insert tab)

Tools in this ribbon group:

Link — Create or edit a D2HML link to another topic or a hyperlink. See Creating Links (on page 212) or Creating Hyperlinks in the XHTML Editor (on page 124) for more information. Use the drop-down to choose **In Project** or **External**.

Bookmark — Insert or edit a bookmark. See Creating Bookmarks in the XHTML Editor (on page 124) for more information.

Illustrations ribbon group (Insert tab)

Tools in this ribbon group:

Picture — Insert or edit an image. See Inserting Images in the XHTML Editor (on page 123) for more information.

Flash Movie — Insert a movie in Flash format. See Inserting Flash Movies in the XHTML Editor (on page 123) for more information.

D2HML Styles ribbon group (Insert tab)

Tools in this ribbon group:

Inline Text — Create a hot spot that appears as a link in the help target, and when clicked, shows the inline text. Text can be expanding, drop-down, or in a popup window. See Creating Inline, Dropdown, or Popup Text (on page 215) for more information.

Keyword — Associate one or more index keywords to this topic. See Inserting an Index Entry (on page 215) for more information.

Group — Create a link in the online Help associating one or more topic groups with this topic. See Adding a Topic to a Group (on page 216) for more information.

Link Tag — Assign or edit the link tag (unique identifier) for this topic. See Adding a Link Tag (on page 217) for more information.

Topic Properties — Assign the properties of this topic, including the context ID. See Setting D2HML Topic Properties (on page 218) for more information.

Conditional Text — Mark the selected text so that it will be used in only the specified Target, Platform, or Attribute. See Marking Text as Conditional (on page 220) for more information.

Variable — Insert variable text or content at the selected location. See Inserting a Variable (on page 221) for more information.

Tools ribbon group (Insert tab)

Tools in this ribbon group:

Clear Condition — Remove D2HML Condition formatting from the selected text.

Clear D2HML — Remove all D2HML formatting from the selected text. See Clearing D2HML Styles (on page 222) for more information.

Apply Style — Format the selected text with any of the D2HML styles.

Topics tab

The **Topics tab** allows you to work efficiently in the Topics window. With this ribbon you can:

- Print the Topic list, or export it to Microsoft® Excel® or text (.txt). See Printing and Exporting the Topic List (on page 227) for more information.
- Set topic filters
- Open the **Contents**, **Index** and **Groups**, and **Related Topics** panes for the selected topic
- Apply multiple conditions to entire topics
- Set Related Topics settings (label, subtopic links)
- Edit Topic Properties
- Open selected topics directly

See Ribbons (on page 97) for more information on Doc-To-Help's ribbons.

View ribbon group (Topics tab)

Tools in this ribbon group:

Print and Export — Print the Topics list or export it to Microsoft® Excel® or text (.txt). See Printing and Exporting the Topic List (on page 227) for more information.

Filter view — View topics with or without the selected criteria.

Contents — Synchronize the Contents pane (on page 109) with the selected topic.

Index — Display the keywords for the selected topic only.

Related Topics— Display the Related Topics pane (on page 111) for the selected topic only.

Condition ribbon group (Topics tab)

Tools in this ribbon group:

Platforms — Set a platform-based condition for the selected topic. The topic will be included in all the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets — Set a target-based condition for the selected topic. The topic will be included in all the target(s) selected.

Attributes — Set an attribute-based condition for the selected topic. The topic will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > Attributes button) to create custom attributes.

See Utilizing Conditions (on page 174) for more information.

Related Topics ribbon group (Topics tab)

Tools in this ribbon group:

Related Links Label — This is the label that will appear above Related (Subtopic) Links for the selected topic.

Subtopic Links — Links automatically displayed in a parent topic to all of its subtopics. Select the **Hide** check box to turn off Subtopic Links for the selected topic. See Subtopic Links for more information.

"Hide subtopic links" check box — If this check box is selected, all Subtopic Links will be hidden in the selected parent topic.

See Managing Related Topics (on page 229) for more information.

Properties ribbon group (Topics tab)

Tools in this ribbon group:

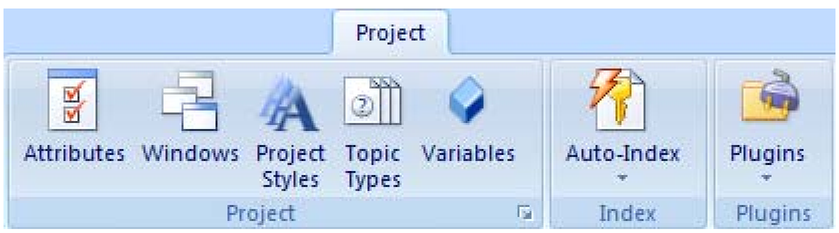
Properties — Click the **Properties** button to open the Topic Properties dialog box (on page 225) for the selected topic.

Editor ribbon group (Topics tab)

Tools in this ribbon group:

Open Topic — Open the selected topic for editing.

Project tab



The **Project tab** is used to specify the global settings for this project. With this ribbon you can:

- Add and edit Attributes
- Add and edit Windows
- Add and edit Character Styles, Paragraph Styles, and Topic Types
- Define Variables
- Set the global Project Settings
- Set Auto-Indexing criteria
- Add Plugins

Project ribbon group (Project tab)

Tools in this ribbon group:

Attributes — Opens the Attributes dialog box (on page 175) where you can add or edit project attributes. Use these attributes to tag content for conditional builds. Attribute conditions can be applied to text, topics, and/or entire documents.

Windows — Opens the Windows dialog box (on page 176), where you can edit the default window types. Custom windows can also be created.

Project Styles — Opens the Project Styles dialog box (on page 181), where you can customize the source Paragraph and Character styles. These styles are from the Source Template. You can also add new styles.

Topic Types — Opens the Project Styles dialog box, where you can add or edit Topic Types. A topic type is assigned to each topic. You can customize the default topic types or create new ones.

Variables — Opens the Variables window (on page 114), where you can create and edit plain and rich content variables.

Click the dialog box launcher to open the Project Settings dialog box (on page 196).

Index ribbon group (Project tab)

Tools in this ribbon group:

Auto-Index — Automatically create the index based on one or more topic types or styles. Open the drop-down menu to set preferences.

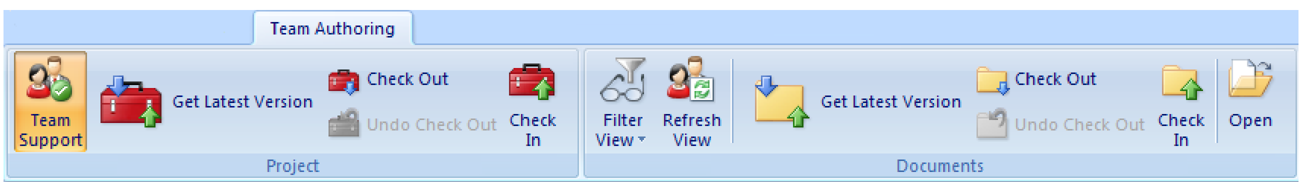
See Creating an Index or Groups (on page 231) for more information.

Plugins ribbon group (Project tab)

Tools in this ribbon group:

Plugins — Add new or existing plugin documents. For example, the Sandcastle plugin allows authors to use reference documentation as part of the Doc-To-Help project. See Documenting Your Class Library with Microsoft® Sandcastle on page 265.

Team Authoring tab



The **Team Authoring tab** provides access to Doc-To-Help's team functionality, including:

- Enabling team support
- Project and Document check in/check out

See Working on a Team on page 241 for more information.

See Ribbons (on page 97) for more information on Doc-To-Help's ribbons.

Project ribbon group (Team Authoring tab)

Tools in this ribbon group:

Team Support — Enable team authoring support.

Get Latest Version — Get the latest version of the project and documents from the repository.

Check Out — Check project out of the repository.

Undo Check Out — Undo the last checkout operation.

Check In — Check project and documents into the repository.

Documents ribbon group (Team Authoring tab)

Tools in this ribbon group:

Filter View — Select the files that will be displayed in the Team Authoring window.

Refresh View — Refresh the current view to see file status.

Get Latest Version — Get the latest version of the project and documents from the repository.

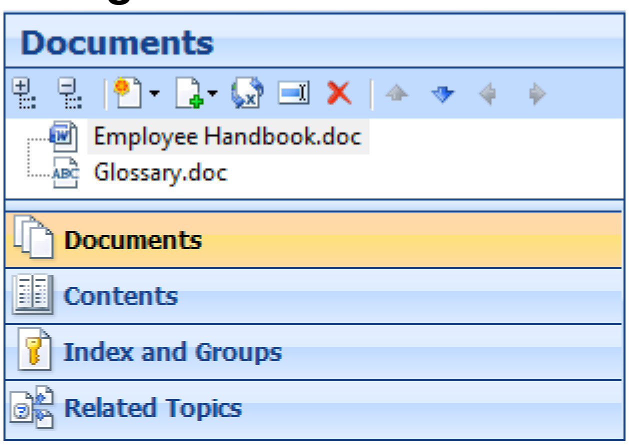
Check Out — Check documents out of the repository.

Undo Check Out — Undo the last checkout operation.

Check In — Check documents into the repository.

Open — Open the file selected in the Team Authoring window for editing.

Navigation Panes



Doc-To-Help's navigation panes make it possible to quickly work with your documents, table of contents, index, and related topics. Click on the button to expand the pane and open the accompanying window (on page 112).

See Navigating Doc-To-Help (on page 95) for information on other navigation methods.

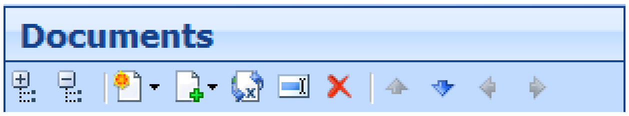
Documents pane

The Documents pane is "home base" in Doc-To-Help.

Use the **Documents pane** to:

- View all project documents (.doc, .docx, .html, .htm, .xml)
- Open a project document (double-click on it)

Using the **Documents pane toolbar**:



- Expand/collapse the list
- Add new and existing documents (See Adding a Document to a Project on page 116 for more information.)
- Convert documents to XHTML (See Converting Existing Documents to XHTML on page 117 for more information.)
- Rename a document
- Remove a document from the project
- Rearrange your documents

Right-click on a document to:

- Open it
- View document properties
- Convert Word or HTML documents to XHTML
- Designate a document as the project Glossary
- Create a link to the document (choose **Copy Link**). The link created can be pasted into any document.

Note: You can open .xml files within Doc-To-Help (in the **XHTML Editor window**); .doc and .docx files, and .htm and .html files are edited in external editors. To set your default HTML editor, see Setting Doc-To-Help Options on page 45.

Using the Source ribbon group (on page 99) (Home tab) you can:

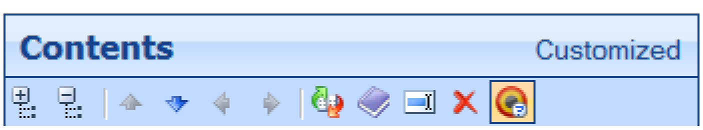
- Assign Source Templates and CSSs to your documents
- Designate Platform, Target, and Attribute-based conditions
- View Document Properties (on page 136) (click the dialog box launcher)

Contents pane

Use the **Contents pane** to:

- View/Edit your table of contents

Using the **Contents pane toolbar**:



- Expand/collapse the list
- Rearrange your topics
- Rebuild your TOC based on your document structure (Doc-To-Help's automatic TOC)
- Create a book
- Rename a topic/book

- Remove a topic/book
- Designate the displayed TOC as Target specific. (Choose the target using the Select Target button on page 98 on the **Home** tab.)

Right-click on a TOC item to:

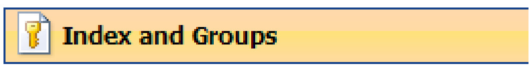
- Convert that topic to a book
- Sort the topic list alphabetically

Note: You can create custom TOCs for each Help target. To do so, choose the appropriate Help Target from the **Home tab**, then click the **Target-Specific Table of Contents button** in the Contents pane. When you build the Target, a Target-specific TOC will be created. Target-specific TOCs will be flagged as "Customized" in the Contents pane. When you select a Target using the **View Target button**, the Contents pane will display the correct TOC for that Target.

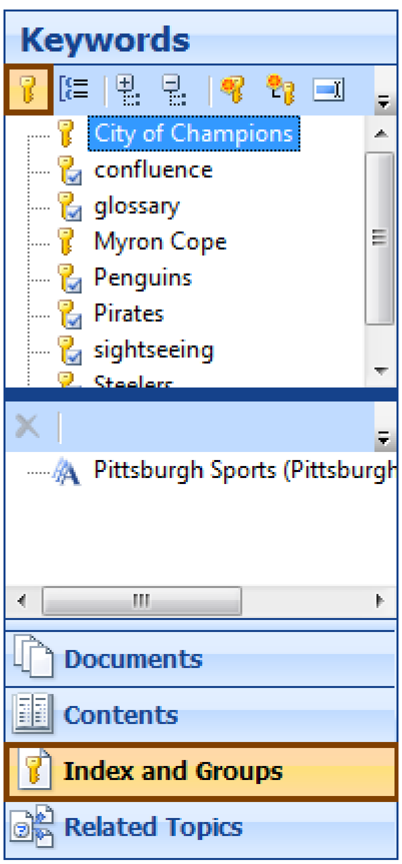
See Creating a Table of Contents (on page 232) for more information.

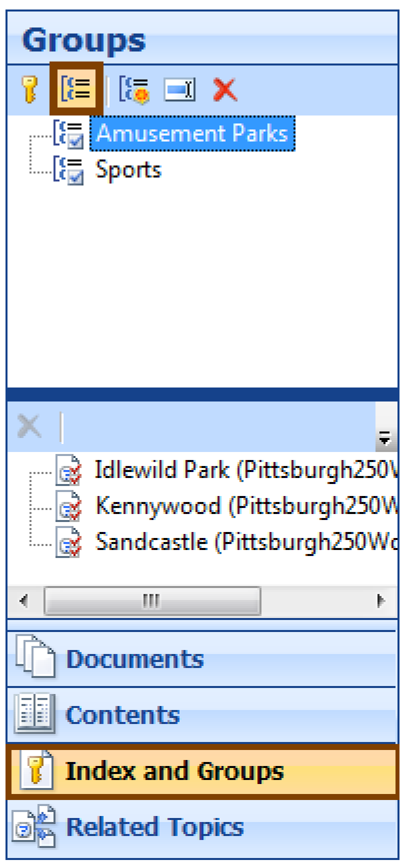
Index and Groups pane

The display in this pane varies based on the toolbar button chosen.



Choosing the **Index button** (default) displays the project Keywords (Index); choosing the **Groups button** displays the project Groups.



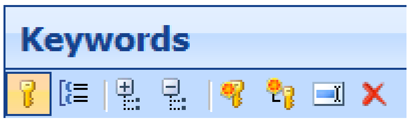


Keywords with a check box next to them have been added within the **Index and Groups pane**. Those without check boxes were added within the source documents.

Use the **Index and Groups pane** to:

- View/Edit your index or groups

Using the Keywords toolbar:



Expand/collapse the list

Add a new keyword

Add a secondary keyword

Rename a keyword

Remove a keyword

Using the Groups toolbar:

Add a new group

Rename a group

Remove a group

See Creating an Index or Groups (on page 231) for more information about adding/linking to keywords and groups.

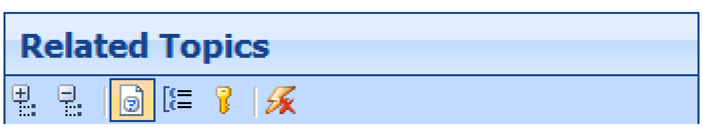
Related Topics pane

Use the **Related Topics pane** to:

- View/Edit your related topics

You can view related topics three ways: by the hierarchy (TOC) structure, by keyword (index) entry, or by group.

Using the **Related Topics pane toolbar** you can:



- View Hierarchy (TOC) structure and related topics
- View Groups and related topics
- View Keywords and related topics
- Disable subtopic relations
- Create custom related topics

Right-click on a topic in the Hierarchy to:

- Change the sort to “by Document” or “Alphabetical”
- Disable subtopic relations

Clicking on a topic, group, or keyword in the upper half of the **Related Topics** pane will display related topics in the lower half.

See Managing Related Topics (on page 225) for more information about relating topics.

Windows

Doc-To-Help has several windows that will be displayed based on the task you are working on. Navigate through them by clicking the tabs.

See Navigating Doc-To-Help (on page 95) for information on other navigation methods.

Start Page window

From the **Start Page** window you can:

- Create a new project
- Open an existing project
- Convert a project from several different formats (HTML Help, WinHelp, RoboHelp, Doc-To-Help 2000)
- Select Team Authoring actions
- Open Doc-To-Help Help
- Access how-to videos and several useful documents about working with Doc-To-Help

See Creating and Converting Projects (on page 139) for more information.

Topics window

The **Topics** window displays all of the Topics in your project.

From this window, you can select topics that can be dragged and dropped into the Contents (on page 109), Index and Groups (on page 110), or Related Topics (on page 111) panes. You can also drag topics into source documents to create links, see Drag-and-Drop Linking on page 214 for more information.

You can also select topics and apply Platform, Target, or Attribute conditions to them. See Condition ribbon group (on page 105) for details.

To specify a topic as the default for the project (the first displayed in the Help window), right-click on it and choose **Default Topic**.

To display/edit the Properties for a topic, right-click on it and choose **Properties.**

Right-click in the **Topics** window to:

- Specify the columns to display (choose **Columns**) Columns available: Document, Title, ASCII Name, Style, Type, Context ID, Keywords, Groups, Link Tags, Context String, Order, TOC)
- Return to the default column display (choose **Reset Column Layout**)
- Show/hide deleted topics (select/clear **Show Deleted Topics**)
- Purge deleted topics (choose **Purge Deleted Topics**)

- Show/hide the filter row (select/clear **Show Filter Row**)
- Clear the filter (choose **Clear Filter**)

Columns may be rearranged by selecting the column header and dragging to the desired position. Click on a column header to sort by that column.

Watch the video: Quick Tour of the Topics Window (1:32) [http://helpcentral.componentone.com/videos/D2H2009_Tour/QuickTourTopicswindow.html]

Output window

The **Output** window will appear when a target is built. It displays the status of the build as it progresses.

The window will split after the first build is completed. One window will be the **Build Log**; the other will display the **Error(s) and Unresolved Link(s)** (if any). These windows can be accessed by clicking the tabs at the bottom of the **Output** window. (To reopen, click the **Build Log** button on the **Home** tab.)

The **Errors and Unresolved Links** window will display detailed information so that you can troubleshoot the issues found during the build. Right-click in the window for options to change the column display or sort.

XHTML Editor window

Doc-To-Help's XHTML Editor conforms to the W3C XHTML 1.0 strict specification. Doc-To-Help's built-in XHTML editor makes it easy to edit and validate your .xml documents.

The **XHTML Editor** opens when you open or create an XML document in your project. It cannot be opened independently. When it opens, three additional tabs will appear — Editor (on page 99), Table (on page 101), and D2HML (on page 103) — that work with the **XHTML Editor**. See Editing XHTML Documents on page 119 for more information.

To open an XML document

To open a project document in the XHTML editor, open the **Documents** pane and double-click on the document name.

To add a new XHTML document to your project, see Adding a Document to a Project on page 116.

To convert an HTML or Microsoft Word document to XHTML, see Converting Existing Documents to XHTML on page 117.

XHTML Editor modes

Click on the buttons at the bottom of the XHTML Editor window to work in the available modes. You can edit in every mode but Preview.

- **Design** — The default WYSIWYG editing mode.
- **Design with D2HML Markers** — In this view, D2HML objects (links, expanding text, etc.) are enclosed in blue brackets. When you hover over the icon within the brackets, the D2HML Style and Text will be displayed along with other details. If there are no D2HML markers in the document, this view will not display.
- **Source** — Displays the XHTML source. This mode includes **Validate** and **Fix** buttons to validate the XHTML document.

- **Preview** — Displays the document as it will look when the target is built; variables will be added in, D2HML hotspots will be replaced with links, conditional text will be removed that does not apply to the current target, etc. The links will not work, but if you hover over them the destination will display in a popup.

Variables window

The **Variables** window is used to create and display all the **Text** and **Rich Content Variables** available in your project.

Variables allow you to manage content in one place for reuse across your project. See Creating Variables (on page 194) for instructions on the uses of variables, as well creating and inserting variables in your project.

To open the Variables window

From the Project tab, Project ribbon group (on page 106), choose the **Variables** button.

Team Authoring window

The **Team Authoring** window only appears when you open a Team Project. It displays all of the document names in the repository, as well as their status, size, modification date, version number, and the date they were retrieved from the repository. You can perform team authoring actions on individual files using the Team Authoring tab (on page 107).

See Working with a Team Project (on page 246) for more information.

To modify the column display, right-click in the window and choose **Columns**.

Dynamic Help

The **Dynamic Help** window displays relevant Help automatically as you use your mouse to click or hover over the Doc-To-Help interface. For the complete Help file, click on the Help icon located on the upper right .

The **Dynamic Help toolbar** allows you to return to topics you've already visited (using the **Browse** buttons), open the Help file in a browser window (with the option to go directly to the **Search**, **Index**, or **Table of Contents**) and to lock the current Help topic as you work through a task (using the **Pin** button).

If you need quick Help on a specific toolbar button, hover over it to display its Super Tooltip (on page 97) — both the name of the button and an overview of its use will appear.

If you are creating documentation for a software product and would like your interface to include Dynamic Help, see Implementing Context Sensitive Help on page 200 to learn more. The .NET control used to create Super ToolTips may also be purchased from ComponentOne.

To close/open the Dynamic Help window

Close the window by clicking the close icon in the upper right corner of the window. Reopen the **Dynamic Help** window by clicking the Dynamic Help icon on the upper right .

Working with Source Documents

You have three source document options in Doc-To-Help. You can choose to use one, two, or all three in your project if you wish:

Source Document	File Type(s)	Editor	Ribbons/Toolbars
XHTML	.xml	Doc-To-Help XHTML editor	**Editor**, **Insert**, **Table** (in Doc-To-Help)
Microsoft® Word	.doc .docx (Word 2007)	Microsoft Word	**Doc-To-Help** (added when Doc-To-Help is installed) (Depending on the template, **Doc-To-Help Special Formatting** may be added as well.)
HTML	.html, .htm	Microsoft FrontPage®, Adobe® Dreamweaver®, your choice	**ComponentOne Doc-To-Help D2HML Styles** (added when Doc-To-Help is installed; Microsoft FrontPage and Adobe Dreamweaver only)

You edit .xml documents using the XHTML Editor window (on page 113) in Doc-To-Help, but Microsoft Word (.doc and .docx) and HTML documents (.htm or .html) are edited in those environments. This makes it possible for you to work in the editor you prefer while still creating the output you need.

Word documents are edited in Microsoft Word using the **Doc-To-Help** or **Doc-To-Help Special Formatting** toolbars or ribbons. See Editing Word Documents (on page 128) for more information

HTML documents are edited using the **ComponentOne Doc-To-Help D2HML Styles toolbar** or menu (added when Doc-To-Help is installed; Microsoft FrontPage and Adobe Dreamweaver only). HTML documents may also be edited in the external HTML editor of your choice. See Editing HTML Documents (on page 135) for more information.

Three ribbons (**Editor**, **Insert**, and **Table**) in Doc-To-Help are used to edit your XHTML documents in the **XHTML Editor** window. See Editing XHTML Documents (on page 119) for more information. Doc-To-Help includes a converter you can use to convert existing documents to XHTML. The converted documents are edited in the XHTML Editor window (on page 113). For more information on converting documents to XHTML, see Converting Existing Documents to XHTML on page 117.

D2HML styles are used to create links, inline text, apply conditional text, insert variables, etc. in all three types of documents. See Using D2HML (Doc-To-Help Markup Language) (on page 210) for more information.

Adding a Document to a Project

You may add three different types of new or existing documents to your project — Microsoft® Word, HTML, or XHTML.

After adding a new document to your project, you should build the Target (see Building a Target on page 235). This processes the document and adds its topics to the project.

To add a new document to your project

1. Open the Documents pane (on page 108).
2. If you are adding a new HTML or XHTML document, select the document that you would like the new document to be a sibling or child of. (You can always move it later by dragging it to the desired location.) (A sibling is a document at the same level; a child is one level lower.)
3. Click the **Create New button** 

.
4. Choose **XHTML Document**, **HTML Document**, or **Word Document**. The **Save New Document As** dialog box will open. Enter a name and click **Save**.
5. If you chose **Word**, the document will open and be added to the Documents pane. If you chose **XHTML** or **HTML**, the **New Document** dialog box opens.
6. Choose the **Multiple topics in the document** or **Single topic in the document** radio button. The default is **Single Topic**. Single Topic documents are not separated by heading levels; Multiple Topic documents are. (If you later add additional heading levels to a Single Topic document, they will not be separated, but appear in the project as one continuous topic.)

   If **Single Topic** was selected, the document can be either a **Child** or a **Sibling** of the document you originally selected. A child document will appear one level below the document in the project structure; a sibling will appear at the same level.
7. If you wish, change the **Title** you already specified.
8. Choose the **Style** for the topic. Heading1 is the default. (Unless the document is the child of another, then the appropriate style will be displayed.)

   The default CSS for the project will be used for HTML and XHTML documents. The **Default CSS** is specified in the Project Settings dialog box. Word documents added will use the **Default Template** specified in the **Project Settings** dialog box.
9. Click **OK**. The file will appear in the **Documents** pane.

To add an existing document to your project or import existing Word or HTML documents into your project as XHTML

1. Open the Documents pane (on page 108).
2. Click the **Add Existing Documents** drop-down. 

 The **Document Import Wizard** will open.
3. Choose the type of document(s) you would like to import — **XHTML Document**, **HTML Document**, or **Word Document**.
4. Choose your import option. If you choose Word or HTML, you will have the option to convert them to XHTML while they are being imported. (XHTML documents can be edited in Doc-To-Help's XHTML editor.)

5. In the **Select Files or Folders to Import** screen, click the **Add File(s)** or **Add Folder** button to select your documents. By default, these documents will be copied to the appropriate folder of your project (by default, **Documents**). Your original documents will remain in their original location, with a **.backup** file extension. Choosing **Leave documents at their original location** is not recommended, because references to other documents in your project may be broken in your targets.

   If you are converting Word documents to XHTML, the **Word Style Map** that will be used for the conversion will be displayed; click the **Browse** button to change it.

   Note: If you attempt to add a document to your project that has the same name as one that already exists in your source document folder, it will display in the Wizard with a red X icon next to it. Since you cannot import a document with the same name, the **Import** button will be disabled.

6. Click the **Import** button to begin the import. The imported files will appear in the **Documents pane**. Use the **Documents pane toolbar** to rearrange the files as you wish.

Note: You can drag-and-drop an existing document from Windows Explorer into the **Documents** pane, but you should copy it into the appropriate project folder first. By default, this folder is named **Documents**. These default names can be changed using the **Project Settings** dialog box. See Setting Project Properties (on page 196) for more details.

When an HTML document is added to a project, Doc-To-Help will use the document's <title> tag as the **Title** field. The **Title** field can be viewed in the **Topics window** and the **Topic Properties** dialog box.

To rename a document in your project

In the Documents pane, right-click on the document and choose **Rename Document**. Build the Target.

To remove a document from your project

In the Documents pane, right-click on the document and choose **Remove Document**. Build the Target.

To designate a document as the project Glossary

In the Documents pane, right-click on the document and choose **Glossary**. Build the Target. For more information on glossaries, see Creating a Glossary on page 138.

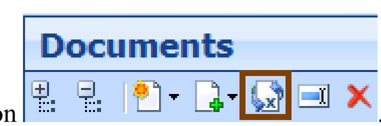
To convert an existing Word or HTML document to XHTML, see Converting Existing Documents to XHTML on page 117.

Converting Existing Documents to XHTML

You can convert a Microsoft® Word or HTML document that is already part of your project to XHTML, or you can import a Microsoft Word or HTML document into your project as XHTML.

To convert existing project documents to XHTML

Doc-To-Help makes it easy to convert project source documents to valid .xml files.

1. Open the Documents pane (on page 108).
2. Click the **Convert Multiple Documents to XHTML** button 

. The **Convert Documents to XHTML** dialog box will open.
3. By default, the converted documents will be saved to their original location. If you would like to save them to another location, choose **Save converted files to** and click the **Browse** button (by default documents are saved to the **Documents** folder). If the documents are Word files, confirm the **Word Style Map** (the Word to XHTML styles map file). Click the **Browse** button to change it.

4. If your original document included local formatting (formatting that was applied without using a style) and you would like to retain it, select the **Keep local formatting** check box.

5. The **Keep backup copy** check box is selected by default. If you want the option to revert to the original source documents, leave it selected. The files will be saved with a **.backup** file extension.

   If you would like to convert one or more files individually, select the document name(s) on the left, then select the **Use individual conversion settings** check box. Choose the desired options.

6. Click the **Convert** button.

7. A message box will appear, reminding you that all local formatting (formatting applied without using a style) will be lost in the conversion. Click **Yes** if that is acceptable.

8. The file will be converted and appear in the **Documents** pane.

If you would like to convert a single document, right-click on it in the **Documents** pane and choose **Convert to XHTML**. The **Convert Documents to XHTML** dialog box will open, with the single document selected in the list.

Watch the video: How to Convert Documents to XHTML (1:39) [http://helpcentral.componentone.com/videos/D2H2009_Tour/ConvertingtoXHTML.html]

Importing an existing Word or HTML document into your project as XHTML

1. Open the Documents pane (on page 108).

2. Click the **Add Existing Documents** drop-down 

. The **Document Import Wizard** will open.

3. Choose the type of document(s) you would like to import and convert — **HTML** or **Word**.

4. In the **Choose an Import Option** screen, choose **Convert to XHTML**.

5. In the **Select Files or Folders to Import** screen, click the **Add File(s)** or **Add Folder** button to select your documents. By default, these documents will be copied to the appropriate folder of your project (by default, **Documents**). Your original documents will remain in their original location, with a **.backup** file extension. Choosing **Leave documents at their original location** is not recommended, because references to other documents in your project may be broken in your targets.

   If you are converting Word documents to XHTML, the **Word Style Map** that will be used for the conversion will be displayed; click the **Browse** button to change it.

6. Click the **Import** button to begin the import. The imported files will appear in the **Documents pane**. Use the **Documents pane toolbar** to rearrange the files as you wish.

7. A message box will appear, reminding you that all local formatting (formatting applied without using a style) will be lost in the conversion. Click **Yes** if that is acceptable.

8. The file will be converted and appear in the **Documents** pane.

See the Editor (on page 99), Table (on page 101), and Insert (on page 103) tabs, as well as Editing XHTML Documents (on page 119) for information on editing XHTML.

See Working with Source Documents (on page 115) for more on the types of documents you can work with in Doc-To-Help.

To add a new document to your project, see Adding a Document to a Project on page 116.

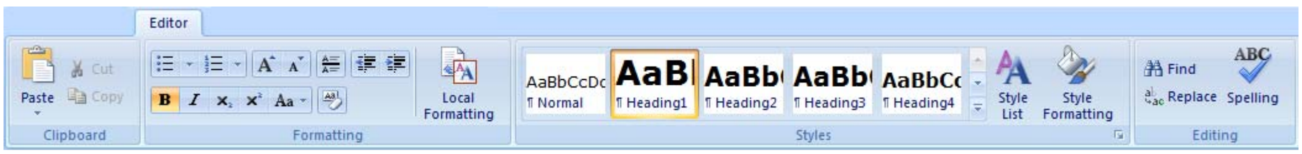
Editing XHTML Documents

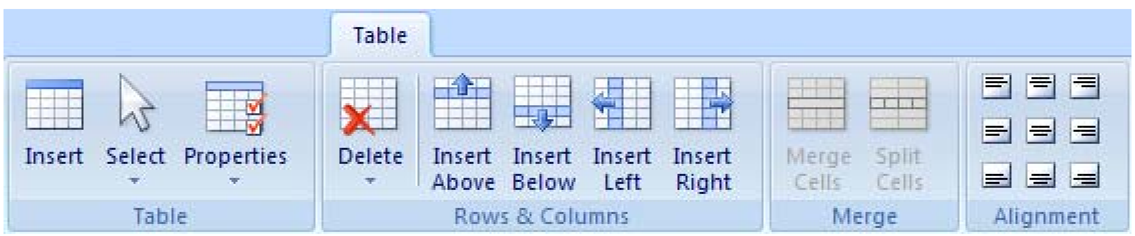
XHTML documents are edited in the XHTML Editor window (on page 113).

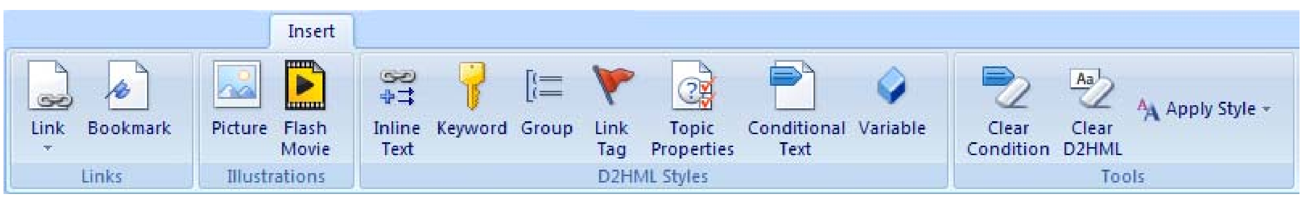
The **XHTML** Editor opens when you open an .xml document from the Documents pane (on page 108).

The Editor (on page 99), Table (on page 101), and Insert (on page 103) ribbons are used to apply styles, insert tables/images/movies, and create links in .xml documents. These tabs appear only when the **XTHML** Editor is open.

Doc-To-Help's **XHTML** Editor and accompanying ribbons make it easy to create and edit valid .xml documents that conform to the W3C XHTML 1.0 Strict specification.







Applying Styles in the XHTML Editor

You can apply styles and formatting in the XHTML Editor window (on page 113) several ways.

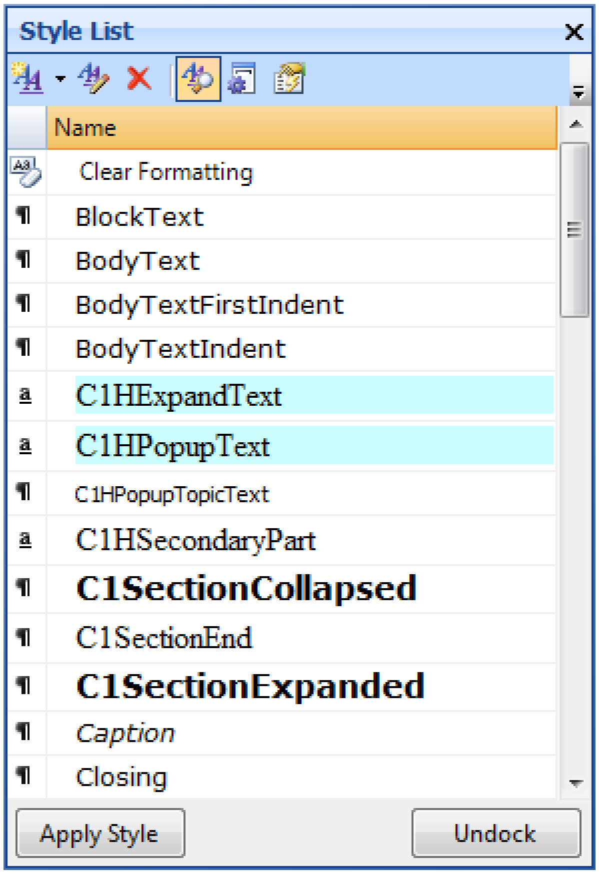
See Editing a CSS (on page 189) for information about editing and changing your project's CSS files.

To apply a style using the Style List

1. Click the **Editor** tab.
2. Select the text in the **XHTML** Editor.
3. Click the **Style List** button in the **Styles** ribbon group. The **Style List** window will open.
4. Choose a style from the **Style List** and click the **Apply Style** button at the bottom of the window.

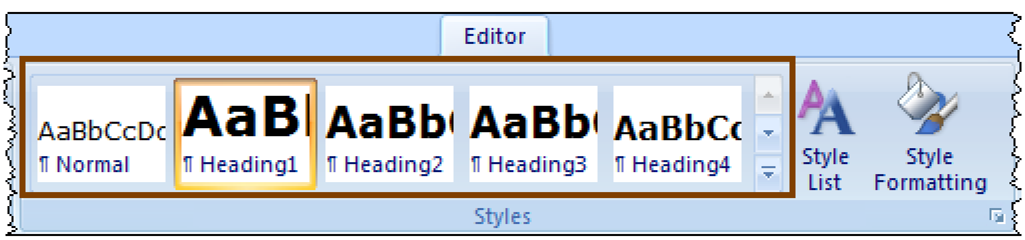
   Paragraph Styles are noted with a paragraph symbol ¶, while Character Styles are indicated with a character symbol . It is best to apply default Character Styles using the **D2HML** ribbon. See **To apply a D2HML Style** below.

See Style List Window Tour (on page 192) for more information about the **Style List** window.



To apply a Style using the Style Gallery

1. Click the **Editor** tab.
2. Select the text in the **XHTML** Editor.
3. Click the appropriate style in the **Style Gallery**. The style will be applied.



You can change the list of styles displayed in the **Style Gallery**. See Managing Style Gallery Styles (on page 122) for details.

To apply a D2HML Style

Doc-To-Help Markup Language (D2HML) styles are used to add links, keywords, groups, variables, etc. in your XHTML document. There are pre-defined styles, which can be edited.

To apply a D2HML style, first select the text.

D2HML styles are applied with the Insert tab (on page 103). See Using D2HML (on page 210) for more information on each D2HML style.

Applying Formatting on the fly

On-the-fly formatting is a way to create a style exactly when you need it.

When using this method, you will create a new style if necessary.

1. Click the **Editor** tab.
2. Select the text in the **XHTML** Editor.
3. Click the **Style Formatting** button  in the **Styles** ribbon group. The **Style Formatting** dialog box will open.
4. Change the **Font**, **Background**, **Border**, **Box**, **Paragraph** and/or **Position** via the tabbed windows.
5. Click **OK**. Now a new style needs to be created, so a dialog box opens entitled **Choose how to handle the specified formatting** so that you can do so.
6. There are three options:
   - Create a new style
   - Create a new style that is derived (based on) an existing one. This means the behavior of the new style will be the same as the one it is based on.
   - Use an existing style. This means that your changes will not be applied; you have decided to use an existing style instead.
7. Choose an option using the radio buttons.

   If you choose to create a new style: Click the **Next** button. Enter the **Name** of the style, verify the **Type** (Paragraph or Character Style) and the **Style Sheet** it should be saved to. Click **Finish**. The new style will appear in the **Style List** window, and will be added to both your Source and Target CSSs.

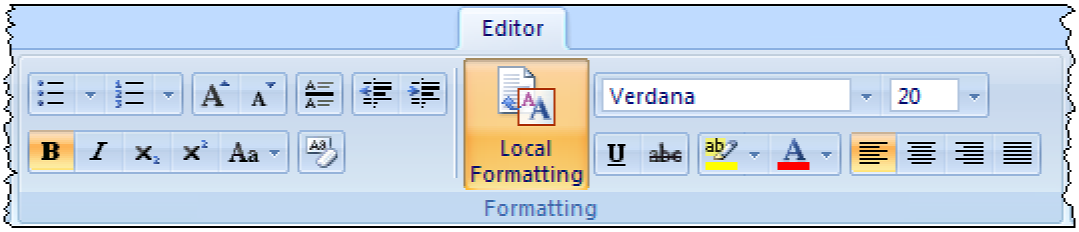
   If you choose to create a new derived style: Choose a style from the list of styles displayed. Note that when you hover over each style, the properties of that style will be displayed. Click the **Next** button. Enter the **Name** of the style, verify the **Type** (Paragraph or Character Style) and the **Style Sheet** it should be saved to. Click **Finish**. The new style will appear in the **Style List** window, and will be added to both your Source and Target CSSs

   If you choose to use an existing style: Select a style from the list. Click **Finish**.

See Editing a CSS (on page 189) for more information on creating styles.

Applying Local Formatting

If you would like to apply formatting without using a style (local formatting), you may do so in the XHTML Editor window (on page 113), using the Formatting ribbon group (on page 100) of the **Editor** tab.



There are many factors to consider before using local formatting. Using local formatting (without styles) makes your document less standards-compliant, harder to maintain, and may introduce inconsistencies. However, if you use the **Style Formatting** button in the **Styles ribbon group** to specify formatting, and then create a style, it will be available for use in the future and won't need to be recreated.

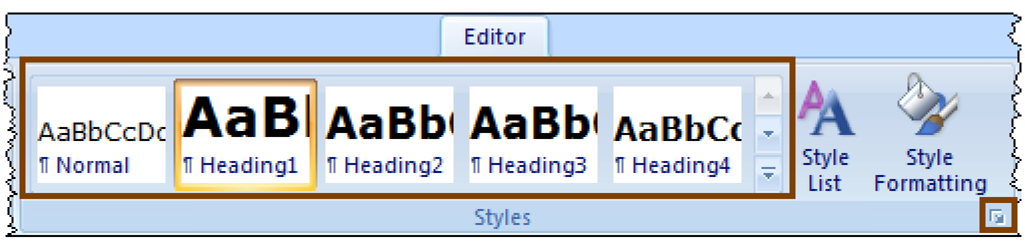
There are two types of buttons in the **Formatting ribbon group**. The ones available to the left of the **Local Formatting** button (bullets, numbering, bold, italic, subscript, superscript, and more) may be applied to your document and the document will still conform to the W3C XHTML 1.0 Strict Specification. This is because those options are part of that specification.

If you click the **Local Formatting** button, additional options for applying formatting without styles will be revealed. These options are: font name, font size, text align left, text align center, text align right, text align justify, highlight color, font color, underline and strikethrough. If you use these options, your document will no longer conform to the W3C XHTML 1.0 Strict specification; it will switch to the Transitional specification.

PLEASE NOTE: If you use **Local Formatting** options (font name, font size, text align left, text align center, text align right, text align justify, highlight color, font color, underline and strikethrough) then turn off Local Formatting by clicking the button, all of the formatting applied with the local formatting options will be removed, returning the document to the Strict specification.

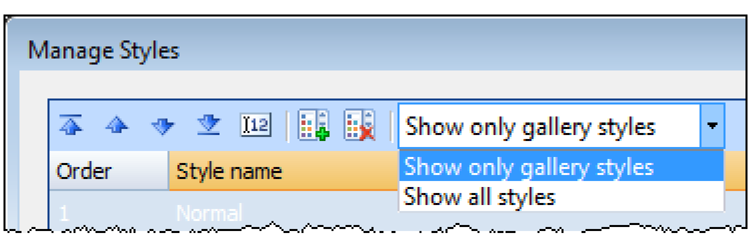
Managing Style Gallery Styles

Using the **Manage Styles** dialog box, you can specify and arrange the styles that appear in the **Style Gallery** of the **Editor** tab.



To open the Manage Styles dialog box

1. Open the **Editor** tab.
2. Click the **Styles** ribbon group dialog box launcher. The **Manage Styles** dialog box will open.



To manage the Style Gallery

If you plan to add additional styles to the **Style Gallery**, choose **Show All Styles** from the drop-down in the dialog box toolbar.

Select a **Style name,** then…

- To make it the first or last style displayed, click the **Make First** or **Make Last** button .
- To move the style up or down in the list, click the **Move Up** or **Move Down** button .
- To assign the order numerically, click the **Assign Value** button .
- To remove the style from the list, click the **Remove** button .
- To add the selected style to the list, click the **Add** button .

Inserting Images in the XHTML Editor

Using the **Picture Properties** dialog box, you may insert .gif, .jpg, .png, .jpeg, .bmp, .wmf, or .emf image files into the XHTML Editor window (on page 113).

To insert an image

1. Open the **Insert** tab.
2. Place your cursor at the point you'd like to insert an image.
3. Click the **Picture** button  in the **Illustrations** ribbon group. The **Picture Properties** dialog box will open.
4. Browse to the **File name**.
5. If desired, enter the **Alternative text** (for screen readers) for this image. For more on accessibility, see Section 508 Compliant NetHelp on page 35.
6. If desired, specify the **Width** and **Height** of the image, as well as a **Border Width**. The image will appear in the **Preview** area.
7. Click **OK**.

Please note: If the image selected was not already stored in your project, Doc-To-Help will prompt you to save it to your project's **Media** folder.

Inserting Flash Movies in the XHTML Editor

You may insert Flash movies into the XHTML Editor window (on page 113). If you would like to create a Flash movie and don't have a software tool, you may want to check out ComponentOne DemoWorks (http://www.componentone.com/Products/DemoWorks/Overview.aspx).

To insert a Flash movie

1. Open the **Insert** tab.
2. Place your cursor at the point you'd like to insert a movie.
3. Click the **Flash Movie** button  in the **Illustrations** ribbon group. The **Movie in Flash Format Properties** dialog box will open.

4. Specify the location of the movie (.swf) using the **Browse** button. The movie can be stored in the project, or you can link to a URL.
5. If desired, enter the **Alternative text** (for screen readers) for this movie. For more on accessibility, see Section 508 Compliant NetHelp on page 35
6. If desired, specify the **Width** and **Height** of the image. Also specify if the movie should **Autoplay** (begin playing when the Help file is displayed) and/or **Loop** (play continuously).
7. Click **OK**.

Please note: If the movie selected was not already stored in your project, Doc-To-Help will prompt you to save it to your project's **Media** folder.

Creating Hyperlinks in the XHTML Editor

Using the **Hyperlink Properties** dialog box, you may create a hyperlink to a URL or another file from the XHTML Editor window (on page 113). You may also add a screen tip (popup) that will display when a user hovers over the hyperlink and other advanced features.

To create a hyperlink

1. Open the **Insert** tab.
2. Select the text you would like to create a hyperlink from.
3. Click the **Link** button  drop-down in the **Links** ribbon group. Choose **External**. The **Hyperlink Properties** dialog box will open.
4. Browse to the **Address** of the URL or file.
5. If you would like a tooltip to appear when the user hovers over the hyperlink, enter it in the **Screen tip** field.
6. Choose the **Target Frame** the hyperlink should open in. Options are: **Default**, **Same frame, Whole page**, **New window**, **Parent frame**.
7. An **Access Key** (a single character that will open the hyperlink if it has focus) may be entered if desired. Also, a **Tab index** number may be specified. The tab index defines the order in which this hyperlink will receive focus when the user navigates the page by tabbing.
8. Click **OK**.

Please note: Choose **In Project** from the **Link** button drop-down to create topic links within the Help project. For more information, see Creating Links on page 212.

Creating Bookmarks in the XHTML Editor

Using the **Bookmark Properties** dialog box, you may insert or delete a bookmark in your document in the XHTML Editor window (on page 113). Bookmarks are a great way to create links within your project to specific sections; sending Help users to the exact paragraph of a topic, rather than the main topic it resides in.

To create a link to a bookmark, see Creating Links on page 212.

To insert a new bookmark

1. Open the **Insert** tab.
2. Select the text you would like to bookmark.

3. Click the **Bookmark** button  in the **Links** ribbon group. The **Bookmark Properties** dialog box will open.
4. Enter a **Name** for the bookmark.
5. Click **OK**. The bookmark will appear in the list.

To navigate to the existing bookmarks in the document, click on the bookmark name, then the **Go to Bookmark** button.

To delete a bookmark

1. Open the **Insert** tab.
2. Click the **Bookmark** button in the **Links** ribbon group. The **Bookmark Properties** dialog box will open.
3. Choose the bookmark from the list. It will then appear in the **Name** field.
4. Click the **Remove Bookmark** button.
5. Click **OK**. The bookmark will be removed from the list.

Editing and Proofing in the XHTML Editor

The **Editor** tab includes several different functions to edit your document in the XHTML Editor window (on page 113).

The Clipboard ribbon group (on page 100) can be used to cut/copy/paste. The Editing ribbon group (on page 101) includes **Find** and **Replace** functions, as well as a spell checker. You can find/replace within a single document, or across the entire project.

Inserting Tables in the XHTML Editor

A table may be configured and inserted in the XHTML Editor window (on page 113) using the **Table** tab.

To make changes to a table (insert/delete rows or cells, split the table, align text, etc.) see Viewing and Modifying Table Cells, Rows, and Columns on page 125.

To insert a table

1. Open the **Table** tab.
2. Place your cursor at the point you'd like to insert a table.
3. Click the **Insert** button  in the **Table** ribbon group. The **Table Properties** dialog box will open.
4. Specify the number of columns and rows.
5. If desired, enter the **Caption** and **Summary** text (for screen readers). For more on accessibility, see Section 508 Compliant NetHelp. on page 35
6. If desired, specify the cell height and width, spacing and padding, table alignment (left, center, right), and border width.
7. Click **OK**.

Viewing and Modifying Table Cells, Rows, and Columns in the XHTML Editor

A table may be modified in the XHTML Editor window (on page 113) using the **Table** tab.

To insert a table see Inserting Tables on page 125.

To view/edit cell, column, row, or table properties

1. Open the **Table** tab.
2. Place your cursor in the table you wish to edit.
3. Click the **Select** button  drop-down list.
4. Choose **Select Cell**, **Select Row**, or **Select Table**.
5. Click the **Properties** button drop-down list.
6. Make a selection: **Cell Properties**, **Column Properties**, **Row Properties**, or **Table Properties**.

   The appropriate dialog box will open: **Cell Properties**, **Column Properties**, **Row Properties**, or **Table Properties**.
7. The options will vary by dialog box. Make desired changes and click **OK**.

To add rows and columns to a table

1. Open the **Table** tab.
2. Place your cursor in the table at the point you'd like to make an addition.
3. If you'd like to insert a row, click the **Insert Above** or **Insert Below** buttons. To insert a column, click **Insert Left** or **Insert Right**.

To delete a row, column, or table

1. Open the **Table** tab.
2. Place your cursor in the table you wish to edit.
3. Click the **Select** button drop-down list.
4. Choose **Select Cell**, **Select Row**, or **Select Table**.
5. Click the **Delete** button drop-down list.
6. Make a selection: **Delete Columns**, **Delete Rows**, or **Delete Table**.
7. Click **OK**.

To merge or split cells

1. Open the **Table** tab.
2. Place your cursor in the table you wish to edit.
3. Click the **Select** button drop-down list.
4. Choose **Select Cell**, **Select Row**, or **Select Table**.
5. Choose the **Merge Cells** or **Split Cells** button from the Merge ribbon group (on page 102). If the option is not available the button will be grayed out.

To change text alignment

1. Open the **Table** tab.
2. Place your cursor in the table you wish to edit.
3. Click the **Select** button drop-down list.
4. Choose **Select Cell**, **Select Row**, or **Select Table**.
5. Choose the desired text alignment from the Alignment ribbon group (on page 102).

Saving XHTML Documents

There are several options for saving XHTML documents. These options are available only when an XHTML document is open in the **XHTML Editor** window.

To save the document you are currently editing: Doc-To-Help button  > Save.

To save all open documents: Doc-To-Help button > Save All. (Or use the button on the Quick Access Toolbar on page 96).

To save a document with another name: Doc-To-Help button > Save As. A document created this way is saved in the **Documents** project folder, but is not added to the project. You can use the **Add Existing Documents** button in the **Documents** pane if you would like to add it to the project.

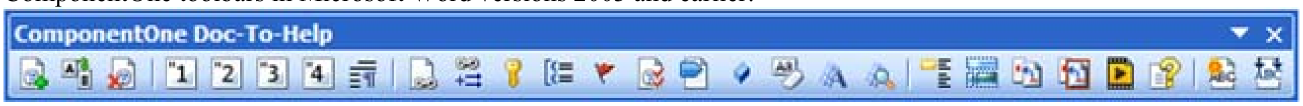
Open Microsoft® Word documents and HTML documents must be saved in those applications.

Editing Word Documents

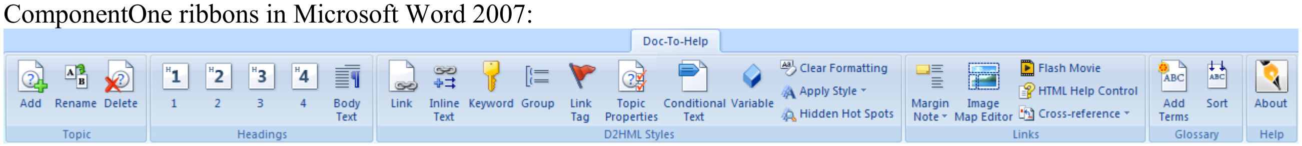
Styles control the look and behavior of your final Targets. Doc-To-Help adds two toolbars or ribbons that are used to apply styles and D2HML Styles, as well as perform other useful functions, such as creating image maps and inserting Flash movies.

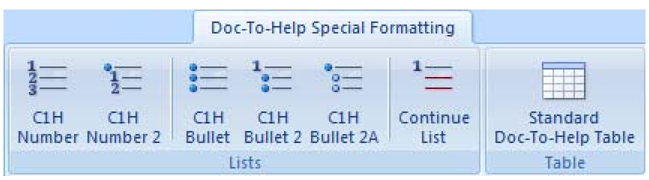
These toolbars or ribbons are: **ComponentOne Doc-To-Help** and **Doc-To-Help Special Formatting**.

ComponentOne toolbars in Microsoft Word versions 2003 and earlier:

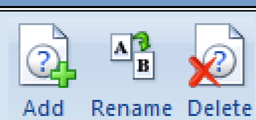
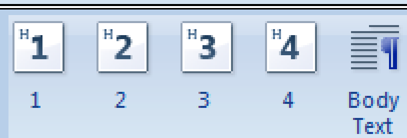
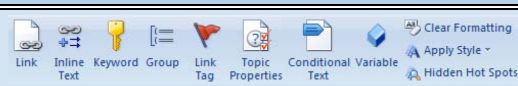
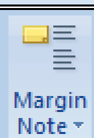


ComponentOne ribbons in Microsoft Word 2007:





Note: The **Doc-To-Help Special Formatting** toolbar or ribbon will not be displayed in all templates. In those cases, use the Bullet, Numbering, and Table buttons in Microsoft® Word.

Toolbar buttons	Function	Notes
	Add, Rename, or Delete Topics	Use to rename or delete an existing topic in a document, or add a new topic. See Adding, Renaming, and Deleting Topics on page 130.
	Heading 1, 2, 3, 4, Body Text Style	Apply the Heading 1, 2, 3, 4, or Body Text style to the selected text in the Source document.
	D2HML Styles	Apply or clear a D2HML style. See Using D2HML on page 210.
	Margin Note	Use to insert a note or graphic in the left-hand margin of a printed target. The margin note can be displayed as a popup in Help Targets if desired. See Creating Margin Notes on page 131.

Toolbar buttons	Function	Notes
Image Map Editor	Image Map Editor	In Help Targets, use the Image Map Editor to create hyperlinks within a graphic. See Using the Image Map Editor on page 132.
Flash Movie	Insert Flash Movie	You may insert Flash movies (.swf) into Help Targets — see Inserting Flash Movies on page 133.
HTML Help Control	Insert HTML Help ActiveX Control	Use to insert HTML Help ActiveX Controls into your Word documents. See Inserting an HTML Help ActiveX Control on page 133.
Cross-reference	Insert Cross Reference/ Complete Cross Reference	Use to insert cross-references that will appear as page numbers in printed targets and hyperlinks in Help Targets. See Inserting Cross References on page 132.
Add Terms Sort	Add/Sort Glossary Terms	Use to add terms to your glossary from any Source document. See Adding Terms to the Glossary on page 133.
C1H Number C1H Number 2 C1H Bullet C1H Bullet 2 C1H Bullet 2A Continue List	C1H Number, C1H Number 2, C1H Bullet, C1H Bullet 2, C1H Bullet 2A, Continue List	Apply the List style to the selected text in the Source document.
Standard Doc-To-Help Table	Standard Doc-To-Help Table	Use to insert a Table in your Doc-To-Help project. See Inserting a Standard Doc-To-Help Table on page 134.

These toolbars reside in individual Doc-To-Help templates (such as C1H_NORM.dot), so you should create your files using a predefined Doc-To-Help template and edit it to your specifications. See Guide to Templates and Styles on page 5 for more information.

For more on applying styles, see Applying Styles in Microsoft Word on page 129.

Applying Styles in Microsoft Word

There are two ways to apply a style in Microsoft Word.

To apply a style using the Doc-To-Help toolbars

1. Select the text.
2. Using the **Doc-To-Help** and **Doc-To-Help Special Formatting** toolbars, click the toolbar buttons to apply **Heading 1, 2, 3, 4**, and **Body Text** styles, as well as bulleted and numbered lists.

   Please note that the **Doc-To-Help Special Formatting** toolbar will not be displayed in all templates. In those cases, use the Bullet, Numbering, and Table buttons in Microsoft® Word.

D2HML Styles (links, popups, conditional text, variables, etc.) are also applied with the **Doc-To-Help** toolbar. See Using D2HML (on page 210) for more information.

To apply a style using the Microsoft® Word Styles window

1. Select the text.
2. Use the appropriate window for your version of Word.

In Word 2007, the **Styles** window is used to apply Styles.

- Click the **Home ribbon** > **Styles ribbon group** dialog box launcher. The **Styles** window will open.
- Click the name of the style you wish to apply.

In Word 2003 and earlier:

- From the **Format** menu, choose **Styles and Formatting**. The **Styles and Formatting** window will open.
- Click the name of the style you wish to apply.

To clear a style from your text, select the text and choose **Clear All** or **Clear Formatting** from the list of styles.

D2HML Styles (links, popups, conditional text, variables, etc.) can also be applied this way if you wish. See Using D2HML (on page 210) for the list of Word Styles that correspond to the D2HML toolbar buttons.

See Editing a Template (on page 8) for more information on editing styles in Word.

Adding, Renaming, and Deleting Topics

Topics may be added, renamed, and deleted from a Word document using the **Doc-To-Help** toolbar or ribbon. Adding a topic using this method makes it possible to perform certain functions – for example, creating a link to the new topic – without building the Target first. The topic that is added, renamed, or deleted will automatically be updated in the table of contents (unless the Table of Contents is customized).

You can also add a topic by entering the text and applying a Heading style. Existing topics can also be renamed or deleted by editing or deleting the topic text. When using these methods make sure to build the Target to incorporate your changes.

To add a new topic

1. Open your source document (.doc or .docx) in Word.
2. Place your cursor at the point you'd like to insert the new topic. This should be at the start of a new paragraph. (Place an empty paragraph at the end of the previous topic.)
3. Click the **Add** button . You will be asked to confirm the new topic location. Click **Yes**. The **Add Topic** dialog box will open.
4. Enter the **Title** of the topic and choose the **Style** from the drop-down list. The location of the new topic will display in the **Topic hierarchy**.
5. Click **OK**.

To rename a topic

1. Open your source document (.doc or .docx) in Word.
2. Select the entire name of the topic you wish to rename.

3. Click the **Rename** button . The **Rename Topic** dialog box will open. The name and location of the topic will be bolded in the **Topic hierarchy**.

4. Change the **Title** of the topic and, if you wish, choose a different **Style** from the drop-down list. The **URL**, **ASCII Name**, **Link tag**, and **Keyword** will be changed by default (see Topic Properties dialog box on page 225). If you would prefer any or all of these properties do not change, clear the check box(es).

5. Click **OK**.

To delete a topic or topics

1. Open your source document (.doc or .docx) in Word.
2. Select the entire text of the topic(s) you wish to delete.
3. Click the **Delete** button . The **Delete Topics** dialog box will open. The name and location of the topic(s) will be bolded in the **Topic hierarchy**.
4. Click **OK**.

Creating Margin Notes

Margin Notes are used to place text or graphics in the left margin of a manual, next to the main body of the text. Margin notes do not appear in Help Targets unless you explicitly link them to the text, where they will appear as popups. Margin Notes are created, deleted, and linked using the **Margin Note** button in the **Doc-To-Help** toolbar or ribbon.

To create a margin note

1. Open your source document (.doc or .docx) in Word.

2. Place your cursor at the point you'd like to insert the margin note. Click the **Margin Note** button and select **Create** from the drop-down list.

   A two-column table will be placed in the document with the text of the paragraph in the right hand table cell. If Table Gridlines are turned "on" in Word, you'll see a dotted outline of the table. The gridlines do not print.

3. Type the margin note (or insert an image) in the left hand column of the table. The paragraph text will be in the right hand column.

   If you would like to add additional margin note for the same paragraph, click in the left hand column, click the **Margin Note** button and choose **Add Definition** from the drop-down list.

To create a hotlink to a margin note (Help Targets only)

1. Highlight the hotspot text in the right hand column.

2. Click the **Margin Note** button and select **Set Link** from the drop-down list.

3. A dialog will confirm that you want to link the selected text with the contents of the highlighted cell (the Margin Note). If yes, click **Set Link**. If not, use the **Previous** and **Next** buttons to navigate to the proper cell.

   The hotspot will be created. The margin note will appear as a popup in Help Targets.

The hotspot will be indicated in the Word document with a Word comment that notes the link information. If you would like the see this relationship more graphically, click in the right hand column, click the **Margin Note** button, and choose **Highlight Definition Links** from the drop-down list.

Using the Image Map Editor

For Help Targets, you may create hotspots within a graphic to topics, keywords, or groups using the **Image Map Editor**. This button is available on the **Doc-To-Help** toolbar or ribbon in Word.

To create an image map

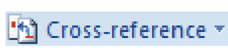
1. Open your source document (.doc or .docx) in Word.
2. Insert a graphic (**Insert** > **Picture** in Word).
3. Select the graphic. Click the **Image Map Editor** button . The **Image Map Editor** dialog box will open.
4. Click the **Draw** button at the top left and select a region.
5. Click the **Link** tab at the bottom right and click the **Link** button. The **Link** dialog box will open.
6. Choose the Topic, Keyword, or Group to link the region to.
7. Click **OK**. Click the **Save** button at the top left. Select additional regions if desired and create links. Each region created will be named “Hot Region #X” by default, which will display in a pop-up when the user hovers over the graphic. To rename, select the Region name and click the **Rename** button on the top right. Rename the region.
8. Click **OK**.

Inserting Cross References

By using the **Insert Cross Reference** and **Complete Cross Reference** buttons on the **Doc-To-Help** toolbar or ribbon, you can automatically add updateable page references in printed manuals and hyperlink jumps in online Help. The page numbers will not appear in Help Targets.

You can also specify that all hyperlinks in your manual output remain active when you create a PDF. See Live Links for more information.

To insert and complete a cross-reference

1. Open your source document (.doc or .docx) in Word.
2. Place your cursor where you want the cross reference to appear.
3. Type the introductory text, such as "For more information, see" or "See also" followed by a space.
4. Click the **Insert Cross Reference** button 

in the Doc-To-Help toolbar or ribbon (in Word 2007, this will be a drop-down from the **Cross-Reference** button). The **Cross Reference** dialog box will open.
5. Set the **Reference Type** to *Heading*.
6. Set **Insert Reference To** to *Heading Text*.
7. Select the heading you want to refer the reader to from the displayed list.
8. Click the **Insert** button. The heading text is inserted.
9. Click the **Close** button.
10. With the insertion point immediately following the cross-reference (reference field), click the **Complete Cross Reference** button (in Word 2007, this will be a drop-down from the **Cross-Reference** button).

    The page reference is inserted and the heading text is enclosed in quotes.

The **On Page** text is set to "on page" by default. To change it, use the [Project Settings dialog box](#) (on page 196).

11. Type a period after the page number, if necessary.

Inserting Flash Movies in Word

It is easy to insert Flash movies into a Word document using the **Doc-To-Help** toolbar or ribbon. If you would like to create a Flash movie and don't have a software tool, you may want to check out [ComponentOne DemoWorks](#).

To insert a Flash movie

1. Open your source document (.doc or .docx) in Word.
2. Place your cursor at the point you'd like to insert a movie.
3. Click the **Flash Movie** button 

. The **Movie in Flash Format Properties** dialog box will open.
4. Navigate to the movie (.swf) location using the **Browse** button. You can point to either a file in a folder location or a URL on the internet or an intranet.
5. If desired, specify the **Width** and **Height** of the image. Also specify if the movie should **Autoplay** (begin playing when the Help file is displayed) and/or **Loop** (play continuously).
6. Click **OK**.

Inserting an HTML Help ActiveX Control

Using the **HTML Help Control** button in the **Doc-To-Help** toolbar or ribbon, you can quickly insert HTML Help ActiveX Controls into your Word documents. These controls can be used to provide features in compiled HTML Help systems (.chm files). Please see the MSDN article "HTML Help ActiveX Control Overview" (http://msdn2.microsoft.com/en-us/library/ms644670.aspx) for more information on HTML Help ActiveX controls.

Note: This is an advanced feature and many of the options available can be performed using other Doc-To-Help functionality.

To insert an HTML Help ActiveX control

1. Open your source document (.doc or .docx) in Word.
2. Place your cursor at the point you'd like to insert an HTML Help ActiveX Control.
3. Click the **HTML Help Control** button 

. The **HHCTRL: HTML Help ActiveX Control Commands** wizard will open.
4. Select the command from the drop-down menu and follow the instructions provided by the Wizard.

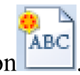
   When you have completed the wizard, Doc-To-Help will insert this code into your document. However, it will be marked with the HTML Passthrough code condition, so the code will appear in the HTML Help target as a control, and will be skipped in other targets. See [Marking Text as Conditional](#) (on page 220) for more information on conditional text.

Adding Terms to the Glossary

If your project includes a glossary, you can quickly insert a glossary term while working in any Word source document using the **Doc-To-Help** toolbar or ribbon.

The glossary document must be a Word document to use this feature.

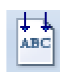
To add a term to the Glossary

1. Open any source document (.doc or .docx) in Word.
2. Select the word you would like to add to the Glossary and click the **Add Terms** button . The **Add Glossary Terms** dialog box will open.
3. Click on the **Definition** field to add a **Definition** for the new term. If you would like to edit a term name, choose it and click the **Rename Term** button (next to the **Add New Term** button in the dialog box).

   Click the **Add New Term** button to add a new term at any time. In the **Terms** list, the term “New Term” will appear. Rename “New Term” and add a definition.
4. If you would like the new term to automatically link to its Glossary definition everywhere in the project, select the **Automatic links** check box. (A glossary term with this designation will be tagged with the Glossary Heading style, rather than the Glossary Heading (no auto links) style.)
5. Click **OK**.

   To delete a term, select it from the list and click the **Remove Term** button.

To re-alphabetize the Glossary

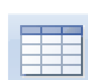
1. Open your Glossary document (.doc or .docx) in Word.
2. Click the **Sort** button . The Glossary terms will now be in alphabetical order.

Inserting a Standard Doc-To-Help Table

You can quickly insert a pre-formatted table into a Word document using the **Doc-To-Help Special Formatting** toolbar or ribbon. The table will be preformatted with the correct Doc-To-Help styles, and you can specify borders, indent, and shading of the heading row.

Note: The **Doc-To-Help Special Formatting** toolbar or ribbon will not be displayed in all templates. In those cases, use the Table functions in Microsoft® Word.

To insert a Standard Table

1. Open your source document (.doc or .docx) in Word.
2. Place your cursor at the point you’d like to insert a table.
3. Click the **Standard Doc-To-Help Table** button . The **Standard Doc-To-Help Table** dialog box will open
4. Specify the number of **Rows** and **Columns**.
5. Select the desired check boxes to add a border (**Create Borders**), align the table with the Body Text style (**Indent table to align with Body Text**), and shade the heading row gray (**Fill heading**).

   To make additional adjustments to your table, select it and right-click. Choose **Table Properties** from the menu to open the **Table Properties** dialog box.
6. Click **OK**.

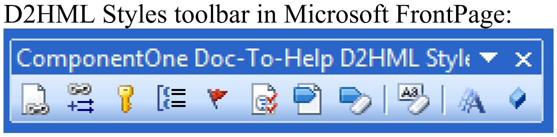
Editing HTML Documents

Styles control the look, and behavior of your final Targets. Doc-To-Help adds a toolbar to Microsoft FrontPage or Adobe Dreamweaver that is used to apply D2HML Styles; the **ComponentOne Doc-To-Help D2HML Styles** toolbar or menu. This toolbar allows you to use D2HML while using FrontPage or Dreamweaver as your editor.

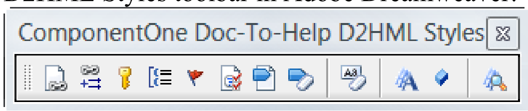
The **D2HML toolbar** will be installed on your machine automatically for Microsoft FrontPage; you will be prompted to install the toolbar in Adobe Dreamweaver during Doc-To-Help installation.

For information about using each toolbar button, see Using D2HML (Doc-To-Help Markup Language) on page 210.

D2HML Styles toolbar in Microsoft FrontPage:



D2HML Styles toolbar in Adobe Dreamweaver:



Other styles are applied using the appropriate method for that editor. See Applying Styles in Microsoft FrontPage and Adobe Dreamweaver on page 135 for details.

Since any style can be applied manually, HTML documents may be edited in the external HTML editor of your choice by applying styles and entering attributes manually.

To set your default HTML editor

1. Click on the **Doc-To-Help** button 
2. Click on the **Doc-To-Help Options** button. The **Options** dialog box will open.
3. Click the **HTML Editors** tab.
4. Choose your preferred editor and click **Set Default**.
5. Click **OK**.

Source and Target styles sheets (CSS files) are assigned using the Home tab (on page 98). See Guide to Templates and Styles (on page 5) for more information about Style Sheets.

Applying Styles in Microsoft FrontPage and Adobe Dreamweaver

To create links, inline text, etc., use **D2HML Styles**, which are applied with the **Doc-To-Help D2HML Styles** toolbar or menu in FrontPage or Dreamweaver. See Using D2HML (on page 210) for more information about D2HML Styles and how to apply them.

To apply a style in Microsoft FrontPage or Adobe Dreamweaver

- Start out by selecting the text.
  - In Microsoft® FrontPage®, use the **Formatting** toolbar buttons.
  - In Adobe® Dreamweaver®, use **Text > CSS Styles** or use the **Style** drop-down list in the **Properties** window. (Open the **Properties** window by selecting **Window > Properties**.)

Doc-To-Help built-in Paragraph and Character Styles, along with their HTML names

To learn more about how Paragraph and Character styles work, see Defining Character/Paragraph Styles and Topic Types on page 181.

To apply Character Styles, use the **Doc-To-Help D2HML Styles** toolbar or menu in FrontPage or Dreamweaver. See Using D2HML (on page 210) for more information.

Paragraph Style	HTML Name	Character Style	HTML Name
Heading 1	H1	C1H Conditional	.C1HConditional
Heading 2	H2	C1H Contents Title	.C1HContentsTitle
Heading 3	H3	C1H Context ID	.C1HContextID
Heading 4	H4	C1H Dropdown Text	.C1HDropdownText
Heading 5	H5	C1H Expand Text	.C1HExpandText
MidTopic	.MidTopic	C1H Group	.C1HGroup
RelatedHead	.RelatedHead	C1H Group Invisible	.C1HGroupInvisible
WhatsThis	.WhatsThis	C1H Group Link	.C1HGroupLink
Glossary Heading	.GlossaryHeading	C1H Index	.C1HIndex
Glossary Heading (no auto links)	.GlossaryHeadingnoautolinks	C1H Index Invisible	.C1HIndexInvisible
		C1H Inline Dropdown	.C1HInlineDropdown
		C1H Inline Expand	.C1HInlineExpand
		C1H Inline Popup	.C1HInlinePopup
		C1H Jump	.C1HJump
		C1H Keyword Link	.C1HKeywordLink
		C1H Link Tag	.C1HLinkTag
		C1H Link Tag Invisible	.C1HLinkTagInvisible
		C1H Manual	.C1HManual
		C1H Online	.C1HOnline
		C1H Popup	.C1HPopup
		C1H Popup Text	.C1HPopupText
		C1H Topic Properties	.C1HTopicProperties
		C1H Variable	

Cascading Style Sheets may be edited in Doc-To-Help, see Editing a CSS (on page 189) for more information.

Setting Document Properties

The **Document Properties** dialog box is used to view or change the properties of a document. If you would like to view or change the properties of a single topic, use the Topic Properties dialog box (on page 225).

To open the Document Properties dialog box

1. Select a document in the Documents pane (on page 108).

2. Open the **Home** tab.
3. Click the **Source** ribbon group dialog box launcher. The **Document Properties** dialog box will open.

   You can also right-click on a document in the **Documents** pane and choose **Properties**.

More on the Source ribbon group on page 99.

Document

Name: The read-only name of the document.

Type: Notes the document type — Word, HTML, or XHTML.

File size: The size of the source document.

File modified: The date and time of the last modification to the source document.

Basic

Single topic: Determines whether the source document can contain multiple topics or just one topic. Word documents are always multi-topic. HTML and XHTML documents can be either multi-topic or single-topic.

Title: Determines the title of a single-topic document. This property applies only to single-topic HTML or XHTML 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** fields of the **Document Properties** dialog box.

Style: Determines the style of the topic contained in a single-topic document. This property applies only to single-topic HTML or XHTML source documents.

Condition

These can also be set and will appear in the Source ribbon group (on page 99).

Platforms: Sets a platform-based condition for the selected document. The document will be included in all the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets: Sets a target-based condition for the selected document. The document will be included in all the target(s) selected.

Attributes: Set an attribute-based condition for the selected document. The document will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (**Project** tab > Project ribbon group > **Attributes** button) to create custom attributes.

Advanced

Keep outline numbers: When selected, outline numbers are included as part of topic titles and are present in help targets. This setting only affects the RTF files generated from source documents; it does not affect the source documents themselves. Modifying this setting for an individual document will override the settings for that document only.

Adjust left indent: Controls whether paragraph indentation is adjusted to account for wide margins when building online help. By default, this check box is selected to accommodate the standard Doc-To-Help templates. Clear this check box if you are using custom templates and want to preserve the indentation used in your source documents.

Include in Natural Search and in NetHelp Search: If selected, ComponentOne Natural Search is enabled for this document. If cleared, this document is excluded from the search.

Keep page breaks: When selected**,** retains the page break characters in the source documents during compilation of a Printed Manual Help target. Clear this check box to discard page break characters. Modifying this setting for an individual document will override the settings for that document only.

Plain text popups: If selected, generates a plain text only version of the help file for context-sensitive help topics. Modifying this setting for an individual document will override the settings for that document only.

Creating a Glossary

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

When you create a new project in Doc-To-Help, a Glossary document is added automatically. You can delete this document if you wish. If you'd like to flag a different document as your glossary, add it to the project and right-click on it in the Documents pane (on page 108). Choose **Glossary** from the menu.

Glossary entries are formatted by default with the **Glossary Heading** style, followed by the definition, formatted as **C1H Popup Topic Text**. Open the Glossary (on page 293) of the Doc-To-Help 2009 Help file to see the behavior of this default formatting. You can, of course, edit these styles to change this behavior, see Defining Character/Paragraph Styles and Topic Types on page 181.

Doc-To-Help can create automatic links to each Glossary item, or you can create manual links to Glossary items. To create a manual link, see Creating Links on page 212. Choose the **Glossary** tab. To set your project to automatically create glossary links, select the **Auto Glossary Links** check box in the **Project Styles** dialog box for the **Glossary Heading** Paragraph Style. See Paragraph Styles (on page 184) for more information.

To add a glossary entry, you can open the Glossary source document and add terms/definitions. You can also use the **Add Glossary Terms** button on the **Doc-To-Help** toolbar or ribbon when working in any Word source document. See Adding Terms to the Glossary (on page 133) for more information.

Creating and Converting Projects

Doc-To-Help includes several Wizards to make creating a new project, or converting an old one, easy.

The **Getting Started With Doc-To-Help Wizard** opens automatically when you open Doc-To-Help*****. This wizard walks you through starting a new project, opening an existing or sample project, or converting an existing project. It also has links to many handy resources that will help you learn more about Doc-To-Help. You can turn this wizard "off" by clearing the **Show this wizard at startup** check box in the wizard. If you'd like to turn it back on later, choose the **Doc-To-Help button > Doc-to-Help Options** button. On the **General** tab, select the **Show Getting Wizard at Startup** check box.

You can also create, open, or convert a project by doing the following:

To create a new project

Choose the **Doc-To-Help button** > **New Project**. The **New Project Wizard** will open. Follow the steps to create a new project. There is a step that allows you to copy settings from an old project, which can save you time if you already have an existing Doc-To-Help project with your desired setup.

See Customizing Your Project (on page 155) to learn more about customizing Help Targets, Windows, and Project Properties. See Working with Source Documents (on page 115), Documents pane (on page 108), and Adding a Document to a Project (on page 116) to learn more about documents. See Editing a CSS (on page 189) for more on editing Source and Target style sheets.

When you create a new project, a glossary document will automatically be created for you. You can delete this document if you wish. If you'd like to flag a different document as your glossary, add it to the project and right-click on it in the Documents pane (on page 108). Choose **Glossary** from the menu.

Watch the video: Starting a New Project in Doc-To-Help (3:21) [http://helpcentral.componentone.com/videos/D2H2009_Tour/StartingANewProject.html]

To open a project

1. Choose the **Doc-To-Help button** > **Open Project**. The **Open Doc-To-Help Project** dialog box will open.
2. Choose your project (.d2h file) and click **OK**.

If you choose a Doc-To-Help 2008 project, a message box will inform you that the project will be updated to the current version. Click **OK**.

To save a copy of a project

When you save a copy of your project, Doc-To-Help will append the name of the project folder with "Copy" unless you rename it. The project can keep the same name, or you can rename it.

1. Choose the **Doc-To-Help button** > **Save Project As**. The **Save Project As** wizard will open.
2. Enter the **Project Name**; the folder **Location** will update accordingly. If you would like to change the directory, click the **Browse** button.

3. If you would like you Target folders to be copied also, select the **Copy Target Folders** check box. If you have renamed the project, or if your Target folders are very large, you may not want to copy them.
4. Click the **Save** button.

To convert an existing legacy project (RoboHelp® HTML, HTML Help, RoboHelp® Word, WinHelp, and Doc-To-Help 2000)

Choose the **Doc-To-Help button** > **Convert**. Choose the appropriate project type. A Wizard will guide you through the process.

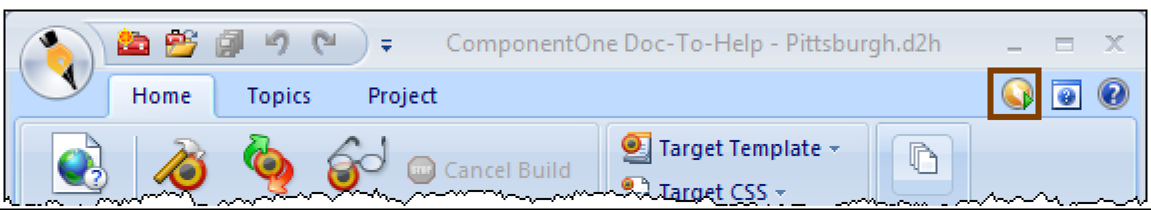
Please read the conversion notes for legacy files, which can be found at Converting RoboHelp HTML Notes (on page 140), Converting RoboHelp Word Notes (on page 144), Converting HTML Help Notes (on page 149), Converting Doc-To-Help 2000 Notes (on page 150), Converting WinHelp Notes (on page 150).

To close a project

Choose the **Doc-To-Help button** > **Close Project**.

For more information on copying and managing your project settings (Keywords, Groups, Variables, Attributes, etc.) see Importing and Exporting Project Settings on page 199.

*** Note:** If you have closed the Wizard, click the **Getting Started Wizard** button in the upper right to reopen it.



Converting RoboHelp HTML Notes

RoboHelp HTML to Doc-To-Help conversion notes. See Creating and Converting Projects (on page 139) for information on converting a legacy project.

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 pane. The 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 the 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 hierarchy) if possible, customized if necessary.

Index keywords — Index keywords are converted to Doc-To-Help index keywords. They appear on the Index and Groups pane in Doc-To-Help. Keywords are defined by D2HML hot spots, using the C1HIndexInvisible style, in topic files, so they can be modified in Doc-To-Help in the Index and Groups pane.

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 style, so they can be modified in Doc-To-Help in the Index and Groups pane.

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 Documents pane 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 documents.

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
HTMLHelp:	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
	Default Window	Help Targets dialog box, Default window field
	Binary TOC	Help Targets dialog box, Binary Table of Contents check box
WebHelp (converted to the Doc-To-Help NetHelp target):		
	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
JavaHelp:	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
	Enable TOC	Windows dialog box, Show Contents tab check box
	Enable Index	Windows dialog box, Show Favorites tab check box
	Enable Search	Windows dialog box, Show Search tab check box
	Enable Favorites	Windows dialog box, Show Favorites tab check box
	TOC Label	Help Targets dialog box, Contents heading field
	Index Label	Help Targets dialog box, Index heading field
	Search Label	Help Targets dialog box, Search field
	Favorites Label	Help Targets dialog box, Favorites field
Printed Documentation (converted to the Doc-To-Help Manual target):		

	Conditional Build Expression	Help Targets dialog box, Attributes field
	Name of the printed documentation	Help Targets dialog box, Supertitle field
	Include expanding text	Help Targets dialog box, Show Expanding Text check box
	Include drop-down text	Help Targets dialog box, Show Dropdown Text check box
	Chapter layout	TOC for the Manual target
Project Settings|Language|Advanced|LNG File:		
	[Common]	
	Contents	Help Targets dialog box, Contents Heading field
	Index	Help Targets dialog box, Index heading field
	Search	Help Targets dialog box, Search field
	SyncToc	Help Targets dialog box, Synchronize TOC field
	[BrowseSequence]	
	PreCaption	Help Targets dialog box, Previous field
	NextCaption	Help Targets dialog box, Next field
	[WebHelp]	
	IndexInputPromt	Help Targets dialog box, Index caption field
	FtsInputPromt	Help Targets dialog box, Search caption field
	TopicNotFound	Help Targets dialog box, Found zero field
	FtsBtnText	Help Targets dialog box, Search go field
	[PrintedDoc]	
	TableOfContents	Help Targets dialog box, Contents Heading field (Manual Target)

Printed documentation chapter layout — In 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 pane of Doc-To-Help. 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 a customized TOC. 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 Help Targets dialog box, Style Sheet field 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 dialog box. 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 the Help Targets dialog box, Attributes field.

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 C1HTopicProperties 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 C1HConditional style, in the Doc-To-Help topic text.

Hyperlinks — Hyperlinks are converted to D2HML hot spots, using the C1HJump style, in topic text.

Popups — Pop-ups are converted to D2HML hot spots, using the C1HPopup style, in topic text.

Keyword Links — Keyword links are converted to D2HML hot spots, using the C1HKeywordLink style, in topic text.

See Also Control — These are converted to D2HML hot spots, using the C1HGroupLink style, in topic text.

Related Topics Control — These are converted to D2HML hot spots, using the C1HGroupLink 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 and Groups pane of Doc-To-Help.

Text-only Popups — Text-only pop-ups are converted to D2HML inline pop-up hot spots using the C1HInlinePopup style.

Expanding Text — Expanding text is converted to D2HML inline expand hot spots using the C1HInlineExpand style.

Drop-down Text — Drop-down text is converted to D2HML inline drop-down hot spots using the C1HInlineDropdown 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 Topic Properties dialog box, URL field, 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.

Converting RoboHelp Word Notes

RoboHelp Word to Doc-To-Help conversion notes. See Creating and Converting Projects (on page 139) for information on converting a legacy project.

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 not selected. If you select 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 Help Target dialog box, Template field 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 ASCii Name field. 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 TopicTypes — 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 the Topic Properties dialog, Topic Type field to a topic type that has a specific window in the Project Styles dialog box, Window field. 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 Properties dialog box, Topic Type field as needed, to specify various topic properties.

Topic properties
Topic title ($ footnote) is converted to the Topic Properties dialog box, Display Title field 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 Properties dialog box, Contents Title field specified by a D2HML hot spot (style C1HContentsTitle) in topic text.

Topic window (> footnote) is converted to the Project Style dialog box, Window field.

Topic macro (! footnote) is converted to the Topic Properties dialog box, Macro field.

The Project Styles dialog box Nonscrolling check box is selected when a topic heading paragraph in the source document has its Word paragraph format setting Keep with next = True.

The Project Styles dialog box, Midtopic check box is selected 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 in the Index and Groups pane or in the topic text using D2HML.

See Also (A-keywords) — A-keywords are converted to Doc-To-Help groups. They appear in the Index and Groups pane 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 in the Index and Groups pane 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 Topics window 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 selecting the Project Styles dialog box Auto navigate check box. 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 dialog box.

Conditional build tags in RoboHelp layouts — Conditional Build Expressions in RoboHelp layouts are converted to the Help Targets dialog box, Attributes field.

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 Help Targets dialog box, Macro field.

Macros defined in the project file for a window (CONFIG:window section) are converted to the Windows dialog box, Macro field.

Macros defined for a topic (! Footnote) are converted to the Topic Properties dialog box, Macro field.

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.

Note: 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 the Display As Button check box selected.

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
HTMLHelp:	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
	Background Watermark	Help Targets dialog box, Image file field
WebHelp (converted to the Doc-To-Help NetHelp target):		
	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
	Background Watermark	Help Targets dialog box, Image file field
JavaHelp:	Conditional Build Expression	Help Targets dialog box, Attributes field
	Default Topic	Default Topic
	Navigation Pane TOC	Help Targets dialog box, Show Contents tab check box
	Navigation Pane Index	Help Targets dialog box, Show Index tab check box
	Navigation Pane Full-text Search	Help Targets dialog box, Show Search tab check box

Printed Documentation (converted to the Doc-To-Help Manual target):		
	Conditional Build Expression	Help Targets dialog box, Attributes field
	Name of the printed documentation	Help Targets dialog box, Supertitle field
	Chapter layout	TOC for the Manual target
Project Settings|Language|Advanced|LNG File:		
	[Common]	
	Contents	Help Targets dialog box, Contents Heading field
	Index	Help Targets dialog box, Index Heading field
	Search	Help Targets dialog box, Search field
	SyncToc	Help Targets dialog box, Synchronize TOC field
	[BrowseSequence]	
	PreCaption	Help Targets dialog box, Previous field
	NextCaption	Help Targets dialog box, Next field
	[WebHelp]	
	IndexInputPromt	Help Targets dialog box, Index caption field
	FtsInputPromt	Help Targets dialog box, Search caption field
	TopicNotFound	Help Targets dialog box, Found zero field
	FtsBtnText	Help Targets dialog box, Search go field
	[PrintedDoc]	
	TableOfContents	Help Targets dialog box, Contents heading field (Manual target)

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 pane. 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 a target-specific TOC. 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 Help Target dialog box, Template field 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 selecting the Project Styles dialog, Auto navigate check box.

Converting HTML Help Notes

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 so. If you only have a .CHM file, you can still do it, but first you must decompile it to an HTML Help project using either HTML Help Workshop or one of third-party CHM decompilation tools, such as KeyTools: http://www.keyworks.net/keytools.htm.

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. The Generic HTML Help converter does not convert RoboHelp-specific features.

See Creating and Converting Projects (on page 139) for information on converting a legacy project.

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 Documents pane. 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 in the Index and Groups pane. Keywords are defined by D2HML hot spots in topic files using the C1HIndexInvisible style, so they can be modified in Doc-To-Help in the Index and Groups pane or in topic HTML text using D2HML.

ALink keywords — ALink keywords defined in topics are converted to Doc-To-Help groups. They appear in the Index and Groups pane. Groups are defined by D2HML hot spots, using the C1HGroupInvisible style, in topic files, so they can be modified in Doc-To-Help either in the Index and Groups pane or in topic HTML text using D2HML.

Links — Links (A-tags) to topic files are converted to D2HML hot spots, using the C1HJump style, in topic text.

KLinks — Keyword links (KLink HTML Help controls) are converted to D2HML hot spots, using the C1HKeywordLink style, in topic text.

ALinks — Associative links (ALink HTML Help controls) are converted to D2HML hot spots, using the C1HGroupLink style, in topic text.

Related Topics Control — These are converted to D2HML hot spots, using the C1HGroupLink 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 in the Index and Groups pane. Select Groups in the Index and Groups pane toolbar.

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 Topic Property dialog box URL field, 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.

Converting Doc-To-Help 2000 Notes

If you have projects that were produced in Doc-To-Help 2000, you can convert them to Doc-To-Help 2009 by using the Doc-To-Help 2000 conversion wizard. See Creating and Converting Projects (on page 139) for information on converting a legacy project.

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.

Accessing the Converted Project

By default, Doc-to-Help 2009 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 Notes

The list below includes some known items that you should check after converting your Doc-To-Help 2000 project to Doc-To-Help 2009.

- 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.
- 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.
- 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 dialog box Label field to add a related topics heading.
- If a {bmc} reference or HTML passthrough code cannot locate a file, 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.

Converting WinHelp Notes

You can convert a WinHelp project produced with any help authoring tool other than RoboHelp to a Doc-To-Help project. You need the 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 http://www.geocities.com/mwinterhoff/helpdeco.htm.

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.

See Creating and Converting Projects (on page 139) for information on converting a legacy project.

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 not selected. If you select 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 Help Targets dialog box, Template field 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 Ascii name field in the Topic Properties dialog box. 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 Properties dialog box, Topic type field, as needed, to specify various topic properties.

Topic properties
Topic title ($ footnote) is converted to the Topic Properties dialog box, Display Title field specified by a D2HML hot spot (style C1HTopicProperties) in topic text.

Topic title specified in table of contents (TOC) is converted to the Topic Properties dialog box, Contents field specified by a D2HML hot spot (style C1HContentsTitle) in topic text.

Topic window (> footnote) is converted to the Project Styles dialog box, Windows field.

Topic macro (! footnote) is converted to the Topic properties dialog box, Macro field.

The Project Styles dialog box Nonscrolling check box is selected when a topic heading paragraph in the source document has its Word paragraph format setting Keep with next = True.

The Project Styles dialog box, Midtopic check box is selected 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 in the Index and Groups pane 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 in the Index and Groups pane or in the topic text using D2HML.

See Also (A-keywords) — A-keywords are converted to Doc-To-Help groups. They appear in the Index and Groups pane 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 in the Index and Groups pane 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 Topics window 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 selecting the Project Styles dialog box Auto navigate check box. 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 dialog box.

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 are 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 Help Targets dialog box, Macro field.

Macros defined in the project file for a window (CONFIG:window section) are converted to the Windows dialog box, Macro field.

Macros defined for a topic (! Footnote) are converted to the Topic Properties dialog box, Macro field.

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 the Display As Button check box selected.

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).

Customizing Your Project

You can customize your project so that your output looks and behaves the way you want it to. Start by deciding on your targets and then move on to specifying the look of your windows, character and paragraph styles, topic types, build attributes, variables, and project properties.

Creating Help Targets

In Doc-To-Help, Help Targets are the final output of your project. You can generate the following outputs with Doc-To-Help:

- Help 2.0 (on page 156)
- HTML Help (on page 158)
- JavaHelp (on page 162)
- Manual (on page 165) (Printed or Online)
- NetHelp (on page 167)
- WinHelp (on page 171)

The **Help Targets** dialog box is used to specify the targets that you would like to create and specify key settings for each output. You may create more than one output of a specific type — for example; you could have two different HTML Help outputs, one for "Administrator HTML Help" and one for "Manager HTML Help." Simply add those targets to the list and configure each as you wish. To make the text in each unique, use conditional text and variables.

For details on building targets, see Building a Help Target on page 235.

To open the Help Targets dialog box

1. Open the **Home** tab.
2. Click the **Target** ribbon group dialog box launcher. The **Help Targets** dialog box will open.

   Click on the desired target to view or edit its properties.

To add/delete a Target

1. In the **Help Targets** dialog box, click **Add New Target** in the box on the left.
2. Select the desired target from the drop-down list. The target will be added to the list with the default name **New Target**. Change the name by editing the **Name** field in the **Basic** section, then clicking anywhere in the dialog box to change it in the list on the left. You should create a new folder to store the new Target output in. Click the ellipsis button next to the **Folder** field to do so. (By default, the folder will be named **New Folder**.)

To delete a target, click the name of the target from the list and click **Remove Target**.

Help 2.0 Target

Use the following options to configure your Help 2.0 output. Please note that this output is only used for developing help for Microsoft Visual Studio.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Theme: The theme or "skin" used for the target. Themes are created/edited in the Theme Designer (on page 11). (The theme can also be changed using the **Theme** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Localization

Locale: Specifies the language in the 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). 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

CodePage: Defines the windows codepage number to be used by the help file. This integer is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Charset: Specifies one of the character set aliases listed in HTML Character Sets and Named Entities. Used for localization.

Background

Color: Displays the current color setting for the target help window. To change the color, click the adjacent button to select or define a color. If you change a System color for a NetHelp or JavaHelp Target, Doc-To-Help will display a message box about System colors and Windows. This message can be turned off using the Doc-To-Help Options (on page 45) dialog box.

Image file: Displays the image file to be used as a background in the target help window. To insert a background image, click the adjacent button to open the selection dialog.

Repeat image: Specifies how the background image is arranged in the target help window. Options are Tiled, Horizontal, or Vertical.

Advanced

Caption: The text displayed in the Caption bar of the Help window.

Color Reduction: Determines the color threshold for images embedded in Word source documents. This setting has no impact on images used in HTML source documents.

- **Keep original color format** — Images are not converted.
- **Reduce to 16 Colors** — Images with more than 16 colors are reduced to 16.
- **Reduce to 256 Colors** — Images with more than 256 colors are reduced to 256.

Image format: Determines the storage format in the Help target of the images used in Word source documents. This setting has no impact on images used in HTML source documents.

- **Original format** — All graphics are stored in the original format without conversion.
- **JPG always**— All graphics are stored in JPG files.
- **JPG if transformed**—Graphics are converted to JPG if they undergo any transformation when the help target is built.
- **PNG always**— All graphics are stored in PNG files.
- **PNG if transformed** — Graphics are converted to PNG if they undergo any transformation when the help target is built.

Graphics scaling: Determines the scaling applied to the graphics in Word source documents. This setting has no impact on images used in HTML source documents.

For graphics embedded within Word documents, **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in 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 browser that displays the online help target. Choose whichever produces best results. For WinHelp targets, **Scale in target** and **Scale in build** options are equivalent for all graphics, the resampling is performed by Doc-To-Help.

- **Do not scale graphics** — Graphics are displayed at their original size. If graphics are rescaled in the source Word document, they are reset.

- **Scale in target** — Graphics stay 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. Linked image files remain at their original size; scaling is delegated to the browser by setting the IMG tag dimensions.
- **Scale in build** — Graphics stay 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.

Graphics scale with fonts: When selected, all graphics in Word source documents are resized proportionally when the system font size is changed (for example, changing the Text Size in your browser). This setting has no impact on images used in HTML source documents.

Overwrite CSS: If selected, the existing cascading style sheet will be overwritten when building the Help target. This check box should be selected by default.

Generate XHTML: If selected, the online help output is generated as XHTML 1.0. The XHTML output generated conforms to the W3C XHTML 1.0 strict specification. For more information, see Generating XHTML Output on page 25.

Namespace: Specifies the Help 2.0 namespace that is used to register the Help file after a successful build. If this field is empty, the Base Name is used. If the Base Name field is empty, the file name of the project itself is used. Please note, the Namespace and Parent Namespace 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.

Parent Namespace: 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 Parent Namespace to "MS.VSCC.2003". Please note, the Namespace and ParentNamespace fields 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.

Related Links

Label: Specifies the text that precedes the subtopic links. If you clear this field, there will be no text above the links. If you would like to change the label text for a specific topic, change it in the Topic Properties dialog box (on page 225).

Label Style: The name of the style used to format the label that precedes the subtopic links.

Links Style: The name of the style used to format the automatically generated subtopic links.

Add separator between topic text and subtopic links: If selected, a dividing line will separate the topic text from the subtopic links.

Strings

Breadcrumbs label: For HMTL Help targets, specifies the text that precedes topic breadcrumbs (for example, “You are here:”). The breadcrumbs label is set with this field; to make additional changes or disable breadcrumbs, see Customizing with the Theme Designer on page 11.

HTML Help Target

Use the following options to configure your compiled HTML Help output. Your output will be a .chm file.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Theme: The theme or “skin” used for the target. Themes are created/edited in the Theme Designer (on page 11). (The theme can also be changed using the **Theme** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Localization

Locale: Specifies the language in the 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). 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

CodePage: Defines the windows codepage number to be used by the help file. This integer is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Charset: Specifies one of the character set aliases listed in HTML Character Sets and Named Entities. Used for localization.

Background

Color: Displays the current color setting for the target help window. To change the color, click the adjacent button to select or define a color. If you change a System color for a NetHelp or JavaHelp Target, Doc-To-Help will display a message box about System colors and Windows. This message can be turned off using the Doc-To-Help Options (on page 45) dialog box.

Image file: Displays the image file to be used as a background in the target help window. To insert a background image, click the adjacent button to open the selection dialog.

Repeat image: Specifies how the background image is arranged in the target help window. Options are Tiled, Horizontal, or Vertical.

Advanced

Color Reduction: Determines the color threshold for images embedded in Word source documents. This setting has no impact on images used in HTML source documents.

- **Keep original color format** — Images are not converted.
- **Reduce to 16 Colors** — Images with more than 16 colors are reduced to 16.
- **Reduce to 256 Colors** — Images with more than 256 colors are reduced to 256.

Image format: Determines the storage format in the Help target of the images used in Word source documents. This setting has no impact on images used in HTML source documents.

- **Original format** — All graphics are stored in the original format without conversion.
- **JPG always**— All graphics are stored in JPG files.
- **JPG if transformed**—Graphics are converted to JPG if they undergo any transformation when the help target is built.
- **PNG always**— All graphics are stored in PNG files.
- **PNG if transformed** — Graphics are converted to PNG if they undergo any transformation when the help target is built.

Graphics scaling: Determines the scaling applied to the graphics in Word source documents. This setting has no impact on images used in HTML source documents.

For graphics embedded within Word documents, **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in 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 browser that displays the online help target. Choose whichever produces best results. For WinHelp targets, **Scale in target** and **Scale in build** options are equivalent for all graphics, the resampling is performed by Doc-To-Help.

- **Do not scale graphics** — Graphics are displayed at their original size. If graphics are rescaled in the source Word document, they are reset.
- **Scale in target** — Graphics stay 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. Linked image files remain at their original size; scaling is delegated to the browser by setting the IMG tag dimensions.
- **Scale in build** — Graphics stay 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.

Graphics scale with fonts: When selected, all graphics in Word source documents are resized proportionally when the system font size is changed (for example, changing the Text Size in your browser). This setting has no impact on images used in HTML source documents.

Binary table of contents: If selected, a binary table of contents will be generated instead of a site map table of contents. In modular HTMLHelp projects (on page 257), this setting can have any value in the module projects, but the check box must be cleared in the hub project. You should select this check box if your main window uses the built-in Next and Previous buttons.

Overwrite CSS: If selected, the existing cascading style sheet will be overwritten when building the Help target. This check box should be selected by default.

Skip glossary: If selected, the Glossary topic is omitted from the generated Help contents. For each component Help file in a modular system (on page 257), select this checkbox to eliminate duplicate Glossary topics when viewing the hub.

Binary Index: If selected, generates a binary index instead of a site map index. In modular HTML Help projects (on page 257), this check box can have any value in the module projects, but must be selected in the hub project.

Generate XHTML: If selected, the online help output is generated as XHTML 1.0. The XHTML output generated conforms to the W3C XHTML 1.0 strict specification. For more information, see Generating XHTML Output on page 25.

Default window: Specifies the default window definition for the compiled help file. Choose from the list, or edit the available options by selecting **Edit Windows**, which will open the Windows dialog box (on page 176).

Topic list type: Selection determines whether ALink/KLink topic lists are displayed in a dialog box (the default) or a pop-up menu.

Related Links

Label: Specifies the text that precedes the subtopic links. If you clear this field, there will be no text above the links. If you would like to change the label text for a specific topic, change it in the Topic Properties dialog box (on page 225).

Label Style: The name of the style used to format the label that precedes the subtopic links.

Links Style: The name of the style used to format the automatically generated subtopic links.

Add separator between topic text and subtopic links: If selected, a dividing line will separate the topic text from the subtopic links.

Natural Search

Answer count: Sets the maximum number of answers to be returned when a natural language search is performed.

Confidence level: Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed. Matching answers that have a lower score than the set value will be discarded.

String no match: Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Search caption: If ComponentOne Natural Search is enabled, this text will be displayed on the Natural Search tab of HTML Help (on the left side of the tri-pane window), or the button in WinHelp.

Stem size: Sets the maximum number of characters used as a root word when building natural language search indexes. The default value of this property is 5. Highly technical documentation, where longer search terms are common, may benefit from a longer stem size.

Stem percent: Sets 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. This means that one character is added to the root word for every three additional characters in the word. Some international languages, where lengthy suffixes are common, may benefit from a higher percentage.

Enabled: If selected, ComponentOne Natural Search is enabled for this Help target. In HTML Help, WinHelp, and NetHelp targets, a Natural Search index will be generated. However, only HTML Help (a tab) and WinHelp (a button) will have a Natural Search interface. See Natural Search Requirements (on page 37) in Doc-To-Help Outputs and Deliverables (on page 30) for more information on files that need to be delivered for Natural Search.

Strings

Breadcrumbs label: For HMTL Help targets, specifies the text that precedes topic breadcrumbs (for example, "You are here:"). The breadcrumbs label is set with this field; to make additional changes or disable breadcrumbs, see Customizing with the Theme Designer on page 11.

Previous: The text displayed on the **Previous** button in your Help project. This field is editable.

Next: The text displayed on the **Next** button in your Help project. This field is editable.

JavaHelp Target

Use the following options to configure your JavaHelp output.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Theme: The theme or "skin" used for the target. Themes are created/edited in the Theme Designer (on page 11). (The theme can also be changed using the **Theme** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Localization

Locale: Specifies the language in the 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). 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

CodePage: Defines the windows codepage number to be used by the help file. This integer is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Charset: Specifies one of the character set aliases listed in HTML Character Sets and Named Entities. Used for localization.

Background

Color: Displays the current color setting for the target help window. To change the color, click the adjacent button to select or define a color. If you change a System color for a NetHelp or JavaHelp Target, Doc-To-Help will display a message box about System colors and Windows. This message can be turned off using the Doc-To-Help Options (on page 45) dialog box.

Image file: Displays the image file to be used as a background in the target help window. To insert a background image, click the adjacent button to open the selection dialog.

Repeat image: Specifies how the background image is arranged in the target help window. Options are Tiled, Horizontal, or Vertical.

Advanced

Caption: The text displayed in the Caption bar of the Help window.

Color Reduction: Determines the color threshold for images embedded in Word source documents. This setting has no impact on images used in HTML source documents.

- **Keep original color format** — Images are not converted.
- **Reduce to 16 Colors** — Images with more than 16 colors are reduced to 16.
- **Reduce to 256 Colors** — Images with more than 256 colors are reduced to 256.

Image format: Determines the storage format in the Help target of the images used in Word source documents. This setting has no impact on images used in HTML source documents.

- **Original format** — All graphics are stored in the original format without conversion.
- **JPG always**— All graphics are stored in JPG files.
- **JPG if transformed**—Graphics are converted to JPG if they undergo any transformation when the help target is built.
- **PNG always**— All graphics are stored in PNG files.
- **PNG if transformed** — Graphics are converted to PNG if they undergo any transformation when the help target is built.

Graphics scaling: Determines the scaling applied to the graphics in Word source documents. This setting has no impact on images used in HTML source documents.

For graphics embedded within Word documents, **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in 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 browser that displays the online help target. Choose whichever produces best results. For WinHelp targets, **Scale in target** and **Scale in build** options are equivalent for all graphics, the resampling is performed by Doc-To-Help.

- **Do not scale graphics** — Graphics are displayed at their original size. If graphics are rescaled in the source Word document, they are reset.
- **Scale in target** — Graphics stay 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. Linked image files remain at their original size; scaling is delegated to the browser by setting the IMG tag dimensions.
- **Scale in build** — Graphics stay 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.

Scale font sizes: This field is used to scale the font sizes in the help target. It is enabled when the **Ignore Font Sizes** check box is cleared (false). JavaHelp has a known problem that makes all fonts appear smaller (approximately 1.3 times smaller) than they should be. Setting this field to 1.3 will scale all fonts to adjust their sizes. Enabling the **Ignore Font Sizes** check box will disable this field, but will also make adjustments so there are consistent default JavaHelp font sizes.

Ignore font sizes: If enabled, the font sizes specified in the source document are ignored and default font sizes are used in the help target. This check box is used in conjunction with **Ignore font names**. If this check box is cleared, the **Scale Font Sizes** field controls the scale of the font sizes.

Ignore font names: If enabled, the font names specified in the source document are ignored and default font names are used in the help target. This check box is used in conjunction with **Ignore font sizes**.

Overwrite CSS: If selected, the existing cascading style sheet will be overwritten when building the Help target. This check box should be selected by default.

Show expanding text: If selected, expanding text is shown in the target, otherwise it is removed.

Show dropdown text: If selected, dropdown text is shown in the target, otherwise it is removed. Dropdown text is a Dynamic HTML effect that some help targets do not support.

Generate XHTML: If selected, the online help output is generated as XHTML 1.0. The XHTML output generated conforms to the W3C XHTML 1.0 strict specification. For more information, see Generating XHTML Output on page 25.

Related Links

Label: Specifies the text that precedes the subtopic links. If you clear this field, there will be no text above the links. If you would like to change the label text for a specific topic, change it in the Topic Properties dialog box (on page 225).

Label Style: The name of the style used to format the label that precedes the subtopic links.

Links Style: The name of the style used to format the automatically generated subtopic links.

Add separator between topic text and subtopic links: If selected, a dividing line will separate the topic text from the subtopic links.

Strings

Breadcrumbs label: For HMTL Help targets, specifies the text that precedes topic breadcrumbs (for example, "You are here:"). The breadcrumbs label is set with this field; to make additional changes or disable breadcrumbs, see Customizing with the Theme Designer on page 11.

Previous: The text displayed on the **Previous** button in your Help project. This field is editable.

Next: The text displayed on the **Next** button in your Help project. This field is editable.

Contents: The text displayed on the **Contents** button in your Help project. This field is editable.

Index: The text displayed on the **Index** button in your Help project. This field is editable.

Search: The text displayed on the **Search** button in your Help project. This field is editable.

Favorites: The text displayed on the **Favorites** button in your Help project. This field is editable.

Found many: The text displayed when more than one topic is found in your Help project. This field is editable.

Found one: The text displayed when only one topic is found in your Help project. This field is editable.

Found zero: The text displayed when no topics are found in your Help project. This field is editable.

Return to Index: The text displayed on the **Return to Index** button in your Help project. This field is editable.

Manual Target

Use the following options to configure your Manual output. Your output will be a Microsoft Word file, and a pdf if you choose.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Build

Output format: Used to set the default Word format for the manual output. The default is Word 200-2003 (.doc).

Generate PDF target: If selected, enables the generation of a PDF from the manual target.

Create outline in PDF target: If selected, the TOC of the manual target is included in the TOC pane on the left side of the PDF viewer.

Live links: If enabled, hyperlinks are live in the manual Word document. If the document is converted to PDF, the hyperlinks will also be live in the PDF. The **Create Master Document** check box must be enabled for this setting to take effect. If this check box is cleared, only cross-references are hyperlinks in the manual Word document.

There is an issue with live links in the Adobe Acrobat PDF converter (including 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, clear the **Enable accessibility and reflow with Tagged PDF** check box in Adobe Acrobat before creating a PDF or use a PDF converter other than Adobe Acrobat.

Advanced

Fix lists: Controls how numbered lists are corrected in the printed manual output when the **Create master document** check box is enabled. The numbering issues this field corrects were for the most part resolved in Word 2002.

- **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.

Show expanding text: If selected, expanding text is shown in the target, otherwise it is removed.

Show dropdown text: If selected, dropdown text is shown in the target, otherwise it is removed. Dropdown text is a Dynamic HTML effect that some help targets do not support.

Create master document: If selected, subdocuments are linked together into a single master document in printed manual builds. If cleared, the master document will use Word RD fields to reference subdocuments.

Use template text: If selected, 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. If cleared, 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.

Strings

Title: The text displayed below the **Supertitle** on the cover of the printed manual. This field is editable. This field is often the book's name, such as "Administrator Guide" or "User Guide."

Supertitle: The text displayed above the **Title** on the cover of the printed manual. This field is editable. This field is often the product or company name.

"By" line: The byline displayed on the cover of the printed manual. This field is editable. It is usually the author's name.

Contents heading: This field is used only if there is no **Template** selected for the manual, or if the **Use template text** check box is cleared. Otherwise, the template text is used.

Index heading: This field is used only if there is no **Template** selected for the manual, or if the **Use template text** check box is cleared. Otherwise, the template text is used.

NetHelp Target

Use the following options to configure your NetHelp output. NetHelp is a cross-platform, browser-based output that is displayed in the user's default browser. It is uncompiled HTML Help, so the number of files delivered for the project will depend on the number of topics in the project.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30. This section includes information on installing NetHelp locally and on a server.

See Section 508 Compliant NetHelp (on page 35) for more information on creating accessible NetHelp.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Theme: The theme or "skin" used for the target. Themes are created/edited in the Theme Designer (on page 11). (The theme can also be changed using the **Theme** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Localization

Locale: Specifies the language in the 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). 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

CodePage: Defines the windows codepage number to be used by the help file. This integer is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Charset: Specifies one of the character set aliases listed in HTML Character Sets and Named Entities. Used for localization.

Background

Color: Displays the current color setting for the target help window. To change the color, click the adjacent button to select or define a color. If you change a System color for a NetHelp or JavaHelp Target, Doc-To-Help will display a message box about System colors and Windows. This message can be turned off using the Doc-To-Help Options (on page 45) dialog box.

Image file: Displays the image file to be used as a background in the target help window. To insert a background image, click the adjacent button to open the selection dialog.

Repeat image: Specifies how the background image is arranged in the target help window. Options are Tiled, Horizontal, or Vertical.

Advanced

Caption: The text displayed in the Caption bar of the Help window.

Color Reduction: Determines the color threshold for images embedded in Word source documents. This setting has no impact on images used in HTML source documents.

- **Keep original color format** — Images are not converted.
- **Reduce to 16 Colors** — Images with more than 16 colors are reduced to 16.
- **Reduce to 256 Colors** — Images with more than 256 colors are reduced to 256.

Image format: Determines the storage format in the Help target of the images used in Word source documents. This setting has no impact on images used in HTML source documents.

- **Original format** — All graphics are stored in the original format without conversion.
- **JPG always**— All graphics are stored in JPG files.
- **JPG if transformed**—Graphics are converted to JPG if they undergo any transformation when the help target is built.
- **PNG always**— All graphics are stored in PNG files.
- **PNG if transformed** — Graphics are converted to PNG if they undergo any transformation when the help target is built.

Graphics scaling: Determines the scaling applied to the graphics in Word source documents. This setting has no impact on images used in HTML source documents.

For graphics embedded within Word documents, **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in 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 browser that displays the online help target. Choose whichever produces best results. For WinHelp targets, **Scale in target** and **Scale in build** options are equivalent for all graphics, the resampling is performed by Doc-To-Help.

- **Do not scale graphics** — Graphics are displayed at their original size. If graphics are rescaled in the source Word document, they are reset.
- **Scale in target** — Graphics stay 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. Linked image files remain at their original size; scaling is delegated to the browser by setting the IMG tag dimensions.
- **Scale in build** — Graphics stay 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.

Graphics scale with fonts: When selected, all graphics in Word source documents are resized proportionally when the system font size is changed (for example, changing the Text Size in your browser). This setting has no impact on images used in HTML source documents.

Default file: The file name to be used as the base URL for the NetHelp target. Typically, the filename will be *default.htm* or *index.html*.

Search stop list: Specifies the path to the text file containing noise words (words ignored in searches). This field is empty by default and a default noise word list (below) is used for searches. If you wish to create your own list and map to it here, this default list is overridden. Your file must be a .txt and each noise word must be a separate line in the file. See Natural Search Requirements (on page 37) in Doc-To-Help Outputs and Deliverables (on page 30) for more information on the stop list.

A, about, after, against, all, also, among, an, and, are, as, at, be, became, because, been, between, but, by, can, come, do, during, each, early, for, form, found, from, had, has, have, he, her, his, however, in, include, into, is, it, its, late, later, me, med, made, many, may, more, most, near, no, non, not, of, on, only, or, other, over, several, she, some, such, than, that, the, their, then, there, these, they, this, through, to, under, until, use, was, we, were, when, where, which, who, with, you

Search Type: In NetHelp targets, specifies the type of search used — Java or JavaScript. If NetHelp will be installed locally, JavaScript search is recommended to avoid dependence on Java being installed on end-user machines. JavaScript search supports wildcards * (any substring) and ?(any character).

If NetHelp will be installed on a server, Java search is recommended because the search will be slightly faster, and yield higher-quality results. NetHelp deployed on a server does not require that Java be installed on end-user machines.

Accessibility mode: Determines whether special features for Section 508 compliance are enabled. See Section 508 Compliant NetHelp (on page 35) for more information.

- **Normal** — No special features for Section 508 compliance are enabled.
- **Section 508 Compliance** — Special features for 508 compliance are enabled.

Frameset: If selected, a frameset version of the HTML help project is generated, with the left frame displaying the contents and the right frame displaying the help topics.

Overwrite CSS: If selected, the existing cascading style sheet will be overwritten when building the Help target. This check box should be selected by default.

Dynamic table of contents: If selected, a Dynamic HTML version of the contents page with an expandable outline is generated.

Generate XHTML: If selected, the online help output is generated as XHTML 1.0. The XHTML output generated conforms to the W3C XHTML 1.0 strict specification. For more information, see Generating XHTML Output on page 25.

Email: The text to be used for the **Email** option in NetHelp

Related Links

Label: Specifies the text that precedes the subtopic links. If you clear this field, there will be no text above the links. If you would like to change the label text for a specific topic, change it in the Topic Properties dialog box (on page 225).

Label Style: The name of the style used to format the label that precedes the subtopic links.

Links Style: The name of the style used to format the automatically generated subtopic links.

Add separator between topic text and subtopic links: If selected, a dividing line will separate the topic text from the subtopic links.

Strings

Breadcrumbs label: For HMTL Help targets, specifies the text that precedes topic breadcrumbs (for example, "You are here:"). The breadcrumbs label is set with this field; to make additional changes or disable breadcrumbs, see Customizing with the Theme Designer on page 11.

Previous: The text displayed on the **Previous** button in your Help project. This field is editable.

Next: The text displayed on the **Next** button in your Help project. This field is editable.

Contents: The text displayed on the **Contents** button in your Help project. This field is editable.

Index: The text displayed on the **Index** button in your Help project. This field is editable.

Index caption: The text to be displayed in the Index window above the index text box. This field is editable.

Search: The text displayed on the **Search** button in your Help project. This field is editable.

Search caption: If ComponentOne Natural Search is enabled, this text will be displayed on the Natural Search tab of HTML Help (on the left side of the tri-pane window), or the button in WinHelp.

Search Go: The text to be displayed in the Search button within the search window. This field is editable.

Search highlight: The text to be used for the check box controlling the highlighting of search phrases.

Favorites: The text displayed on the **Favorites** button in your Help project. This field is editable.

Favorites caption: The text to be displayed in the **Favorites** tab of NetHelp navigation. This field is editable.

Favorites Add: The text to be used for the **Add to Favorites** option in the **Favorites** tab of NetHelp navigation.

Favorites Delete: The text to be used for the **Delete** option in the **Favorites** tab of NetHelp navigation.

Synchronize TOC: The text to be used for the **Sync TOC** option in NetHelp.

Print: The text to be used for the **Print** option in NetHelp.

Email: The text to be used for the **Email** option in NetHelp

Hide navigation pane: The text to be used for the **Hide Navigation Pane** option in NetHelp.

Found zero: The text displayed when no topics are found in your Help project. This field is editable.

WinHelp Target

Use the following options to configure your WinHelp output.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 30.

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This is the internal name for the project; the name used as the label on the final output is set in the Advanced section. This name is also used when you select a target using the **View Target** button in the Target ribbon group (on page 98).

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Style sheet: The cascading style sheet used to format the target. This style-sheet controls the formatting of the Target for HTML source documents. Target CSS styles will override any matching styles in the Source CSS. See the Guide to Templates and Styles on page 5. (The style sheet can also be changed using the **Target CSS** button in the **Target Design** ribbon group.)

Template: The Word template used to format the target. This template controls the formatting of the Target for Word source documents. Target template styles will override any matching styles in the Source Template. See the Guide to Templates and Styles on page 5. (The template can also be changed using the **Target Template** button in the **Target Design** ribbon group.)

Attributes: The Attributes or Expression this target includes in the final build. For more information, see Defining Attributes on page 175.

Folder: The folder where the target output will be stored in your Doc-To-Help project. If you change the name of the folder after generating this target, you should delete the original folder to avoid confusion.

Localization

Locale: Specifies the language in the 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). 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

CodePage: Defines the windows codepage number to be used by the help file. This integer is most commonly used when producing help files in foreign languages, providing special character sets for translation on the screen or in print.

Charset: Specifies one of the character set aliases listed in HTML Character Sets and Named Entities. Used for localization.

Advanced

Title: The text displayed in the Caption bar of the Help window. WinHelp captions are limited to 50 characters.

Color Reduction: Determines the color threshold for images embedded in Word source documents. This setting has no impact on images used in HTML source documents.

- **Keep original color format** — Images are not converted.
- **Reduce to 16 Colors** — Images with more than 16 colors are reduced to 16.
- **Reduce to 256 Colors** — Images with more than 256 colors are reduced to 256.

Graphics scaling: Determines the scaling applied to the graphics in Word source documents. This setting has no impact on images used in HTML source documents.

For graphics embedded within Word documents, **Scale in target** and **Scale in build** options are equivalent, the size of the picture is the same as it is displayed in 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 browser that displays the online help target. Choose whichever produces best results. For WinHelp targets, **Scale in target** and **Scale in build** options are equivalent for all graphics, the resampling is performed by Doc-To-Help.

- **Do not scale graphics** — Graphics are displayed at their original size. If graphics are rescaled in the source Word document, they are reset.
- **Scale in target** — Graphics stay 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. Linked image files remain at their original size; scaling is delegated to the browser by setting the IMG tag dimensions.
- **Scale in build** — Graphics stay 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.

Copyright: The custom copyright notice that will appear in the Version dialog box of WinHelp.

Citation: The citation that will be added to the end of any information (except a context-sensitive popup window) that is copied from the WinHelp file.

Compression: Determines the level of file compression used by the WinHelp compiler.

- **None** — No compression.
- **Choose best method** — 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.

Skip glossary: If selected, the Glossary topic is omitted from the generated Help contents. For each component Help file in a modular system (on page 257), select this checkbox to eliminate duplicate Glossary topics when viewing the hub.

One browse sequence: If selected, a continuous browse sequence that spans multiple source documents will be generated.

Show expanding text: If selected, expanding text is shown in the target, otherwise it is removed.

Show dropdown text: If selected, dropdown text is shown in the target, otherwise it is removed. Dropdown text is a Dynamic HTML effect that some help targets do not support.

Hub contents file: The name of the contents (.cnt) file to associate with the Help file. For modular WinHelp systems (on page 257), use this field to associate a component Help file with its hub.

WinHelp macro: Specifies a macro to run when the Help file is opened. The macro set here for the WinHelp target will override the WinHelp macro set in the Windows dialog box (on page 176).

Related Links

Label: Specifies the text that precedes the subtopic links. If you clear this field, there will be no text above the links. If you would like to change the label text for a specific topic, change it in the Topic Properties dialog box (on page 225).

Label Style: The name of the style used to format the label that precedes the subtopic links.

Links Style: The name of the style used to format the automatically generated subtopic links.

Add separator between topic text and subtopic links: If selected, a dividing line will separate the topic text from the subtopic links.

Natural Search

Answer count: Sets the maximum number of answers to be returned when a natural language search is performed.

Confidence level: Sets the minimum score percentage that controls which answers will be displayed when a natural language search is performed. Matching answers that have a lower score than the set value will be discarded.

String no match: Specifies the text used in the message box when ComponentOne Natural Search does not find a match for a question.

Search caption: If ComponentOne Natural Search is enabled, this text will be displayed on the Natural Search tab of HTML Help (on the left side of the tri-pane window), or the button in WinHelp.

Stem size: Sets the maximum number of characters used as a root word when building natural language search indexes. The default value of this property is 5. Highly technical documentation, where longer search terms are common, may benefit from a longer stem size.

Stem percent: Sets 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. This means that one character is added to the root word for every three additional characters in the word. Some international languages, where lengthy suffixes are common, may benefit from a higher percentage.

Enabled: If selected, ComponentOne Natural Search is enabled for this Help target. In HTML Help, WinHelp, and NetHelp targets, a Natural Search index will be generated. However, only HTML Help (a tab) and WinHelp (a button) will have a Natural Search interface. See Natural Search Requirements (on page 37) in Doc-To-Help Outputs and Deliverables (on page 30) for more information on files that need to be delivered for Natural Search.

Utilizing Conditions

Conditions make it possible to efficiently single-source your content.

Using conditions, you can easily include/exclude information from a Help Target. You can quickly mark text, graphics, topics, or documents so that they are included or excluded from your final output — manuals or any combination of online Help.

Some examples of using conditions:

- If you'd like specific graphics to appear only in your printed output, mark those graphics with the **Platform** condition named "Printed Manual."
- You'd like links to related topics to be included only in your NetHelp output (and excluded from your Manual Targets). Mark that text with the **Target** condition named "NetHelp."
- Reminder notes to you and your team can be included in your project for reference, but can be marked with the "Internal" **Attribute** so that they are not included with the Targets delivered to customers.
- You'd like to output two different manuals; one for "Pharmacists" and one for "Nurses." The "Nurses" do not need certain administrative information. Simply create an **Attribute** condition for each audience. Mark the Administrative information with the "Pharmacist" attribute.

With Doc-To-Help, you can conditionalize:

- **Text** (and **graphics**) To apply, see Marking Text as Conditional on page 220. Text within **Variables** can also be conditionalized, see Creating Variables on page 194.
- **Topics** To apply, see Setting Topic Conditions on page 229.
- **Documents** To apply, see Setting Document Properties on page 136.

Conditions available are:

- **Platforms**
  HTML (Any) — Text/Topic/Document will be included in all HTML Targets (includes JavaHelp).
  HTML Help 1.x — Text/Topic/Document will be included in compiled HTML Help Targets (.chms).
  JavaHelp — Text/Topic/Document will be included in all JavaHelp Targets.
  Microsoft Help 2.0 — Text/Topic/Document will be included in Help 2.0 Targets.
  NetHelp — Text/Topic/Document will be included in NetHelp (uncompiled HTML Help) Targets.
  Online Help — Text/Topic/Document will be included in all online Help Targets (all Targets except Printed Manual).
  Printed Manual — Text/Topic/Document will be included in printed manual Targets (.doc, .docx, and .pdf).
  WinHelp 4.0 — Text/Topic/Document will be included in all WinHelp Targets.
- **Targets** (the list below are the new project defaults; you can add additional and delete those you don't need)
  (Your project name) **Microsoft Help 2.0** — Text/Topic/Document will be included in this Help 2.0 Target.
  (Your project name) **HTML Help** — Text/Topic/Document will be included in this compiled HTML Help Target (.chm).
  (Your project name) **JavaHelp** — Text/Topic/Document will be included in this JavaHelp Target.
  (Your project name) **Manual** — Text/Topic/Document will be included in this printed manual Target (.doc, .docx, and .pdf).
  (Your project name) **NetHelp** — Text/Topic/Document will be included in this NetHelp (uncompiled HTML Help) Target.
  (Your project name) **WinHelp** — Text/Topic/Document will be included in this WinHelp Target.
  Any custom Targets you create will be included also.

- **Attributes**
  Two default Attributes are included with Doc-To-Help (**Internal** and **Release**), but you create your own.

Platforms are defaults, and Targets will vary by project, but Attributes are created by you to meet your specific needs. See Defining Attributes (on page 175) for information about creating Attributes.

To include conditions in your build

Once you have applied **Platform** and **Target** conditions to your text, topics, and/or documents, you don't need to do anything else but build your Target(s). The appropriate information will automatically be included/excluded from your output.

If you have created and applied custom **Attributes** in your project, you need to choose the Attributes for each Help Target in the Help Targets dialog box (on page 155). Click the drop-down on the **Attributes** field and choose the appropriate Attribute(s) or create a custom Expression. See Defining Attributes on page 175.

Defining Attributes

Attributes make it possible to assign user-defined build criteria to text, topics, documents, and styles — which makes it possible to single source one project several different ways (for example, you could create both an Administrator and Manager version of a manual and/or help system from the same project).

Once you have created an attribute, it can be chosen when you mark text as Conditional Text (on page 220). Build attributes can also be assigned to specific Character Styles (on page 181), Topics (on page 229), and even entire Documents (on page 136). Then specify the attributes to include in the build of each Target (on page 155) — and when that Target is built the text, topic, or document flagged with those attributes will appear only in the appropriate output.

Doc-To-Help has two default Build attributes built in: **Internal** and **Release**.

To open the Attributes dialog box

1. Open the **Project** tab.
2. From the **Project** ribbon group, click the **Attributes** button. The **Attributes** dialog box will open.

To create a new attribute

In the **Attributes** dialog box, click the **Add New Attribute** button. An Attribute named **NewAttribute** will appear in the Attributes list. The **NewAttribute** name will initially be editable; if you wish to change it later, select it, then click the **Edit** (pencil) icon.

Example: An example of an attribute name would be "Audience." The "Audience" attribute would then need values.

To add a new value to an attribute

In the **Attributes** dialog box, click the **Add New Value** button. A Value named **Value** will appear in the list. The **Value** name will initially be editable; if you wish to change it later, select it, then click the **Edit** (pencil) icon.

Example: An example of attribute values for the "Audience" attribute would be "Pharmacist" and "Nurse."

To use attributes

1. Create an attribute.
2. Assign it in any combination of five ways:
   - Within a document as **Conditional Text**. See Marking Text as Conditional on page 220.

- To applicable **Character Styles**. See Defining Character/Paragraph Styles and Topic Types on page 181.
- To **Text Variables**. See Creating Variables on page 194.
- To **Topics**. See Setting Topic Conditions on page 229.
- To entire Documents. See Setting Document Properties on page 136.

3. Specify which attributes should be included in the build of each Target type. You can create additional targets if you need to (for example, you want to create one HTML Help Target that includes Attribute A, and another HTML Help Target that includes Attribute B). See Creating Help Targets (on page 155) for more information on creating additional Targets.

   When you select the drop-down next to the **Attributes** field in the **Help Targets** dialog box:

- Click on the **Attributes** tab to assign one or more Attributes to the Target. Select the check box(es) next to the appropriate Attributes. The Attributes chosen will display in the **Attributes** field.
- Click on the **Expression** tab to create a custom expression using your attributes. Expressions are statements that combine attributes in more complex ways. The Expression created will display in the **Attributes** field.

More on the Target ribbon group on page 98.

Setting the Help Window Display

The **Windows** dialog box controls the position and — depending on the Target — the buttons, caption, background color, and other characteristics of Help windows. Any window options not specified here will be set using the Theme Designer (on page 11).

This dialog box has three panels:

- The left panel displays the names of the windows already created. Click on a window name to edit it in the center panel. You can also add a new window in this panel.
- The center panel displays complete information about the window chosen on the left. Choose the Help Target windows you'd like to edit from the drop-down box at the top. Unavailable options for the window selection will be grayed out.
- The right panel displays how each window chosen on the left will be positioned on your desktop. All windows in the left panel with selected check boxes will be displayed by default.

To open the Windows dialog box

1. Open the **Project** tab.
2. From the **Project** ribbon group, click the **Windows** button. The **Windows** dialog box will open.

To add a new window

1. In the **Windows** dialog box, choose the proper **Help Target** from the center panel drop-down list.
2. Click on the **Add New Window** button in the left panel.

   The window will be added to the list with the default name **newwindow**. Change the name by editing the **Name** field in the **Basic** section, then clicking anywhere in the dialog box to change it in the list on the left.

To edit an existing window

1. In the **Windows** dialog box, choose the proper **Help Target** from the center panel drop-down list.

2. Click on the window you wish to edit in the left panel. If you would like to edit the name of the window, edit the **Name** field in the **Basic** section.
3. Make the desired changes to the window options and click **OK**.

More on the Project ribbon group on page 106.

WinHelp Window

Use the following options to configure your WinHelp windows. Unavailable options will be grayed out.

Basic

Name: The name of the window. It is read-only; if you'd like to create a new window, Add one in the panel on the left. See Setting the Help Window Display (on page 176) for more details.

Title: For WinHelp targets, the text that will be displayed in the caption bar of the help window. Because of limitations imposed by the Windows help compiler, WinHelp captions are limited to 50 characters.

Position

Left: Determines the location of the left edge of the help window. Works in conjunction with the Top field to determine the position of the help window on the screen.

Top: Determines the location of the top edge of the help window. Works in conjunction with the Left field to determine the position of the help window on the screen.

Width: Determines the width of the help window.

Height: Determines the height of the help window.

Adjust for screen size: Determines whether the help window conforms to the user's screen resolution. If selected, the help window will not conform to the resolution of the reader's display.

Auto size height: If selected, the heights of secondary Help windows are automatically resized to fit the length of the current topic.

Always on top: If selected, the help window always remains on top of all other windows open on the desktop.

Maximized: If selected, the help window is automatically maximized when displayed.

Buttons

Contents: If selected, the **Contents** button is included in the help window.

Index: If selected, the **Index** button is included in the help window.

Find: If selected, the **Find** button is included in the help window.

Help Topics: If selected, the **Help Topics** button is included in the help window.

Back: If selected, the **Back** button is included in the help window.

Print: If selected, the **Print** button is included in the help window.

Options: If selected, the **Options** button is included in the help window.

Browse: If selected, the **Browse** buttons are included in the help window.

Color

Nonscrolling: Displays the current color setting for the non-scrolling area of WinHelp help windows. To change the color, click the button to open a color selection dialog.

Topic region: Displays the current color setting for the topic area of WinHelp help windows. To change the color, click the button to open a color selection dialog

Advanced

Macro: Specifies the macro to run when this window is opened. The macro specified for the WinHelp target in the Help Targets dialog box (on page 155) will override the macro set in this dialog box. This option is only available for WinHelp targets.

NetHelp Window

Use the following options to configure your NetHelp windows. Unavailable options will be grayed out.

Basic

Name: The name of the window. It is read-only; if you'd like to create a new window, Add one in the panel on the left. See Setting the Help Window Display (on page 176) for more details.

Position

Left: Determines the location of the left edge of the help window. Works in conjunction with the Top field to determine the position of the help window on the screen.

Top: Determines the location of the top edge of the help window. Works in conjunction with the Left field to determine the position of the help window on the screen.

Width: Determines the width of the help window.

Height: Determines the height of the help window.

Navigation Pane

Show Contents tab: If selected, the **Contents** tab is included in the help window.

Show Index tab: If selected, the **Index** tab is included in the help window.

Search tab: If selected, the **Search** tab is included in the help window.

Show Favorites tab: If selected, the **Favorites** tab is included in the help window.

HTML Help Window

Use the following options to configure your HTML Help windows. Unavailable options will be grayed out.

Basic

Name: The name of the window. It is read-only; if you'd like to create a new window, Add one in the panel on the left. See Setting the Help Window Display (on page 176) for more details.

Caption: The text that will be displayed in the caption bar of the Help window.

Tri-pane format: If selected, enables the standard tri-pane help format for HTML Help.

Position

Left: Determines the location of the left edge of the help window. Works in conjunction with the Top field to determine the position of the help window on the screen.

Top: Determines the location of the top edge of the help window. Works in conjunction with the Left field to determine the position of the help window on the screen.

Width: Determines the width of the help window.

Height: Determines the height of the help window.

Save user position: If selected, stores the size and position of the help window in the registry when the user modifies it.

Always on top: If selected, the help window always remains on top of all other windows open on the desktop.

Maximized: If selected, the help window is automatically maximized when displayed.

Buttons

Hide: If selected, the **Show** and **Hide** buttons are included in the help window.

Locate: If selected, the **Locate** button is included in the help window.

Previous: If selected, the **Previous** button is included in the help window. The **Binary table of contents** check box in the Help Targets dialog box (on page 155) must be selected for the navigation buttons to function.

Next: If selected, the **Next** button is included in the help window. The **Binary table of contents** check box in the Help Targets dialog box (on page 155) must be selected for the navigation buttons to function.

Back: If selected, the **Back** button is included in the help window.

Forward: If selected, the **Forward** button is included in the help window.

Stop: If selected, the **Stop** button is included in the help window.

Refresh: If selected, the **Refresh** button is included in the help window.

Home: If selected, the **Home** button is included in the help window.

Print: If selected, the **Print** button is included in the help window.

Options: If selected, the **Options** button is included in the help window.

Jump1: If selected, the **Jump1** button is included in the help window. The caption and URL for the button are specified in the **Jump1 Caption** and **URL** fields.

Caption: The text that will be displayed on the Jump1 button.

URL: The URL of the Jump1 button.

Jump2: If selected, the **Jump2** button is included in the help window. The caption and URL for the button are specified in the **Jump2 Caption** and **URL** fields.

Caption: The text that will be displayed on the Jump2 button.

URL: The URL of the Jump2 button.

Navigation Pane

Width: Defines the width of the navigation pane of the help file in pixels.

Show Search tab: If selected, the **Search** tab is included in the help window.

Use advanced search: If selected, adds additional functionality to the **Search** tab.

Favorites tab: If selected, the **Favorites** tab is included in the help window.

Hidden by default: If selected, the navigation pane for the help file is hidden by default.

Hide when window deactivated: If selected, the HTML help navigation pane will be minimized when HTML help is not the active window.

Auto-synchronize contents: If selected, the heading or topic in the HTML help table of contents is automatically synchronized with the topic in the Topic pane.

Manual Window

The manual target is not an online Help format, and has no windows.

Help 2.0 Window

The size and position of windows in the Help 2.0 target (used for developing help for Microsoft Visual Studio) cannot be customized.

JavaHelp Window

Use the following options to configure your JavaHelp windows. Unavailable options will be grayed out.

Basic

Name: The name of the window. It is read-only; if you'd like to create a new window, Add one in the panel on the left. See Setting the Help Window Display (on page 176) for more details.

Position

Left: Determines the location of the left edge of the help window. Works in conjunction with the Top field to determine the position of the help window on the screen.

Top: Determines the location of the top edge of the help window. Works in conjunction with the Left field to determine the position of the help window on the screen.

Width: Determines the width of the help window.

Height: Determines the height of the help window.

Navigation Pane

Show Contents tab: If selected, the **Contents** tab is included in the help window.

Show Index tab: If selected, the **Index** tab is included in the help window.

Search tab: If selected, the **Search** tab is included in the help window.

Show Favorites tab: If selected, the **Favorites** tab is included in the help window.

Defining Character/Paragraph Styles and Topic Types

In Doc-To-Help, Styles can control the look of your output, as well as the behavior. See Guide to Templates and Styles (on page 5) for more information.

Doc-To-Help includes many predefined Styles. The **Project Styles** dialog box is used to modify these **character styles**, **paragraph styles**, and **topic types**, or to add new styles to the project. If you create a new style and want it to employ a special help authoring behavior, you must define those behaviors in this dialog box. Styles that are created strictly for formatting do not need to be added here.

Character Styles are used 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 project. Character styles allow you to create these types of hot spots and more using Doc-To-Help Markup Language (D2HML) (on page 210).

Paragraph Styles are used to assign specific behaviors to entire paragraphs. For example, Doc-To-Help built-in Heading styles specify the structure and hierarchy of your topics in Help Targets and the generation of automatic subtopic links for them. They also control the structure of the automatically created Table of Contents for both online and manual Targets.

A **Topic Type** is a named collection of topic attributes: what window the Help topic appears in, the navigation for the topic, whether it is automatically added to the index, etc. Paragraph Styles can have Topic Types assigned to them. If they do, the Topic Type properties override any duplicate Paragraph Style properties. If a Paragraph Style property is overridden in this way, it will be grayed out in the **Project Styles** dialog box.

A Topic Type can also be used to customize an individual topic, overriding the style properties. For more information, see Viewing/Changing Topic Properties on page 225.

To open the Project Styles dialog box

1. Open the **Project** tab.
2. From the **Project** ribbon group, click the **Project Styles** or **Topic Types** button. The **Project Styles** dialog box will open.

To add a new style or topic type

1. In the **Project Styles** dialog box, choose **Character Styles**, **Paragraph Styles**, or **Topic Types** from the panel on the right.

2. Click on the **Add New Style** button in the left panel.

   The style or topic will be added to the list with the default name **NewStyle** or **NewTopicType**. Change the name by editing the **Name** field in the **Basic** section, then clicking anywhere in the dialog box to change it in the list on the left.

   When you add a new paragraph style, by default, its **Type** is Body Text. This should be changed to the appropriate **Type** to activate it. See the **Type** field in Paragraph Styles (on page 184) for more information on each Type.

To edit an existing style or topic type

1. In the **Project Styles** dialog box, choose the style or topic type you'd like to edit from the panel on the left.
2. Make the desired changes to the style or topic type and click **OK**. If you would like to edit the name of the window, edit the **Name** field in the **Basic** section.

To remove a style, select it and choose **Remove Style**. If a style has a "padlock" icon next to it in the **Project Styles** dialog box, that style is built-in and cannot be deleted.

More on the Project ribbon group on page 106.

Built-in Styles and Topic Types

Character Styles	Paragraph Styles	Topic Types
C1H Conditional	Heading 1	Conceptual
C1H Contents Title	Heading 2	Contents
C1H Context ID	Heading 3	Glossary of Terms
C1H Dropdown Text	Heading 4	Glossary Term Definition
C1H Expand Text	Heading 5	Margin Note
C1H Group	MidTopic	Procedural
C1H Group Invisible	RelatedHead	Sub-Contents
C1H Group Link	WhatsThis	What's This
C1H Index	Glossary Heading	
C1H Index Invisible	Glossary Heading (no auto links)	
C1H Inline Dropdown		
C1H Inline Expand		
C1H Inline Popup		
C1H Jump		
C1H Keyword Link		
C1H Link Tag		
C1H Link Tag Invisible		
C1H Manual		
C1H Online		
C1H Popup		
C1H Popup Text		
C1H Topic Properties		
C1H Variable		

Character Styles

Use the following options to configure character styles. Unavailable options will be grayed out.

Basic

Name: The name of the style or topic type.

HTML name: Defines the name to be used to identify the style as a cascading style sheet style in HTML source documents. Every style specified in a Doc-To-Help project has two names: its name and its HTML name that identifies it as a CSS style, or selector. When you use a style in an HTML document, make sure you use its HTML name. According to the standard CSS rules, the HTML name can have one of three forms:

- **<stylename>** — The style can be used with any HTML tag.
- **<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.
- **<tag>.<stylename>** — The style can be used only with the specified tag; it is ignored if used in other tags.

Display

Affects appearance: If selected, this style defines the appearance of the text (font, color, etc.) in the target help file. If cleared, the style is only used to create a hotspot or keyword and does not affect target appearance. Character styles with this check box selected can be used to specify the appearance of links generated by other means. For example, a style with **Affect Appearance** selected and a **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.

Include page number in reference: In a manual target, if this check box is selected, the page number of the referenced topic will be placed next to the text formatted with this style. This option is only available for Jump and Popup links.

Hidden: If selected, any text formatted with this paragraph or character style is omitted from the help target. If a topic heading is formatted with this style, the whole topic is omitted from the help target.

Replacement: In certain cases, returns a string that overrides the generated default text. For paragraph styles and topic types with the **Auto next** check box selected, this string is used instead of the topic title next to the generated button. For character styles with the **Include page number in reference** check box selected, 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.

Window: The name of the window in which topics formatted in this style are displayed.

Behavior

Auto Index: If selected, index keywords are automatically created from text formatted with this character style and associated with the topic that contains the text.

Multi link: If selected, each occurrence of a "jump" character style (in a given topic) generates a link. If cleared, only the first occurrence generates a link and all other formatted occurrences are skipped.

Behavior: Determines the help authoring behavior associated with this character style at compile time.

- **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
- **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
- **Variable** — Variable hot spot

Link type: Determines whether a link hot spot is a jump or a pop-up.

- **Jump** — Topic jump hot spot.
- **Popup** — Topic pop-up hot spot.

Script: Displays the script to be run during compilation whenever text formatted with this style is encountered.

Condition

Platforms: Sets a platform-based condition for the selected style. The style will be included in all the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets: Sets a target-based condition for the selected style. The style will be included in all the target(s) selected.

Attributes: Sets an attribute-based condition for the selected style. The style will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > Attributes button) to create custom attributes.

Paragraph Styles

Use the following options to configure paragraph styles. Unavailable options will be grayed out.

Basic

Name: The name of the style or topic type.

HTML name: Defines the name to be used to identify the style as a cascading style sheet style in HTML source documents. Every style specified in a Doc-To-Help project has two names: its name and its HTML name that identifies it as a CSS style, or selector. When you use a style in an HTML document, make sure you use its HTML name. According to the standard CSS rules, the HTML name can have one of three forms:

- **<stylename>** — The style can be used with any HTML tag.

- **<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.
- **<tag>.<stylename>** — The style can be used only with the specified tag; it is ignored if used in other tags.

Type: Determines the help authoring behavior associated with a paragraph style at compile time.

- **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.

Hidden: If selected, any text formatted with this paragraph or character style is omitted from the help target. If a topic heading is formatted with this style, the whole topic is omitted from the help target.

Auto glossary links: If selected, during compilation Doc-To-Help examines the project documents for text that matches glossary entries (in the glossary document) and converts these matches into hyperlinks. When the user clicks on one of the hyperlinks, a DHTML pop-up opens containing the corresponding glossary entry description.

Online only: If selected, paragraphs formatted with this style are omitted from the printed manual target only. This setting has no effect on online Help targets.

Preformatted: This check box affects how text formatted with this style appears in HTML-based targets. It is especially useful for source code examples where you don't want text wrap and white space adjustments. If selected, the text formatted with this style is enclosed in a <pre> tag in the resulting HTML. This ensures that there is no text wrap and white spaces are treated literally. See HTML documentation for more information about the <pre> tag. Note: This check box can only be selected for paragraph styles where **Type**=Body text.

Topic type: Specifies the named set of display, navigation, and indexing characteristics to be associated with this style (such as what window the help topic appears in, how the help topic is accessed, and whether it gets a map number). There are eight pre-defined Topic Types in the Styles dialog box (on page 181) that may be edited, or you can create new ones.

Navigation

Auto Subtopic Links: If selected, subtopic buttons are displayed for paragraph styles with numeric outline levels greater than the parent level.

Auto link: When selected, creates a unique link tag for topics formatted with this paragraph style or topic type. Link tags created in this manner are identical to the topic title except that spaces, hyphens and periods are converted into underscores.

Auto navigate: If selected, this paragraph style or topic types will be included in the navigation sequence.

Auto next: If selected, enables the next topic button for this paragraph style or topic type. A next topic button is displayed at the end of each associated topic that links it to the next topic.

Explicit access: If selected, defines this 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.

Mid-topic: If selected, defines this paragraph style or 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, you can create jump functionality within a lengthy topic.

Display

Contents only: If this check box is selected for this paragraph style or topic type, the topics formatted with this paragraph style or topic type are omitted from the help target, but are used as book titles in the help contents.

Non scrolling: If this check box is selected for this paragraph style or topic type, any topic headings formatted with this paragraph style or topic type will appear in the non-scrolling region of the help window. (Only applies to WinHelp targets.)

Popup: If this check box is selected for this paragraph style or topic type, any topic formatted with this paragraph style or topic type will appear in a pop-up window on top of the help window.

Suppress empty topics: If selected, omits empty topics (those with a title and no text) defined with this paragraph style or topic type from online help. These topics will still be included in the TOC. This check box can be selected for a Contents topic type that is commonly used for the top-level style, such as Heading 1. If this style usually contains no text, it may be preferable that these empty topics are not shown in the help window when the user selects the corresponding book title in the TOC.

Untitled: If selected, the topic title is not displayed in the help window.

Replacement: In certain cases, returns a string that overrides the generated default text. For paragraph styles and topic types with the **Auto next** check box selected, this string is used instead of the topic title next to the generated button. For character styles with the **Include page number in reference** check box selected, 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.

Window: The name of the window in which topics formatted in this style are displayed.

Behavior

Auto context ID: If selected, Context ID’s are automatically created for all topics using this paragraph style or topic type.

Auto keyword: If selected, index keywords will automatically be created from topic titles formatted with this paragraph style or topic type. The keywords will be associated with the appropriate topic.

Script: Displays the script to be run during compilation whenever text formatted with this style is encountered.

Topic Types

Use the following options to configure topic types. Unavailable options will be grayed out.

Basic

Name: The name of the style or topic type.

Navigation

Auto Subtopic Links: If selected, subtopic buttons are displayed for paragraph styles with numeric outline levels greater than the parent level.

Auto link: When selected, creates a unique link tag for topics formatted with this paragraph style or topic type. Link tags created in this manner are identical to the topic title except that spaces, hyphens and periods are converted into underscores.

Auto navigate: If selected, this paragraph style or topic types will be included in the navigation sequence.

Auto next: If selected, enables the next topic button for this paragraph style or topic type. A next topic button is displayed at the end of each associated topic that links it to the next topic.

Explicit access: If selected, defines this 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.

Mid-topic: If selected, defines this paragraph style or 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, you can create jump functionality within a lengthy topic.

Display

Contents only: If this check box is selected for this paragraph style or topic type, the topics formatted with this paragraph style or topic type are omitted from the help target, but are used as book titles in the help contents.

Non scrolling: If this check box is selected for this paragraph style or topic type, any topic headings formatted with this paragraph style or topic type will appear in the non-scrolling region of the help window. (Only applies to WinHelp targets.)

Popup: If this check box is selected for this paragraph style or topic type, any topic formatted with this paragraph style or topic type will appear in a pop-up window on top of the help window.

Suppress empty topics: If selected, omits empty topics (those with a title and no text) defined with this paragraph style or topic type from online help. These topics will still be included in the TOC. This check box can be selected for a Contents topic type that is commonly used for the top-level style, such as Heading 1. If this style usually contains no text, it may be preferable that these empty topics are not shown in the help window when the user selects the corresponding book title in the TOC.

Untitled: If selected, the topic title is not displayed in the help window.

Replacement: In certain cases, returns a string that overrides the generated default text. For paragraph styles and topic types with the **Auto next** check box selected, this string is used instead of the topic title next to the generated button. For character styles with the **Include page number in reference** check box selected, 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.

Window: The name of the window in which topics formatted in this style are displayed.

Behavior

Auto context ID: If selected, Context ID’s are automatically created for all topics using this paragraph style or topic type.

Auto keyword: If selected, index keywords will automatically be created from topic titles formatted with this paragraph style or topic type. The keywords will be associated with the appropriate topic.

Script: Displays the script to be run during compilation whenever text formatted with this style is encountered.

Working with Scripts

The **Scripts** dialog box is used to edit existing scripts, as well as create new ones.

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. Scripts are executed whenever Doc-To-Help encounters a topic or a hot spot defined by a scripted style.

For complete VBScript documentation, visit the Microsoft Scripting Technologies site (http://msdn2.microsoft.com/en-us/library/ms950396.aspx).

To open the Scripts dialog box

1. Open the **Project** tab.
2. From the **Project** ribbon group, click the **Project Styles** or **Topic Types** button. The **Project Styles** dialog box will open.
3. In the Behavior group, click the **Script** drop-down menu and choose **(Edit scripts …)**. The **Scripts** dialog box will open.

To add a new script

1. In the **Scripts** dialog box, click on the **Add New Script** button in the left panel.

   The script will be added to the list with the default name **NewScript**. Change the name by editing the **Name** field in the **Properties** section, then clicking anywhere in the dialog box to change it in the list on the left.
2. Enter the script code in the **Code** box. For examples of code, click on the existing D2HGlossaryRef and D2HGlossaryTerm scripts.
3. Click **OK**.

To edit an existing script

1. In the **Scripts** dialog box, choose the script you'd like to edit from the panel on the left.
2. Make the desired changes to the script and click **OK**. If you would like to edit the name of the script, edit the **Name** field in the **Properties** section.

Suppress Default Script check box – Specifies whether a user-defined script augments a default Character Style behavior or completely overrides it. If a Script is specified in the Scripts dialog box, and the Character Style already has a default D2HML behavior (noted in the Behavior dropdown of the Project Styles dialog box), you can use this option to completely override the default behavior of the style. By default, the D2HML behavior is not overridden, which means that your script will be executed after the default action defined by the style Behavior is performed. If you select this check box, your script will be the only action triggered by the style. See Defining Character/Paragraph Styles and Topic Types (on page 181) for more on styles.

Editing a CSS

The **Style List** window is used to modify/create the styles within your project .css files.

Please see HTML and XHTML File Style Sheets (on page 10) for an explanation of Doc-To-Help's default style sheets, how Source and Target style sheets work together, and where they are stored.

To open the Style List window

For a quick overview of the Style List, see Style List Window Tour on page 192.

1. Open the **Home** tab in Doc-To-Help.
2. To edit **Target** styles:
   - From the **Target Design** ribbon group, click the **Target CSS** drop-down arrow.
   - Choose **Edit CSS**. The **Style List** window will open for the selected CSS.
3. To edit **Source** styles:
   - From the **Source** ribbon group, click the **Source CSS** drop-down arrow.
   - Choose **Edit CSS**. The **Style List** window will open for the selected CSS.

   or

   - Open an XHTML document from the **Documents** pane. In the **Editor** tab, click the **Style List** button in the **Styles** ribbon group.

The **Style List** will open in **Simple Mode**. If you click the **Advanced Mode** button , the **Style List** will display the inheritance of the styles. The **Source** and **Target CSSs** will differ in levels of inheritance. The **Target CSS** will have additional levels because it inherits the Source CSSs. This is a time-saver, since a change to the **Source CSS** is automatically saved to the **Target CSS**.

To add a CSS to the **Source CSS** or **Target CSS** drop-downs, choose **Add CSS**. To remove a CSS from the drop-downs, choose **Remove CSS**. Removing a CSS in this manner will only remove it from the drop-down list; not delete it permanently.

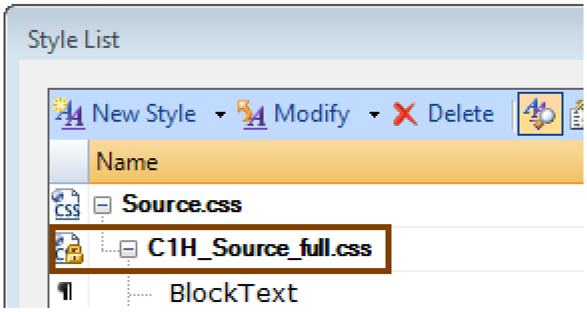
To modify a Style

If you modify a read-only style, it will be saved as a derived style; a style that is based on the original (the original will remain untouched). You can also modify any editable style.

1. In the **Style List** window (**Simple Mode**), choose the style you'd like to edit from the list.
2. Click the **Modify** button . The **Style Formatting** dialog box will open.
3. Edit the **Font**, **Background**, **Border**, **Box**, **Paragraph**, and/or **Position** via the tabbed windows.
4. Click **OK**. A WYSIWYG view of the style will be displayed in the **Name** column.

   If you are in **Advanced Mode**, the **Modify** button will look like this .

   In **Advanced Mode**, you can also view how Doc-To-Help manages styles. When you modify a style in a read-only style sheet, Doc-To-Help will copy the style and save it to the editable version of the style sheet. For example, C1H_Source_full.css is a default style sheet and is locked (indicated by the icon below). If you choose to modify the BlockText style, Doc-To-Help will create a new BlockText style and save it in the editable Source.css style sheet.



See HTML and XHTML File Style Sheets (on page 10) for more information on Doc-To-Help Style Sheets.

To create a new Style

1. In the **Style List** window (either mode), click the **New Style** button . The **Style Formatting** dialog box will open.
2. Enter a **Style name**. This will be the name of the style in the CSS. The name cannot include spaces or punctuation.
3. From the **Style Type** drop-down box on the **General** tab, choose **Character** or **Paragraph**.
4. If you would like to use an existing style as a starting point, select a **Base style**. Please note that the new style will appear as a "child" of this style in the **Style List** dialog box.

   Read-only styles will not appear in this list, if you would like to create a style based on one (a derived style), see **"To modify a Style"** above.
5. Edit the **Font**, **Background**, **Border**, **Box**, **Paragraph**, and/or **Position** via the tabbed windows.
6. Click **OK**. The new **Style name** will appear in the **Style List** dialog box.

To delete a Style

1. In the **Style List** window (either mode), choose the style you'd like to delete from the list.
2. Click the **Delete** button .
3. Click **OK**.

Read-only styles and the styles derived from them cannot be deleted.

To add a CSS

1. In the **Style List** window (**Advanced Mode**), click the **Add CSS** button . The **Add Style Sheet** dialog box will open.
2. Choose to add a new or existing style sheet, using the radio button.

   If you choose **Existing**, click the **Browse** button to navigate to the style sheet.

   If you choose **New**, give the CSS a name, and select a base CSS if applicable.

   You can choose to designate either type as read-only with the check box.
3. Click **OK**.

To modify a CSS

1. In the **Style List** window (**Advanced Mode**), choose the style sheet you would like to modify from the list.
2. Click the **Modify CSS** button . The **Style Sheet Properties** dialog box will open.
3. In this dialog box, you can change the **Base style sheet** of the CSS — this is the style sheet the CSS inherits from. You can also make a style sheet editable/non-editable using the **Read-only** check box.

   Please note that you should not change default Doc-To-Help read-only style sheets (for example, C1H_Source_full.css) from read-only to editable since these CSSs are inherited by the main CSS.
4. Click **OK**.

Style List Window Tour

The **Style List** window displays the styles in your project, and can be used to edit and apply them. This window can be opened several ways — see Editing a CSS (on page 189) to learn how to open the **Style List** window.

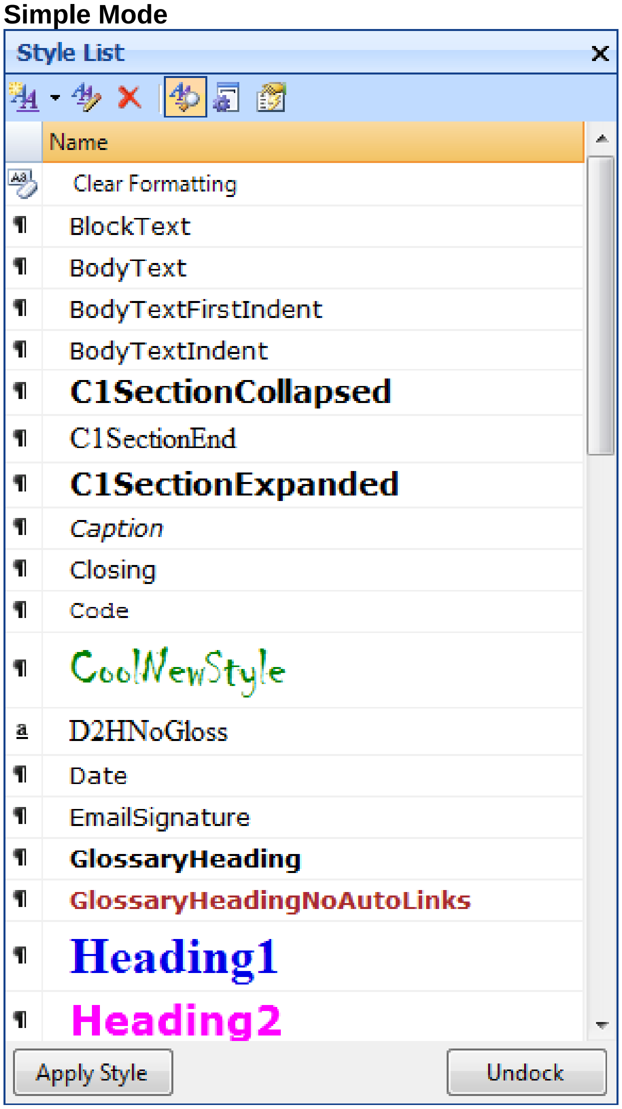
For more information on editing styles, see Editing a CSS (on page 189). If your project does not contain XHTML (.xml) source files, you can edit the styles with the **Style List**, but cannot apply them with it. The styles will be applied using the appropriate HTML editor.

For information on applying styles using the **Styles List**, see Applying Styles in the XHTML Editor on page 119.

The **Style List** can be a dialog box or it can be docked. If you have opened it by clicking the **Apply Style** button in the **Editor** tab (.xml documents only) it will include buttons on the bottom that you can use to apply styles in the **XHTML Editor** window and dock/undock it. If you have opened the **Style List** from the **Home** tab, **Target CSS** or **Source CSS** drop-downs (choose **Edit CSS**), it will be a dialog box.

By default, the **Style List** opens in **Simple Mode**.

Simple Mode



Style List Toolbar

	New Style — Create an original style or tag
	Modify Style — Create a style based on an existing one (a derived style). If the style has a behavior associated with it, that behavior will be retained.
	Delete Style — Delete the chosen style. Read-only styles can not be deleted.
	Preview Styles — Toggle the Style List WYSIWYG view on and off.
	Advanced Mode — Toggle the Style List Advanced Mode view on and off.

Icons

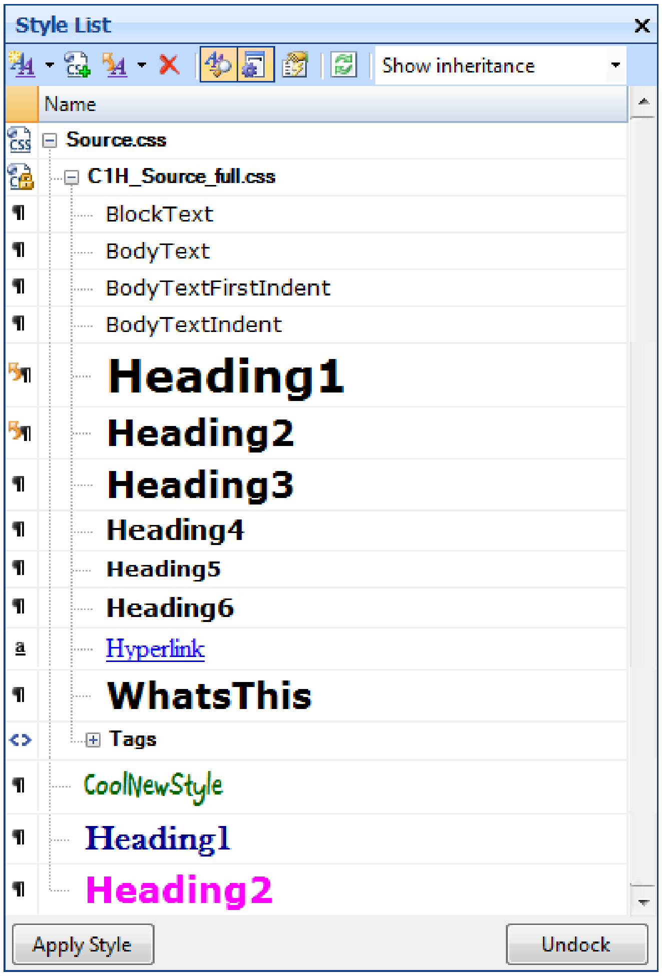
¶	Indicates a paragraph style.
a	Indicates a character style.

Creating Styles

Styles can be created by using the **Style List** toolbar, or by using the right-click menu.

When you click the **Advanced Mode** button in the **Style List** toolbar, the display will changed to illustrate the inheritance of Styles and provide a few additional options.

Advanced Mode



Style List Toolbar

New Style — Create an original style or tag

Add Style Sheet — Add a Style Sheet to the list.

Modify Style — Create a style based on an existing one (a derived style). If the style has a behavior associated with it, that behavior will be retained. (If you select a .css in the list, this will change to the **Modify CSS** button, where you can modify the properties of that CSS.)

Delete Style — Delete the chosen style. Read-only styles can not be deleted.

Preview Styles — Toggle the Style List WYSIWYG view on and off.

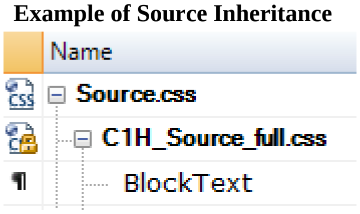
Customize Columns — Click to choose the columns you would like to display in the Style List.

Reload — If you edit one of the project style sheets in another editor, use this button to reload the file(s) and refresh the view.

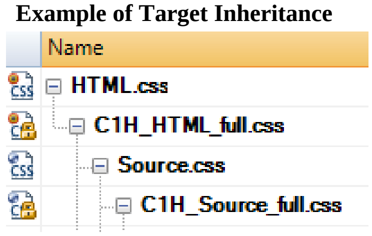
View — Choose to view styles by their CSS inheritance, or in a view that allows you to sort the styles.

Inheritance

Example of Source Inheritance



Example of Target Inheritance



By default, the css files are shown in "inheritance" view. The style sheets can be expanded and collapsed for easy viewing using the "+" and "-" buttons.

Icons

This icon indicates the CSS is unlocked and styles are editable.

This icon indicates the CSS is read-only and not editable (but you can create derived styles from it).

Indicates a paragraph style.

Indicates a character style.

An arrow next to a paragraph style or character style icon indicates that a derived style has been created from this style.

Creating Styles

Styles can be created by using the **Style List** toolbar, or by using the right-click menu.

How Style Sheets are inherited

When you create a project that uses HTML and/or XHTML source documents in Doc-To-Help, you choose Source and Target style sheets for them. These CSS files are included with Doc-To-Help. See HTML and XHTML File Style Sheets (on page 10) to learn more about the available style sheets and how they work together.

When you create a new project in Doc-To-Help 2009, the project style sheets chosen when creating the project will become read-only in the **Style List**, while all your changes will be saved to a separate file, using a truncated file name (for example, changes to Source style sheet “C1H_Source_Full.css” will be saved to “Source.css"; changes to Target style sheet "C1H_HTML_Full.css" will be saved to "HTML.css"). This keeps the original file untouched.

If you go to the **Home** tab and choose the **Target CSS** or **Source CSS**, you will see the style sheets being used by your project. You will notice that the truncated names are listed, because they are the editable CSS files. In the **Style List** inheritance view, you will see that the editable file has inherited the original CSS, and the original CSS is marked as "read-only." In the **Style List** pictured above, Source.css inherits from C1H_Source_Full.css. Target style sheets have a few more layers, but work the same way. You will notice that the Source style sheets also appear in the Target style sheet list. This is because any changes to the Source style sheets are inherited by the Target — which means a change to the Source style sheet is automatically changed in the Target for you, so you don't have to make changes in two places.

What happens when you create a style

In the **Style List**, you can create new Styles, or you can create derived styles — which are styles based on existing styles (usually those in the inherited "read-only" style sheets). Those styles are saved to the editable CSS file. In the example above, you will notice that there are two styles, **Heading1** and **Heading2**, stored in the Source.css file. These styles were derived from (based on) the existing Heading1 and Heading2 styles in C1H_Source_Full.css. Note that the original Heading1 and Heading2 styles have the "derived" icon next to them. A derived style retains the behavior of the original. If you need to change a style that has a specific behavior in Doc-To-Help, see Defining Character/Paragraph Styles and Topic Types (on page 181) for more information. Any new styles created will also be stored in Source.css (in the example, see **CoolNewStyle**).

CSS files can be found in the following default directories:

- All of your project .css files are saved to the \\(project folder)\CSSFiles folder.
- All available Doc-To-Help .css files are stored at \\Program Files\ComponentOne\DocToHelp\DefaultCSSFiles.

Creating Variables

Variables allow you to write content once, and manage it in one place for reuse across your project.

Text Variables may be used for any amount of unformatted text or use **Rich Content Variables** for blocks of formatted content. Both are created in the Variables window (on page 114).

Text Variables can be assigned conditions, providing even more flexibility for their use. If you’d like, you can create a single variable, and assign multiple text values and conditions for it. This allows you to insert a variable once, and have different text be used in each of your conditions. **Rich Content Variables** can include text that has been conditionalized.

Variables make it possible to:

- Change text once and automatically update it everywhere.
- Ignore source issues — use any variable source (HTML, XHTML, or Word) in any source document. (**Rich Text Variables** are stored in source documents.)

Examples of **Text Variables** include:

- Product or company name
- Frequently used descriptions
- Addresses
- Copyright notices

Examples of **Rich Content Variables** include:

- Tables
- Images or other media
- Formatted company names (i.e., **ComponentOne**)
- Entire topics

To open the Variables window

From the **Project** tab, Project ribbon group (on page 106), choose the **Variables** toolbar button. The Variables window (on page 114) will open.

To create a Text Variable

1. Open the **Variables** window.
2. In the **Text Variables** area, click on the **Add New Variable** toolbar button. An editable field named <New Variable> will appear in the **Name** column.
3. Enter a name for the variable, then double-click <Variable text> in the **Text** column to enter the text (one word or more).

   If you'd like, double-click in the **Condition** column to choose a **Platform**, **Target**, or **Attribute** condition for the variable.

   - **Platforms** — Set a platform-based condition for the variable. The text will be included in all of the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.
   - **Targets** — Set a target-based condition for the variable. The text will be included in all the target(s) selected.
   - **Attributes** — Set an attribute-based condition for the variable. The text will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > **Attributes** button) to create custom attributes.

If you would like to assign more than one condition to this variable, select its name and click the **Add Text with Condition** toolbar button. An additional <variable text> field will open. Enter the appropriate text, then double-click in the **Condition** column to choose a condition. Click the **Add Text with Condition** toolbar button to add another text/condition to the variable. One **Text Variable** can have multiple unique outputs.

To create Rich Content Variables

1. Open the **Variables** window.
2. In the **Rich Content Variable** area, click on the **Create New Document** button. Choose **XHTML**, **HTML**, or **Word Document** from the drop-down list. The **Save New Document As** dialog box will open. Enter the document **Name** and click **Save** to add it to your project. (Please note: if using Word 2007, note that the desired file extension is displayed — .doc or .docx)

You can also add an existing variables document (or convert existing HTML or Word documents to XHTML). First, copy the document(s) to the appropriate folder in your Doc-To-Help project. Source documents should be placed in the **Documents** folder. Then click the **Add Existing Documents** button.

3. Double-click on the document name in the **Variable** window to open it. The **Variables** document will have a table with two columns in it.
4. Enter variable name in the column on the left (avoid spaces), and the variable content in the column on the right. Apply styles as desired, including conditional text.
5. Save the document.

   Please note that you can define a variable in any document type, and use it in any source document. **Tip:** If you are authoring in Word, but would like to add HTML-only elements to your online Help, create your Rich Content Variable source documents in HTML or XHTML.

To insert variables in documents

Variables are inserted in Microsoft® Word, Microsoft® FrontPage®, and Adobe® Dreamweaver® documents using the **Variable** button in the Doc-To-Help toolbar or ribbon (on page 115).

Variables are inserted in the XHTML Editor window (on page 113) using the **Variable** button in the Insert tab (on page 103). See Inserting a Variable on page 221.

Setting Project Properties

The **Project Settings** dialog box is used to set the universal properties for the entire project, regardless of target.

To open the Project Settings dialog box

1. Open the **Project** tab.
2. Click the **Project** ribbon group dialog box launcher. The **Project Settings** dialog box will open.

More on the Project ribbon group on page 106.

Documents

Default CSS: The cascading style sheet attached to new source HTML documents when they are added to the project.

Default template: The document template attached to new source Word documents when they are added to the project.

Document folder(s): The name of the folder where all Word, HTML, and XHTML source documents will reside. The default folder name is Documents. It is strongly recommended that all source documents are stored inside this folder, because it helps to keep the project files organized and because doing this will ensure that links to files of any kind (images, multimedia, other HTML files) are preserved.

Output options

Keep outline numbers: When selected, outline numbers are included as part of topic titles and are present in help targets. This setting only affects the RTF files generated from source documents; it does not affect the source documents themselves. Modifying this setting for an individual document will override the settings for that document only.

Keep page breaks: When selected**,** retains the page break characters in the source documents during compilation of a Printed Manual Help target. Clear this check box to discard page break characters. Modifying this setting for an individual document will override the settings for that document only.

Adjust left indent: Controls whether paragraph indentation is adjusted to account for wide margins when building online help. By default, this check box is selected to accommodate the standard Doc-To-Help templates. Clear this check box if you are using custom templates and want to preserve the indentation used in your source documents.

Plain text popups: If selected, generates a plain text only version of the help file for context-sensitive help topics. Modifying this setting for an individual document will override the settings for that document only.

URL mode: Specifies the rules used for naming the .htm files generated for topics. These rules apply when the topic URL is generated automatically. By changing this value, you can optionally apply the new rules to all existing topics. You can modify the URL of a topic later, manually overriding the default rules.

- **Full Title** — File name contains all letters and digits from the topic title, even non-ASCII national alphabet letters.
- **ASCII Only** — Non-ASCII characters are removed from the file name. File name complies with URL standard.
- **Internal Topic IDs** — File name is formed from the numeric topic ID.

'On Page' text: Specifies the default text used when completing a cross-reference in a printed manual target.

Is modular hub project: If selected, the generated help file can dynamically load the contents of other help files, if present. Only WinHelp, HTML Help and NetHelp platforms can support modular hub projects. For WinHelp and HTML Help targets, to specify a component help file in a modular hub project, create a placeholder topic, then set the Module file and Contents file fields of that topic (in the Topic Properties dialog box on page 225) 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 Module file field of that topic to the component filename.

Context IDs

Generate context IDs automatically: Determines whether a unique Context ID (map number) is generated for each topic. For Context IDs to be generated, the **Auto context ID** check box must be selected for each relevant **Topic Type** or **Paragraph Style** in the Styles dialog box (on page 181).

ID offset: The value specified is added to the automatically generated map numbers to prevent numbering conflicts in modular Help systems.

Advanced

PlugIn folder(s): The folder plugin documents — for example, Sandcastle XML documents — are stored in. The default folder name is **XMLDocuments**, but remains empty until at least one plugin document is created in the project. Plugin documents are added from the Project tab, PlugIns ribbon group (on page 107).

Media folder(s): The name of the folder where graphic files (images, videos, audio, etc.) should reside. The default folder name is **Media**. It is recommended that all media be stored inside this folder because it helps to keep the project files organized.

Choosing a Source Folder

The **Choose Source Folder** dialog box is used to set the default folder that your source documents are stored in. This dialog box is accessed from the Project Settings dialog box (on page 196) by clicking on the ellipsis button next to the **Document folder(s), Plugin folder(s), or Media folder(s)** field.

To add a source document folder

Click the **Add Folder** button at the top left. A **Browse** dialog box will open. Choose or create a folder and click **OK**.

To delete a folder, choose the folder in the window and click **Remove Folder**.

The default folder for Word, HTML, and XHTML source documents is **Documents**. This folder is automatically created when you create a project. It is strongly recommended that all source documents are stored inside this folder, because it helps to keep the project files organized and because doing this will ensure that links to files of any kind (images, multimedia, other HTML files) are preserved.

The default folder for XML source documents is **XMLDocuments**. This is the folder plugin documents — for example, Sandcastle XML documents — are stored in. This folder will remain empty until at least one plugin document is created in the project. Plugin documents are added from the **Project** tab, Plugins ribbon group (on page 107). See Documenting Your Class Library with Microsoft® Sandcastle (on page 265) for more information.

The default folder for images, video, etc. is **Media**. This folder is automatically created when you create a project. It is strongly recommended that all media files are stored inside this folder.

Compacting a Project File

Doc-To-Help includes a project compacting utility that can reduce the size of your project file. Doc-To-Help automatically checks and compacts your project when it is opened, or any time a noticeable increase in size is detected, but you can run it manually if you wish.

Note: All Doc-To-Help projects must be closed before using the compact utility.

To compact a Doc-To-Help project

1. Choose the Doc-To-Help button > Tools > Compact Project. The Open Doc-To-Help Project dialog box will open.
2. Choose the project (.d2h file) and click the **Open** button.
3. The project will be compacted. A message box will inform you when the process is complete.
4. Click **OK**.

Importing and Exporting Project Settings

Project Settings can be copied from one project to another using the **Import** or **Export Project Settings Wizard**. This can save significant time if you'd like to use the same settings for multiple projects.

Project Settings can be copied from another Doc-To-Help project, or from an XML file. The setting collections available for import/export are:

- Keywords
- Groups
- Attributes
- Windows
- Styles
- Scripts
- Style Sheets
- Help Targets
- Project Properties

To import project settings

1. Choose the Doc-To-Help button > Tools > Import Project Settings. The Project Settings Import Wizard will open.
2. Choose to import from an **XML** file or an existing **Doc-To-Help** project.
3. Browse to the XML file or Doc-To-Help project location.
4. Select the project settings you'd like to import. By default, all are selected.

   If you'd like to overwrite the existing settings in the current project (rather than add to them), select the **"Overwrite objects existing in the current project"** check box.
5. Click the **Import** button. Any issues with the import will be displayed.

To export project settings

1. Choose the Doc-To-Help button > Tools > Export Project Settings. The Project Settings Export Wizard will open.
2. Choose to export to an **XML** file or an existing **Doc-To-Help** project.
3. Browse to the XML file or Doc-To-Help project location.
4. Select the project settings you'd like to export. By default, all are selected.

   If exporting to another Doc-To-Help project, you can overwrite the existing settings in the receiving project (rather than add to them), by selecting the **"Overwrite objects existing in the current project"** check box.
5. Click the **Export** button. Any issues with the export will be displayed.

Storing Project Settings in XML files

To keep your Doc-To-Help project settings in an XML file that you can modify and use in any project, you must first export your project settings to XML using the **Project Settings Export Wizard**. Any changes you make manually to the .xml file must conform to the XML schema Doc-To-Help uses to validate the file. This schema file is named **c1d2h.xsd** and is located by default in the C:\Program Files\ComponentOne\DocToHelp directory. Storing project settings in an XML file makes it possible to automatically modify Doc-To-Help projects programmatically.

You can also import settings from another Doc-To-Help project using the **New Project Wizard**. See Creating and Converting Projects on page 139.

Implementing Context Sensitive Help

It is possible to map specific Help topics to your software application based on the user's location in the interface. This Help is most commonly accessed using a dialog box Help button or icon, or by pressing the F1 button to open a Help window. "What's this?" Help is also context sensitive, and is accessed by clicking a "question mark" icon in a dialog box and then selecting a field or other object in the dialog box. "What's this?" Help then appears in a popup.

Topics are mapped to the software application using **Context IDs**, which can be specified by the Help Developer or the Software Developer. If the Context IDs will be supplied by the Help Developer, Doc-To-Help can be set (if desired) to automatically generate Context IDs and assign them to specified Topic Types or Paragraph Styles. If the Context IDs will be supplied by the Software Developer, the Help Developer can easily map them to topics in HTML Help, NetHelp, or WinHelp Targets using the steps below.

If your Context IDs were assigned in Doc-To-Help, the appropriate Context ID file type for your software application will be automatically generated and stored in your project. You can also export the IDs to Microsoft Excel or a Text (.txt) file. See Printing and Exporting the Topic List (on page 227) for details.

Doc-To-Help has context sensitive, dynamically updating Help embedded in its interface and dialog boxes. This was created using the ComponentOne DynamicHelp control. Your Doc-To-Help purchase includes a license for this Visual Studio .NET control, which is part of ComponentOne Studio for WinForms. To use DynamicHelp, please download Studio for WinForms from http://www.componentone.com/SuperProducts/StudioWinForms/ and use the serial number you should have received with your Doc-To-Help purchase to unlock it. Note that the rest of the controls included in Studio for WinForms will be available in evaluation mode. If you are mapping Help topics using this control, using Context IDs is not required. See Mapping Context Sensitive Help using Dynamic Help Authoring Mode on page 206.

To automatically generate Context IDs

1. Open the **Project Styles** dialog box. (See Defining Character/Paragraph Styles and Topic Types (on page 181) for more information.)
2. Select a **Topic Type** or **Paragraph Style** that you would like Auto Context IDs to be assigned to.
3. Select the **Auto Context ID** check box.
4. Repeat for all relevant **Topic Types** and **Paragraph Styles**. (You may want to start out by selecting the Heading 1, Heading 2, and Heading 3 **Paragraph Styles** only.)
5. Click **OK**.
6. Open the **Project Settings** dialog box. See Setting Project Properties (on page 196) for more information.
7. Select the **Auto Context ID** check box.

   If you are creating a Modular Help project (a project that contains multiple Help projects) you may want to assign an **ID offset** to avoid duplicate IDs across your projects. See Modifying Context IDs (on page 262) for more information.
8. Click **OK**.

Context IDs will be generated for all selected **Topic Types** and **Paragraph Styles**. They can be viewed in the Topics window (on page 112).

To view/edit/assign a topic's Context ID for HTMLHelp, NetHelp, and WinHelp projects

1. Open the Topics window (on page 112).
2. Right-click on a topic. Choose **Properties** from the menu. The **Topic Properties** dialog box will open.
3. If a **Context ID** has already been assigned to the topic, that ID will appear in the **Context ID** field. You may edit this field or assign a new ID. Even auto-assigned context IDs may be edited.

   If you would like to assign more than one Context ID to a topic, enter them with plus signs separating them. For example 45 + 46 + 47.

Note: You can assign multiple context IDs to a single topic, but you cannot assign the same ID to multiple topics.

4. Click **OK**.

File locations for Context IDs created in Doc-To-Help

The Context ID files are stored in your project, in the appropriate Target folder. The appropriate file should be given to Software Development to implement context-sensitivity. The folder names provided below are the defaults; if you have changed the Target folder names in your project they will be saved to those folders instead.

HTML Help target
Project folder name: **HTMLHelp**
File name: projectname.h

NetHelp Help target
Project folder name: **NetHelp**
File name: _contextIds.js
Additional file needed: D2H_ctxt.*, located in \\Program Files\ComponentOne\DocToHelp\Context-Sensitive Help

WinHelp Help target
Project folder name: **Help**
File name: projectname.h

JavaHelp Help target
Project folder name: **JavaHelp**
File name: projectname.jhm

See Context Sensitive Help in NetHelp (on page 202) for complete information on implementing NetHelp context sensitivity. If you are creating a JavaHelp or Microsoft Help 2.0 Target, see Context Sensitive Help in JavaHelp (on page 201) or Context Sensitive Help in Microsoft Help 2.0 (on page 202) for information about working with those Targets.

Context Sensitive Help in JavaHelp

JavaHelp context sensitive Help does not require context IDs; instead, it 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 **Ascii name** field in the **Topic Properties** dialog box 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 in the **Ascii name** field of the **Topic Properties** dialog box. (See Viewing/Changing Topic Properties on page 225 for more information.)

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. For generated (reference) topics, Context IDs are not needed for context sensitive help. Occasionally, you may need to use the **Context String** field for mapping narrative topics with the **Topic Properties** dialog box. (See Viewing/Changing Topic Properties on page 225 for more information.)

Context Sensitive Help in NetHelp

NetHelp, Doc-To-Help's browser-independent help format, supports context sensitive help that can be used many ways: 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. See Implementing Context Sensitive Help (on page 200) for more information.

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 by default in the C:\Program Files\ComponentOne\DocToHelp\Context-Sensitive Help folder.

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. 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** — txt_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 Implementing Context Sensitive Help (on page 200) 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 Implementing Context Sensitive Help (on page 200) 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.

Examples

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 Implementing Context Sensitive Help (on page 200) 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 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. 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);
```

Mapping Context Sensitive Help using DynamicHelp Authoring Mode

If your application has incorporated the **ComponentOne DynamicHelp control**, you can map your NetHelp or compiled HTML Help (.chm) directly to your application interface. You can do this with or without using Context IDs. See Implementing Context Sensitive Help (on page 200) for more information about this control.

This mapping is created in DynamicHelp Authoring Mode.

The mapping created is saved to an XML file. When you have finished mapping your Help project, the following files must be delivered to the software developer:

- **The Help Target** — either the NetHelp project files or a .chm file (see Doc-To-Help Output and Deliverables on page 30 for more information.)
- **The mapping XML file** — named either default.htm.xml (for NetHelp projects) or HelpFileName.chm.xml (for compiled HTML Help projects)

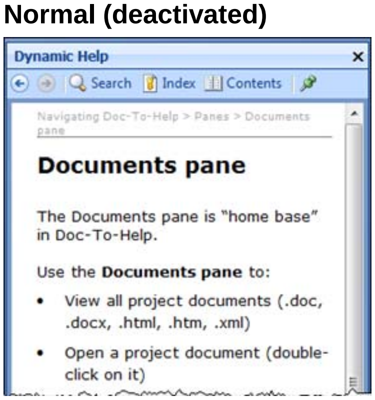
Before you begin mapping, you need to place your Help file(s) in the correct folder. NetHelp or .chm files should be copied to the install folder specified by the software developer for your application (for example: \\program files\componentone\DocToHelp\Help). You can specify that Doc-To-Help automatically drops your Help in a specific folder when building a Target using the **Folder** field in the **Help Targets** dialog box. See Creating Help Targets (on page 155) for more information. The XML file will be created automatically when you begin mapping.

Note: The .xml mapping file should never be edited manually. If mappings need to be deleted, you should do so using Authoring Mode.

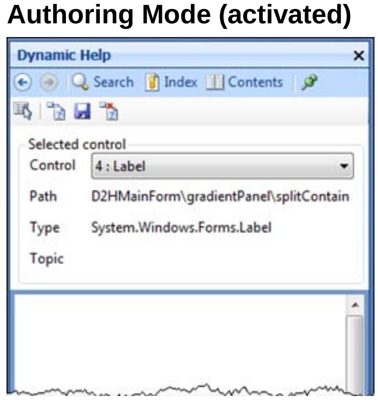
Opening Authoring Mode

The software developer controls how you activate and deactivate Authoring Mode. It is usually two separate key combinations that are specified when configuring the DynamicHelp control. (Software developers should see the Help included with the control for instructions.)

Normal (deactivated)



Authoring Mode (activated)

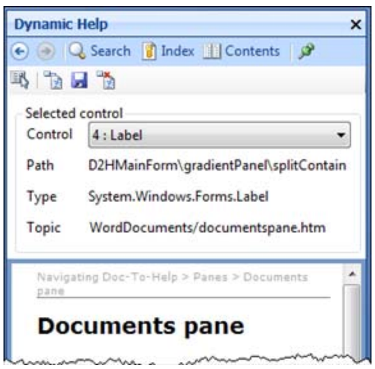


The Help is mapped using the **Authoring Mode** toolbar: 

Information about the control and its mapping is in the **Selected control** area of the panel:

Control — Control name : type of control.
Path — The path of the control relative to its placement on the form.
Type — Type of control.
Topic — The Help topic associated with the control. (If no topic has been assigned yet, it will be blank.)

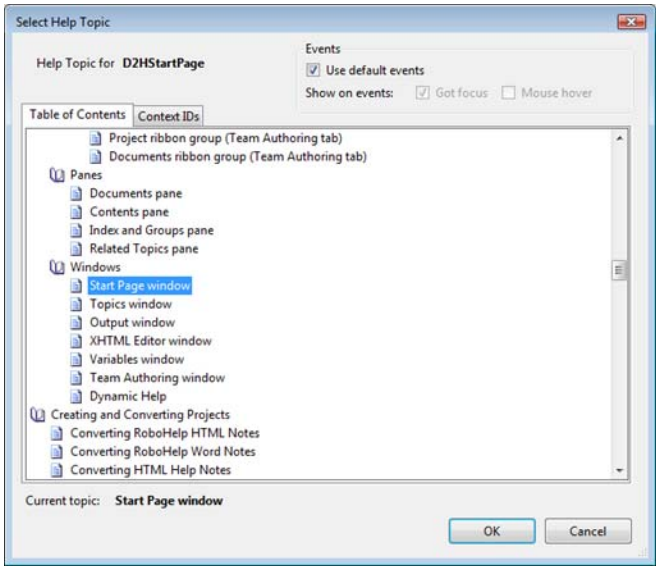
The blank area below is where the Help will display when mapped.



To map a Help topic in Authoring Mode

1. Display the area of the application's interface that you would like to map. For example, if you are going to map ribbons, make sure they are all available. If you are going to map windows, make sure they are all open.
2. Activate Authoring Mode.
3. Click the **Select control** button .
4. Click the item you would like to map a Help topic to. As you move the mouse over the controls, you will see information about them in the Authoring Mode panel: **Control** name, **Path**, control **Type**, and associated **Topic**, if applicable.

   If you are mapping to a dialog box, and you would like to map the entire dialog box to one Help topic, select the title bar of the dialog box.
5. Click the **Attach topic to control** button . The **Select Help Topic** dialog box appears and shows the **Table of Contents** tab, which displays the TOC of the source Help file.



6. By default, the **Use default events** check box is selected (the selected options will vary by the control you've chosen to map to). **Got focus** sets the Help topic to display when the control has been selected; **Mouse hover** sets the Help topic to display when the control has been hovered over. To change the default options, clear the **Use default events** check box and select either the **Got focus** or **Mouse hover** check box.

   If mapping to the TOC: From the **Select Help Topic** dialog box, **Table of Contents** tab, choose a topic. Click **OK**. The topic chosen will display in the **Topic** field of the Authoring Mode panel.

   If mapping to the context ID list: From the **Select Help Topic** dialog box, **Context ID** tab, choose the context ID/Topic pairing. Click **OK**. The topic chosen will display in the **Topic** field of the Authoring Mode panel.

Note: In order to map to a topic, the topic must be in the Help TOC or have a context ID assigned to it; otherwise, it will not appear in the **Select Help Topic** dialog box. Also, if you rename a TOC item in the Help after the Topic has already been mapped, the mapping will break. Simply drop the updated Help into the proper folder and re-create the mapping with the updated TOC.

---

7. Click the **Save** button .
8. Continue mapping.
9. Close Authoring Mode.
10. Deliver the proper Help and .xml mapping files to the software developer.

Note: After mapping, you should backup the .xml mapping file elsewhere on your machine. If you uninstall or reinstall your software product, the .xml mapping file will be deleted and replaced.

To delete a mapping

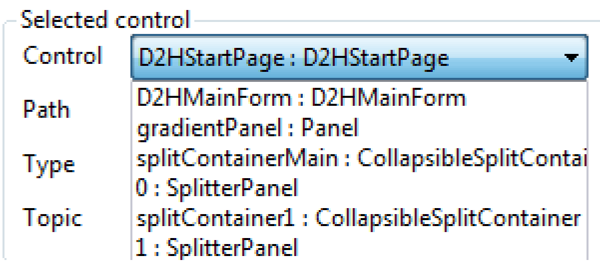
1. Activate Authoring Mode.
2. Click the **Select Control** button  and select the control that you would like to remove the topic mapping from. The control name appears in the **Control** field of the Authoring Mode panel, so you can verify the correct control has been selected.
3. Click the **Detach Topic from Control** button . The topic mapping is removed. Notice there is no longer a topic specified in the **Topic** field.
4. Click the **Save** button .

5. Close Authoring Mode.
6. Deliver the proper Help and .xml mapping files to the software developer.

To map to unselectable controls

There may be times you need to attach a topic not to a control you can select, but to its parent or ancestor. Since the parent can be completely covered by child controls and therefore unselectable, there is a special provision to do it in Authoring Mode.

1. Activate Authoring Mode.
2. Click the **Select control** button .
3. Click the child control of the parent or ancestor control you would like to map a Help topic to.
4. Click the drop-down arrow in the **Control** combo box and select the parent or ancestor to map to.



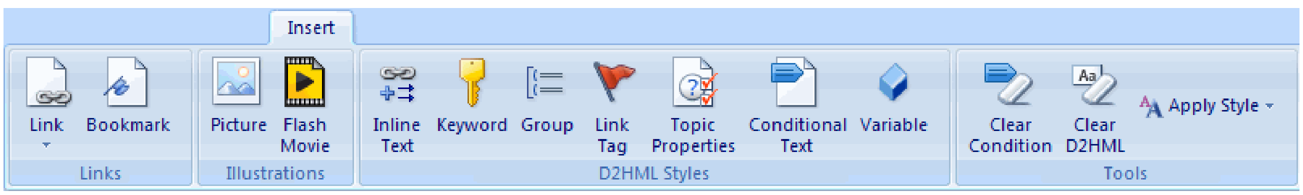
5. Click the **Attach topic to control** button . The **Select Help Topic** dialog box appears and shows the **Table of Contents** tab, which displays the TOC of the source Help file.
6. Select the topic from the **Table of Contents** tab or select the context ID/Topic pairing on the **Context IDs** tab, and specify any desired events.
7. Click the **Save** button .
8. Close Authoring Mode.
9. Deliver the proper Help and .xml mapping files to the software developer.

Using D2HML (Doc-To-Help Markup Language)

Doc-To-Help Markup Language (D2HML) is a set of predefined styles that you can use to mark up your source documents. D2HML makes it easy to create topic links, keywords, groups, and conditional text, as well as insert variables. You don't need to learn any special language. Styles are applied with a single click.

When editing documents in Microsoft® Word, Microsoft® FrontPage®, or Adobe® Dreamweaver®, you can use the appropriate **Doc-To-Help** ribbon, toolbar, or menu to quickly and easily apply D2HML styles:

When editing XHTML documents, use the **Insert** ribbon in Doc-To-Help:



When you apply a D2HML style to text, a hot spot is created.

The Styles are listed below for reference, but you should always use the ribbon or toolbar to apply them.

Toolbar button	Function	Word Style	HTML	Notes
	[Topic Link](#) (on page 212) Create a hotspot link to a topic (jump or popup) Create a hotspot link to topic(s) containing the selected keywords, to a group of topics, to a bookmark in a document, or to a link tag.	C1HJump C1HPopup C1HKeywordLink C1HGroupLink	 	The topic list opened by a Keyword or Group link can be displayed in a dialog box or popup menu. Index entries and groups can also be created in the [Index and Groups pane](#) (on page 110) of Doc-To-Help. Link Tags are created with the **Link Tag** button. See [Adding a Link Tag](#) (on page 217)
	[Inline Text](#) (on page 215) Create a hotspot that will display additional text on the same line, in a dropdown, or in a popup.	Hotspot: C1HInlineExpand C1HInlineDropdown C1HInlinePopup Expanding Text: C1HExpandText C1HDropdownText C1HPopupText	Hotspot: Expanding Text: 	The hotspot will be tagged with the “hotspot” style/HTML; the text that will be displayed by the hotspot is tagged with the “expanding text” style/HTML. The inline/expanding/drop-down text is invisible by default. See [Showing Hidden Hotspots](#) (on page 222) for more information.
	[Keyword](#) (on page 215) Designates a hotspot as an index keyword.	C1HIndex C1HIndexInvisible	 	The hotspot can be visible or invisible. Use the “Invisible” style/HTML to make the hotspot invisible. (The “Visible” check box in the Keyword dialog box controls visibility.) See [Showing Hidden Hotspots](#) (on page 222) for more information.
	[Group](#) (on page 216) Designates a hotspot as a member of a group.	C1HGroup C1HGroupInvisible	 	If invisible the hotspot does not appear in the output. See [Showing Hidden Hotspots](#) (on page 222) for more information.

Toolbar button	Function	Word Style	HTML	Notes
	Link Tag (on page 217) Designates a hotspot as a link target.	C1HLinkTag C1HLinkTagInvisible	 	If invisible the hotspot does not appear in the output. See Showing Hidden Hotspots (on page 222) for more information.
	D2HML Topic Properties (on page 218) Sets a subset of the topic properties.	C1HTopicProperties		All topic properties can be controlled using the Topic Properties dialog box in Doc-To-Help. The Topic Properties are invisible by default. See Showing Hidden Hotspots (on page 222) for more information
	Conditional Text (on page 220) Used to mark text as conditional by platform, target, attribute or any combination of all three.	C1HConditional		Conditional text will also be tagged with platform, target, and/or attribute information (platform=, target=, attribute=)
	Variable (on page 221) Creates a hotspot to a text or rich content variable.	C1HVariable		The hotspot chosen is replaced by the variable.

You can use the predefined D2HML styles as-is, or you can edit them and create your own.

- See Defining Paragraph/Character Styles and Topic Types (on page 181) for more information on editing and creating styles.
- See Editing a CSS (on page 189) for more on working with cascading style sheets.
- The default project **Source Template**, **Source CSS**, **Target Template**, and **Target CSS** are all defined in the Home tab (on page 98). See Guide to Templates and Styles (on page 5) for more information on Doc-To-Help's predefined templates and styles and how to work with them.

Creating Links

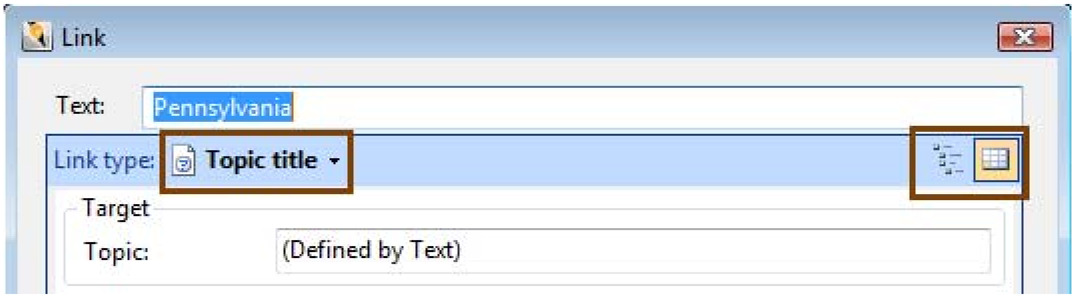
The **Link dialog box** is used to create links to topics, bookmarks, link tags, keywords, groups, and glossary entries.

Where to find the Link button

- Doc-To-Help's Editor: the Insert tab (on page 103) (Choose **In Project** from the drop-down.)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

To create a link

1. Select text in the **Editor window**.
2. Click the **Link button**. The **Link** dialog box will open.
3. Click the **Link type** drop-down to choose the type of link you would like to create. You can choose from the following: **Topic title**, **Topic in Document**, **Link tag**, **Bookmark in Document**, **Keyword**, **Group**, or **Glossary term**. Note that you can choose to view the "link to" information in **Grid View** or **Tree View** (click the toggle highlighted below).



4. Select the appropriate topic, link tag, bookmark, keyword, group, or glossary entry. (You may select more than keyword or group.) For topic links, the default Target topic will be identical to the selected text. The phrase **(Defined by Text)** will be displayed in the **Topic** field.
5. Set your **Link Options** (see below).
6. Set your **Options**. Options will vary slightly depending on your choice of editor.

   In HTML Editors and Doc-To-Help's XHTML Editor:

- **Properties in attribute** puts the link properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window.
- **Properties in text** puts the attribute in the text so it is visible at all times. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Hotspot|tag=link` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode.)

   In Word, you may select the **Hide Properties in Document** check box to hide the tag.

7. If you would like the link to be a button in the Help Target with the selected Text displayed on it, select the **Show as button** check box.
8. Click **OK**.

If you would like to create a new keyword, group, or link tag, see Inserting an Index Entry (on page 215), Adding a Topic to Group (on page 216), or Adding a Link Tag (on page 217). You can also create/edit keywords and groups using the Index and Groups Pane (on page 110) in Doc-To-Help.

Topic link options

Window: By default, the window type of the topic chosen. If you would prefer the destination topic display in another window type, choose one from the drop-down list. If you would like the information to display in a popup, select the **Popup** check box. The (Default) topic window types are set using Topic Types, which are one component of a Paragraph Style. These are set for the project in the **Project Styles** dialog box. See Defining Character/Paragraph Styles and Topic Types on page 181.

Keyword/Group link options

Topic list in: When a hotspot is linked to keywords or a group, clicking on the hotspot will display the list of topics in a dialog box or popup menu. Use this drop-down list to specify your preference.

Window: By default, the window type of the topic(s) chosen. If you would prefer the destination topic(s) display in another window type, choose one from the drop-down list. The (Default) topic window types are set using Topic Types, which are one component of a Paragraph Style. These are set for the project in the **Project Styles** dialog box. See Defining Character/Paragraph Styles and Topic Types on page 181.

Drag-and-Drop Linking

If you would prefer, you don't need to use the Link button (on page 212) to create links to topics, keywords, groups, and documents — or the Variable button (on page 221) to insert Text variables.

Using drag-and-drop linking, you can create links from the:

- **Topics** window
- **Keywords and Groups** pane (Keywords and Groups)
- **Documents** pane
- or
- **Variables** window

to:

- Microsoft® Word documents
- XHTML documents (Doc-To-Help's **XHTML Editor** window)
- HTML documents (any editor that supports drag-and-drop)

Please note that using drag-and-drop linking will create a direct link to the topic, keyword, group, or document selected; if you would like to specify a specific destination window or bookmark — or that the link should appear as a Popup — you should use the Link dialog box (on page 212).

To create a drag-and-drop link

1. Open your source document.

   If your source documents are in Microsoft® Word or HTML, arrange Doc-To-Help and the document windows side-by-side.

2. To create a link to a topic or glossary item, open the **Topics** window in Doc-To-Help.

   To create a link to a Keyword or Group, open the **Index and Groups** pane and choose the appropriate area (**Keywords** or **Groups**).

   To create a link to a Document, open the **Documents** pane.

   To insert a Variable, open the **Variables** window.

3. Select the topic, keyword, group, document, or Text variable and drag it into your document. The link will be created in your document.

Note: When working with the XHTML Editor, dragging from the panes is straightforward; however, there is a trick to drag-and-drop from one window to another window (since they overlap).

1. Select the item in the **Topics** or **Variables** window.
2. Drag it to the **XHTML** window tab. The **XHTML Editor** window will open.
3. Drop the item at the desired place in the **XHTML** editor window. The link will be created.

Creating Inline, Dropdown, or Popup Text

The **Inline text dialog box** is used to create three different options for displaying additional information.

Where to find the Inline Text button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

Click on the hotspots below for examples of inline, dropdown, or popup text.

Inline text displays immediately after a selected link.
The Lobby is the "Home Page" of FrontLine and can be accessed directly via the **Lobby** button.

Dropdown text displays under a selected link.
The Lobby is the "Home Page" of FrontLine and can be accessed directly via the **Lobby** button.
This is an example of dropdown text.

Popup text displays in a popup window when the link is selected.
The Lobby is the "Home Page" of FrontLine and can be accessed directly via the **Lobby** button.

To create inline, dropdown, or popup text

1. Select text in the **Editor** window.
2. Click the **Inline Text** button. The **Inline text** dialog box will open. The link text will be displayed in the **Selection** area.
3. Enter the text to be displayed in the **Text** box.
4. Choose the appropriate **Option** (Expand text inline, Dropdown text, Show text in popup). If working in Microsoft® Word, select the **Text in Comment** check box if you would like the inline, dropdown, or popup text to appear as a comment.
5. Click **OK**.

Doc-To-Help will display a message box informing you that this is an invisible style (meaning the inline/dropdown/expanding text we added will be not be displayed in our Source document — or in our Target until clicked). Since that is OK, click **No** to close the message box. If you'd like to make this information visible, see Showing Hidden Hotspots on page 222.

It is also possible to create expanding/collapsing text in HTML Help, NetHelp, and Help 2.0 targets, see Creating Expanding/Collapsing Sections on page 27 for more information.

Inserting an Index Entry

You can assign and add keywords (index entries or K-links) to your documents using D2HML.

Please note you can also manage keywords from the Index and Groups pane (on page 110) in Doc-To-Help.
Links to keywords are created using the Link dialog box (on page 212). Links can be text or buttons; when selected they display a dialog box or popup window listing all the keyword topics.

Where to find the Keyword button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

The options available will vary slightly depending on your choice of editor.

To add and assign keywords

1. Select text in the **Editor** window.
2. Click the **Keyword** button. The **Keyword** dialog box will open. The selected text will be displayed in the **Text** area.
3. By default, the keyword will be the selected text. The phrase **(Keyword defined by Text)** will be displayed in the **Keyword** field. If you would prefer to use another keyword, use one of these options:

   If you would like to assign one or more existing keywords to the text, select the checkbox(es) in the **Keywords** area.

   If you would like to add a new keyword to the list and assign it, click the **Add New Keyword** button. An editable keyword will be created, titled **New Keyword**. Type the new keyword within the box. To create a secondary keyword, select a keyword and click the **Add Secondary Keyword** toolbar button. Select the keyword(s) check boxes.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

   In HTML Editors and Doc-To-Help's XHTML Editor:

   - **Properties in attribute** puts the keyword properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window.
   - **Properties in text** puts the attribute in the text. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Hotspot|tag=keyword` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode. See Showing Hidden Hotspots on page 222.)

   In Word, you may select the **Hide Properties in Document** check box to hide the tag. Clearing the **Visible** check box will hide the text in your source document. See Showing Hidden Hotspots on page 222.
5. Click **OK**.

Please note that any new keywords created using this method will be added to the Index and Groups pane (on page 110) in Doc-To-Help.

Adding a Topic to a Group

Groups are related topics or associative topics (A-links). You can add a topic to a new or existing group using D2HML. See Creating an Index or Groups (on page 231) for an explanation of Groups and their uses.

Please note you can also create and manage groups from the Index and Groups pane (on page 110) in Doc-To-Help.

Links to groups are created using the Link dialog box (on page 212). Links can be text or buttons; when selected they display a dialog box or popup window listing all the topics in the group.

Where to find the Group button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

To add a topic to a group

1. Select text in the **Editor** window.
2. Click the **Group** button. The **Group** dialog box will open. The selected text will be displayed in the **Text** area.
3. By default, the topic will be added to a group whose name is defined by the selected text. The phrase **(Group defined by Text)** will be displayed in the **Group** field. If you would prefer to add the topic to another group, use one of these options:

   If you would like to add the topic to an existing group (or groups), select the appropriate check box(es).

   If you would like to add a new group to the list, click the **Add New Group** button. An editable group will be created, titled **New Group**. Type the new group name within the box. Select the group(s) check boxes.

   Please note that any new groups will be added to the Index and Groups pane (on page 110) in Doc-To-Help. You may use the Groups pane to manage the topics that belong to the new groups you have created.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

   In HTML Editors and Doc-To-Help's XHTML Editor:

   - **Properties in attribute** puts the group properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window.
   - **Properties in text** puts the attribute in the text. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Hotspot|tag=group` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode. See Showing Hidden Hotspots on page 222.)

   In Word, you may select the **Hide Properties in Document** check box to hide the tag. Clearing the **Visible** check box will hide the text in your source document. See Showing Hidden Hotspots on page 222.
5. Click **OK**.

Adding a Link Tag

A Link Tag is a unique identifier for a topic, and makes it possible to create a jump or popup link to a topic. If a Paragraph Style has the **Auto Link** check box selected, then Doc-To-Help will automatically create a Link Tag for every topic using that style. (See Defining Character/Paragraph Styles and Topic Types on page 181.) Automatic Link Tags are identical to the Topic Title, but spaces, hyphens, and period are replaced by underscores.

Sometimes you may need to manually create a unique Link Tag for a topic, particularly if you have two or more topics with the same name, and therefore the same Link Tag. The **Link Tag** dialog box makes this possible. A topic can have more than one Link Tag.

Link Tags can be viewed in the Topics window (on page 112) (right-click and choose **Link Tag** from the **Columns** menu).

Where to find the Link Tag button 

- Doc-To-Help's Editor: the Insert tab (on page 103)

- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

To add a link tag manually

1. Select text in the **Editor** window.
2. Click the **Link Tag** button. The **Link tag** dialog box will open. The selected text will be displayed in the **Link Tag** area.
3. By default, the **Link Tag** will be the selected text. The phrase **(Link Tag defined by Text)** will be displayed in the **Link Tag** field. If you would prefer to use another link tag, use one of these options:

   If you would like to assign one or more existing link tags to the text, select the checkbox(es) in the **Link tags** area.

   If you would like to add a new link tag to the list and assign it, click the **Add New Link Tag** button. An editable link tag will be created, titled **New Link Tag**. Type the new link tag within the box. Select the link tag(s) check boxes.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

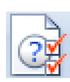
   In HTML Editors and Doc-To-Help's XHTML Editor:

   - **Properties in attribute** puts the link tag properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window. Selecting the **Visible** check box will make the hotspot visible in the Help Target.
   - **Properties in text** puts the attribute in the text. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Hotspot|tag=linktag` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode. See Showing Hidden Hotspots on page 222.)

   In Word, you may select the **Hide Properties in Document** check box to hide the tag. Clearing the **Visible** check box will hide the text in your source document. See Showing Hidden Hotspots on page 222.
5. Click **OK**.

Setting D2HML Topic Properties

Specific topic properties may be set using D2HML; please see Viewing/Changing Topic Properties (on page 225) for information about making adjustments to other topic properties.

Where to find the Topic Properties button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

Topic Properties will be hidden by default, see Showing Hidden Hotspots (on page 222) for more information.

To set D2HML topic properties

1. Select text in the **Editor** window (or simply insert your cursor in the window). Note that the selected text will become invisible by default.
2. Click the **Topic Properties** button. The **D2HML Topic Properties** dialog box will open.

3. Make changes and click **OK**. To view/adjust additional topic properties, see Viewing/Changing Topic Properties on page 225.

General

ASCII name: Specifies the ASCII-only string used to identify the topic in situations where non-ASCII characters are not allowed. 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 language.

URL: Specifies the name of the generated .htm file for this topic. By default, the name of the .htm file matches the topic text with spaces, hyphens and underscores omitted.

Context ID: The unique numeric identifier assigned to the topic during compilation (read-only). This allows the topic to be used in context-sensitive help. Context ID settings are managed in the Project Properties (on page 196) dialog box.

Comments: An editable textbox for comments by the help author. These comments are not accessible by the end user.

Default topic: If selected, this will be the topic displayed when a Help file is opened (the "home page"). The default topic can also be set by selecting a topic in the Topics window (on page 112), and choosing **Default Topic** from the right-click menu.

Appearance

Contents title: This field allows you to change the topic title in the Table of Contents (Contents pane on page 109) without changing the actual topic title in the source document.

Display title: This field allows for modification of a topic title with respect to help file searches without changing the actual topic title in the source document. 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 force 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.

Related Links Label: Specifies the text that precedes the subtopic buttons for this topic. If you clear this field, there will be no text above the buttons. If you would like to change the label text for a specific Help Target, change it in the Help Targets dialog box (on page 155). The default label is **More:**

Include in Natural Search and in NetHelp Search: If selected, ComponentOne Natural Search is enabled for this topic. If cleared, this topic is excluded from the search.

Condition

These can also be set and will appear in the Condition ribbon group (on page 105).

Platforms: Sets a platform-based condition for the selected topic. The topic will be included in all the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets: Sets a target-based condition for the selected topic. The topic will be included in all the target(s) selected.

Attributes: Set an attribute-based condition for the selected topic. The topic will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > Attributes button) to create custom attributes.

Marking Text as Conditional

Using conditional text, you can mark specific text or graphics to display only in specific instances; by platform, target, attribute, or a combination. This makes logical single sourcing (on page 1) easy to accomplish.

You can also mark entire documents and topics as conditional, see Setting Document Properties (on page 136) and Setting Topic Conditions (on page 229).

Where to find the Conditional Text button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

To mark text as conditional

1. Select text in the **Editor** window.
2. Click the Conditional Text button. The Conditional Text dialog box will open. The selected text will be displayed in the Text area.
3. Choose the appropriate Conditional Properties. You may select more than one. Options are:
   - **Platforms** — Set a platform-based condition for the selected text. The text will be included in all of the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.
   - **Targets** — Set a target-based condition for the selected text. The text will be included in all the target(s) selected.
   - **Attributes** — Set an attribute-based condition for the selected text. The text will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > **Attributes** button) to create custom attributes.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

   In HTML Editors and Doc-To-Help's XHTML Editor:

   - **Properties in attribute** puts the conditional text properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window.
   - **Properties in text** puts the attribute in the text. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Conditional text|tag=platform; attribute;target` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode.)

   In Word:

   - **Hide Properties in Document** puts the conditional text properties in the text.
   - **Text in comment** puts the conditional text properties in a Word comment.
5. Click **OK**.

Marking text as **HTML passthrough code** is an advanced feature for Word documents only. HTML passthrough allows you to include HTML code in your document without Word treating that code as text. It is recommended that the **Rich Text variables** feature be used instead of the HTML passthrough code. With this feature, you can define a variable in an HTML document and insert it into a Word document. See Variables window (on page 114) for more information.

If you insert an HTML Help ActiveX Control into your Word document (also an advanced feature), that HTML code will be marked with the HTML passthrough condition. See Inserting an HTML Help ActiveX Control (on page 133) for more information.

To clear conditional text

1. Select conditionalized text in the Editor window.
2. Click the Clear Condition button (FrontPage and Dreamweaver only) . The conditional text styles will be cleared while leaving other formatting intact.

Please note that you can see how your conditional text will display for the currently selected target by clicking the **Preview** button at the bottom of the XHTML Editor window in Doc-To-Help.

Inserting a Variable

Using variables, you can manage content in one place for reuse across your project because variable hotspots are replaced with variable text in the final project. **Text Variables** may be used for unformatted text or use **Rich Content Variables** for blocks of formatted content. You can even assign conditions to variables. Variables make single sourcing (on page 1) easier to accomplish, and also saves time making multiple updates throughout your projects.

Please note that **Text Variables** will use the formatting that is used at their insertion point. **Rich Content Variables** will use the formatting applied to the variable itself.

See Creating Variables (on page 194) for more information about creating variables.

Where to find the Variable button 

- Doc-To-Help's Editor: the Insert tab (on page 103)
- Microsoft® Word: the Doc-To-Help toolbar or ribbon (on page 128)
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon (on page 135).

To insert a variable

1. Select text in the Editor window. (You may want to use the name of the Variable for the hotspot text. The hotpot selected will be completely replaced by the variable.)
2. Click the **Variable** button. The **Variable** dialog box will open. The selected text will be displayed in the **Text** area.
3. Choose the appropriate **Variable** from the list.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

   In HTML Editors and Doc-To-Help's XHTML Editor:

   - **Properties in attribute** puts the variable properties inside the html tag in an attribute. The tag properties may be viewed in the **Code** or **Source** view of the Editor window.
   - **Properties in text** puts the attribute in the text. The tag will display in the **Design** view of the Editor window, as well as in the **Code** or **Source** view. `Hotspot|tag=variable` (In Dreamweaver, you may use the **Hidden** check box to hide this tag in **Design** mode.)

   In Word, you may select the **Hide Properties in Document** check box to hide the tag.

5. Click **OK**.

Shortcut: You can also insert Text Variables using drag-and-drop. See Drag-and-Drop Linking (on page 214) for more information.

Please note that if you are working in the XHTML Editor, you can see how your variables will display for the currently selected target by clicking the **Preview** button at the bottom of the editor window in Doc-To-Help.

When your help target is built, the value of the variable(s) used is automatically inserted in the target. For text variables, the value will have the same formatting (font, etc.) as its insertion point. For rich content variables, the variable will retain the formatting applied to the variable when it was created.

Clearing D2HML Styles

In Doc-To-Help's Editor, the **Clear D2HML** button will be on the Insert tab (on page 103); in Microsoft® FrontPage®, and Adobe® Dreamweaver®, the button will be on the **Doc-To-Help D2HML Styles** toolbar or ribbon.

In Microsoft® Word, the **Clear Formatting** button will be on the **Doc-To-Help** toolbar or ribbon.

See Using D2HML (Doc-To-Help Markup Language) (on page 210) for more information on D2HML styles and how they work.

To clear a style

1. Select text in the **Editor** window.
2. Click the **Clear D2HML** button . The style will be removed from the text. (In Word, the button is named **Clear Formatting**.)

Showing Hidden Hotspots

Invisible hot spot types, such as Invisible Keyword and Topic Properties, are not visible in help targets and certain source documents. The **Show Hidden Hot Spots** button available on the **D2HML Styles** toolbar in Microsoft® Word and Adobe® Dreamweaver® allows you to show any invisible hot spots in your source document, making it possible 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:

- Invisible Keyword
- Invisible Group
- Invisible Link Tag
- Topic Properties
- Inline Text (The inline text is invisible by default; the hot spot is displayed.)
- Dropdown Text (The dropdown text is invisible by default; the hot spot is displayed.)
- Popup Text (The popup text is invisible by default; the hot spot is displayed.)

To make hidden hot spots visible

1. Click the **Show Hidden Hotspots** button . The **Show Hidden Hotspots** dialog box will appear.
2. Select the **Show All** button, or individually choose the hotspot types you'd like to view.
3. Select **Apply to all documents with this template** if you would like your selections in this dialog box to apply to any other documents that have the same template attached.
4. You 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. Clear it if you do not want a dialog to display each time you format text with an invisible hotspot style.
5. Click **OK**.

Managing Topics

Topics can be managed a number of ways in Doc-To-Help. You can change specific topic properties and conditions, as well as create indexes and tables of contents. The "home base" for working with topics is the Topics window (on page 112), which works along with the Topics tab (on page 104) and Index and Groups (on page 110), Contents (on page 109), and Related Topics panes (on page 111).

Viewing/Changing Topic Properties

The properties of specific topics — for example, the topic title and type (conceptual, procedural, etc.), and condition may be reviewed and edited in the **Topic Properties** dialog box.

Certain topic properties can be edited within the **Topics window** itself (for example: **Type**, **Context ID**, **Keywords**, and **Groups**). Simply click in the grid at the appropriate spot to open a drop-down box or activate the field. (To see the columns available for display, right click in the window and choose **Columns**. See Topics window (on page 112) for more information.)

If you would like to view or change the properties of an entire document, use the Document Properties dialog box (on page 136).

To open the Topic Properties dialog box

1. Choose the desired topic from the Topics window (on page 112).
2. Right-click on the topic and choose **Properties**. The **Topic Properties** dialog box will open.

   You can also choose the topic and click the **Properties** button in the **Topics** tab to open the dialog box.

If you would like to edit multiple topics, choose them in the **Topics** window using Shift+Click or Ctrl+Click. Right-click and choose **Properties**.

General

Title: The name of the topic, taken directly from the source document. This field is read-only; you can only change it within the document. In HTML documents, Doc-To-Help will use the document's <title> tag as the **Title** field.

Topic type: Specifies a named set of display, navigation, and indexing characteristics to be associated with this topic (such as what window the help topic appears in, how the help topic is accessed, and whether it gets a map number). To learn more about Topic Types, see Defining Character/Paragraph Styles and Topic Types on page 181.

ASCII name: Specifies the ASCII-only string used to identify the topic in situations where non-ASCII characters are not allowed. 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 language.

URL: Specifies the name of the generated .htm file for this topic. By default, the name of the .htm file matches the topic text with spaces, hyphens and underscores omitted.

Context ID: The unique numeric identifier assigned to the topic during compilation (read-only). This allows the topic to be used in context-sensitive help. Context ID settings are managed in the Project Properties (on page 196) dialog box.

Context string: When using context-sensitive (F1) and dynamic help with Microsoft Help 2.0, specifies the context string for the topic. Each topic can have one or more context strings, or none. The context strings must be 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:
Context string = "MyCompany.MyProduct.MyProperty"

Topic "Properties MyProperty1 and MyProperty2" can have:
Context string = "MyCompany.MyProduct.MyProperty1;MyCompany.MyProduct.MyProperty2"

Comments: An editable textbox for comments by the help author. These comments are not accessible by the end user.

Default topic: If selected, this will be the topic displayed when a Help file is opened (the "home page"). The default topic can also be set by selecting a topic in the Topics window (on page 112), and choosing **Default Topic** from the right-click menu.

Appearance

Contents title: This field allows you to change the topic title in the Table of Contents (Contents pane on page 109) without changing the actual topic title in the source document.

Display title: This field allows for modification of a topic title with respect to help file searches without changing the actual topic title in the source document. 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 force 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.

Related Links Label: Specifies the text that precedes the subtopic buttons for this topic. If you clear this field, there will be no text above the buttons. If you would like to change the label text for a specific Help Target, change it in the Help Targets dialog box (on page 155). The default label is **More:**

Hide Subtopic Links: If selected, the subtopic links will be hidden for this topic.

Include in Natural Search and in NetHelp Search: If selected, ComponentOne Natural Search is enabled for this topic. If cleared, this topic is excluded from the search.

Condition

These can also be set and will appear in the Condition ribbon group (on page 105). See Utilizing Conditions (on page 174) for more information about conditions.

Platforms: Sets a platform-based condition for the selected topic. The topic will be included in all the target platform(s) selected. A platform includes all the targets that produce the output specified; Online Help, HTML Help, NetHelp, Printed Manual, etc.

Targets: Sets a target-based condition for the selected topic. The topic will be included in all the target(s) selected.

Attributes: Set an attribute-based condition for the selected topic. The topic will be included or excluded when creating conditional builds (for example, internal or external.) Use the Attributes dialog box (on page 175) (Project tab > Project ribbon group > Attributes button) to create custom attributes.

WinHelp

Macro: Specifies the macro to run when this topic is opened. Only available for WinHelp targets.

Module link

Module file: For a placeholder topic in a modular hub project, specifies the platform-dependent help file to be loaded dynamically. (Applies only to WinHelp, HTML Help and NetHelp modular hub projects.)

Contents file: For a placeholder topic in a modular hub project, specifies the platform-dependent contents file to be loaded dynamically. (Applies only to WinHelp, and HTML Help modular hub projects.)

Title: For a placeholder topic in a modular hub project, specifies the text used to disambiguate like-named topics in keyword search lists. (Applies only to WinHelp modular hub projects.) If not specified, the help file specified in the **Module file** field is used.

Printing and Exporting the Topic List

You may print your entire Topic List, and also export it to Microsoft® Excel® or text (.txt) if you wish. Exporting your Topic List is useful if you'd like to sort or filter your topics, or if you'd like to provide a list of Topics and Context IDs to the Software Development team for the implementation of context sensitive help. See Implementing Context Sensitive Help (on page 200) for more information on Context IDs.

Printing the Topic List

Before printing the Topic List, you may preview it, change the page settings, and/or choose from three print options.

To print the Topic List

1. Open the **Topics window** to display the Topic List.
2. Choose the **Topics tab**.
3. Click the **Print and Export** button in the View ribbon group (on page 105).
4. Choose **Print** from the list of options. The **Print** dialog box will open.
5. Select the options you'd like and click **OK**.

To preview the Topic List

1. Open the **Topics window** to display the Topic List.
2. Choose the **Topics tab**.
3. Click the **Print and Export** button in the View ribbon group (on page 105).
4. Choose **Preview** from the list of options. The **Print Preview** dialog box will open.
5. Click the **Print** button in the upper left to print the Topic List, or click the **Close** button to close the **Print Preview** dialog box.

To change the Topic List page settings

1. Open the **Topics window** to display the Topic List.
2. Choose the **Topics tab**.
3. Click the **Print and Export** button in the View ribbon group (on page 105).
4. Choose **Page Settings** from the list of options. The **Page Setup** dialog box will open.
5. Choose the paper size, orientation, and margins.
6. Click **OK**.

To choose a print option for the Topic List

1. Open the **Topics window** to display the Topic List.
2. Choose the **Topics tab**.
3. Click the **Print and Export** button in the View ribbon group (on page 105).
4. Choose **Print Options** from the list of options. The **Print Options** dialog box will open.
5. Select the check box next to the appropriate option
   - **Fit to page width** — The list will scale so that it is only one page wide.
   - **Fit to entire page** — The list will scale so that it all fits on a single page.
   - **Extend last column** — The last column of the list will be extended on the printed page.
6. Click **OK**.

Please note: Only the columns displayed will be printed. See Topics window (on page 112) for information on how to display/hide topics.

Exporting the Topic List

Exporting your Topic List is useful if you'd like to sort or filter your topics, or if you'd like to provide a list of Topics and Context IDs to the Software Development team for the implementation of context sensitive help.

See Implementing Context Sensitive Help (on page 200) for more information about Context IDs.

To export the Topic List

1. Open the **Topics window** to display the Topic List.
2. Choose the **Topics tab**.
3. Click the **Print and Export** button in the View ribbon group (on page 105).
4. From the **Export to …** list, choose **Excel** or **Text**.
5. The **Save As** dialog box will open.
6. Choose the file location and name.
7. Click **OK**.

Please note: Only the columns displayed will be exported. See Topics window (on page 112) for information on how to display/hide topics.

Setting Topic Conditions

If you would like to set specific platform, target, and/or attribute conditions for a topic, you may do so two different ways. Select the topic in the **Topics** window then:

1. Open the **Topics** tab, Condition ribbon group (on page 105). Select the condition(s) using the drop-downs.

   or

2. Open the **Topics** tab and click the **Properties** button. The **Topic Properties** dialog box will open. Select the conditions in the **Condition** section.

No matter which method is used to set the conditions, they will always be displayed in both the Topic Properties dialog box (on page 225) and the Condition ribbon group (on page 105).

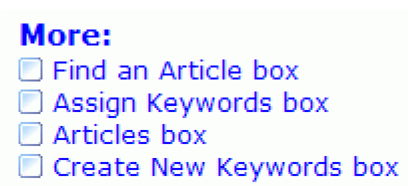
Attributes must be specified in the **Help Targets** dialog box (see Creating Help Targets on page 155) to be properly included/excluded from the final output. See Defining Attributes on page 175.

Managing Related Topics

Related Topics (also known as subtopic links) can be created/edited and viewed using the **Related Topics** pane and the **Topics** window. The **Related Topics** feature is very powerful, because it allows you to customize relevant “See Also” links for any topic, without the need to create individual cross-reference links. Combined with the automatically created subtopic relations, this means your online Help projects can help your users find the exact topic they need, when they need it. It can also eliminate topic “dead ends” in your online Help.

Related Topics can also be part of your single sourcing strategy. See Introduction to Single Sourcing (on page 1) for more information.

By default, Related Topics will display at the bottom of an online Help topic like this:



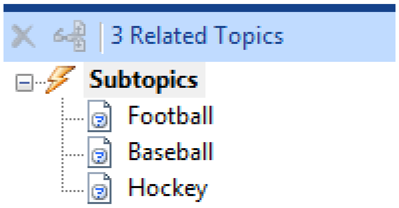
Using the **Related Topics** pane, you can add additional subtopic links to a topic, as well as disable or hide links.

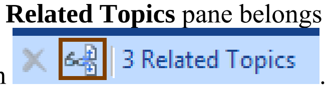
Automatic Subtopic Links

Subtopic links are created automatically by Doc-To-Help based on heading styles. They display in online Help Targets at the bottom of the topic.

By default, the related subtopics are the children of the main topic (Heading 1s will display Heading 2 as subtopics; Heading 2s will display Heading 3s.) The display of subtopic links is controlled by the **Topic Type** of the Heading style, and can be turned on/off in the Project Styles dialog box (on page 181) using the **Auto Subtopic Links** check box.

In the bottom section of the **Related Topics** pane, these automatic relations are flagged as “Subtopics.”



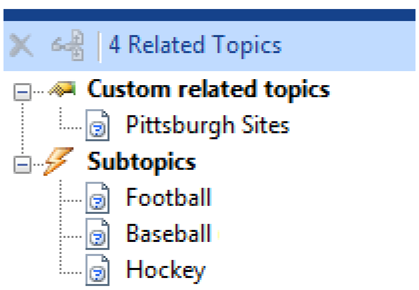
You can note where any topic displayed in the lower half of the **Related Topics** pane belongs in the topic hierarchy by selecting it and clicking the **Go to Related Topic** toolbar button 

.

To create a custom relation to a topic

1. Open the **Related Topics** pane.
2. Select a topic in the **Related Topics** pane. This topic could already have automatic subtopics, custom related topics, or no related topics.
3. Go to the **Topics** window, select the desired related topic and drag it into the lower half of the **Related Topics** pane. The new relation will be flagged as a "Custom Related Topic."

   This custom relation will display in the **Related Topics** list for that topic after the Target is rebuilt.



To remove a custom relation, choose it and click the **Remove Relation** button 

.

To hide subtopic links

If you would prefer that the entire Related Topics list does not display in a specific topic.

1. Open the **Related Topics** pane.
2. Select the topic in the **Related Topics** pane.
3. Click the **Disable Subtopic Relations** toolbar button. 

In the bottom section of the **Related Topics** pane, the message will read "Subtopic Relations Disabled by User."

   Please note that in the Topics tab (on page 104) (**Related Topics** ribbon group), the **Hide Subtopic links** check box is now selected.

Changing the Label for Related Links

If you'd like to change the **Related Links Label** for your entire Help Target (not just a single topic):

1. Open the **Help Targets** dialog box. (**Home** tab > **Target** ribbon group dialog box launcher.)
2. Choose the appropriate online **Help Target** from the list on the left.
3. Change the **Label** field. (By default, the label is **More:**) See Creating Help Targets (on page 155) for more information.

If you would like to change the label for individual topics:

1. Open the **Related Topics** pane.
2. Select the topic in the **Related Topics** pane.
3. In the Topics tab (on page 104) (**Related Topics** ribbon group), edit the **Related Links Label** field.

Changing the Style of the Related Links Label and Links

1. Open the **Help Targets** dialog box. (**Home** tab > **Target** ribbon group dialog box launcher.)
2. Choose the appropriate online **Help Target** from the list on the left.
3. Change the **Label style** and/or **Links style** field(s). See Creating Help Targets (on page 155) for more information.

See Related Topics pane (on page 111) for more information.

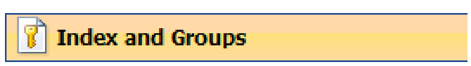
Creating an Index or Groups

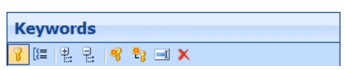
Indexes and Groups can be built quickly and easily using the **Index and Groups** pane and the **Topics** window.

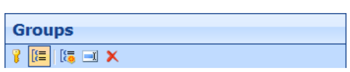
Index entries are also called keywords (or K-links). They will display in your Help or manual targets in an index. You can also create links to keywords.

Groups can also be referred to as related topics or associative topics (A-links). When a group of topics is related, a link can be created in your Help project that displays a dialog box or popup window displaying the list of topics in that Group. Groups do not display in the index. (The graphic below is an example of a popup menu displayed when a user clicks a group link. The user can then choose one topic from the menu and follow it.)

To open the appropriate pane:

- Click **Index and Groups** 

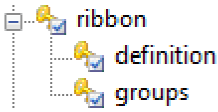
- Click the **Keywords button** 

- or **Groups button** 

To create a keyword

1. In the **Keywords** pane, click the **Add New Keyword** toolbar button  .
2. An editable keyword will be created, titled **New Keyword**. Type the new keyword within the box.
3. To create a secondary keyword, select a keyword and click the **Add Secondary Keyword** toolbar button  .

   If a secondary keyword is assigned to a keyword, the primary keyword acts as a parent. For example:



Topics may be assigned to all keyword levels.

Secondary keywords may be created several levels deep if you wish.

4. To assign a topic to a keyword, see **To assign a topic to a keyword or group** (below).

   Keywords can be renamed or removed using the **Rename** or **Remove Keyword** toolbar buttons.

To create a group

1. In the **Groups** pane, click the **Add New Group** toolbar button .
2. An editable group name will be created, titled **New Group**. Type the new group name within the box.
3. To assign topics to a group, see **To assign a topic to a keyword or group** (below).

   Groups can be renamed or removed using the **Rename** or **RemoveGroup** toolbar buttons.

Note: You can create index and group entries within your document (in Microsoft® Word/FrontPage, Adobe® Dreamweaver, or the XHTML Editor). See Using D2HML (on page 210) for details.

To assign a topic to a keyword or group

1. Create an index keyword or group.
2. Select it in the **Keywords/Groups pane** to highlight it.
3. Drag a topic from the Topics window (on page 112) into the bottom half of the **Keywords/Groups pane**.

   You may assign more than one topic to a keyword. This will open a selection dialog box when the keyword is chosen from the index.

   Groups should always have more than one topic assigned to them.

To create an index automatically (auto-index)

If you'd prefer, Doc-To-Help can create an index for you automatically, based on your Topic Types. Choose the **Topic Type(s)** you'd like to include in your index from the Auto-Index menu (on page 107) (you can select one Topic Type at a time). The next time you build your Target, topics with those Topic Properties will be included in the Index. Topic Types can be edited in the Project Styles dialog box (on page 181).

To create a link to a keyword or group

These steps should be performed within your document editor, for example Microsoft® Word/FrontPage, Adobe® Dreamweaver, or the XHTML Editor. See Creating Links (on page 212) for more information.

See Index and Groups pane (on page 110) for a complete listing of options.

Watch the video: How to Create an Index (1:46) [http://helpcentral.componentone.com/videos/D2H2009_Tour/CreatingAnIndex.html]

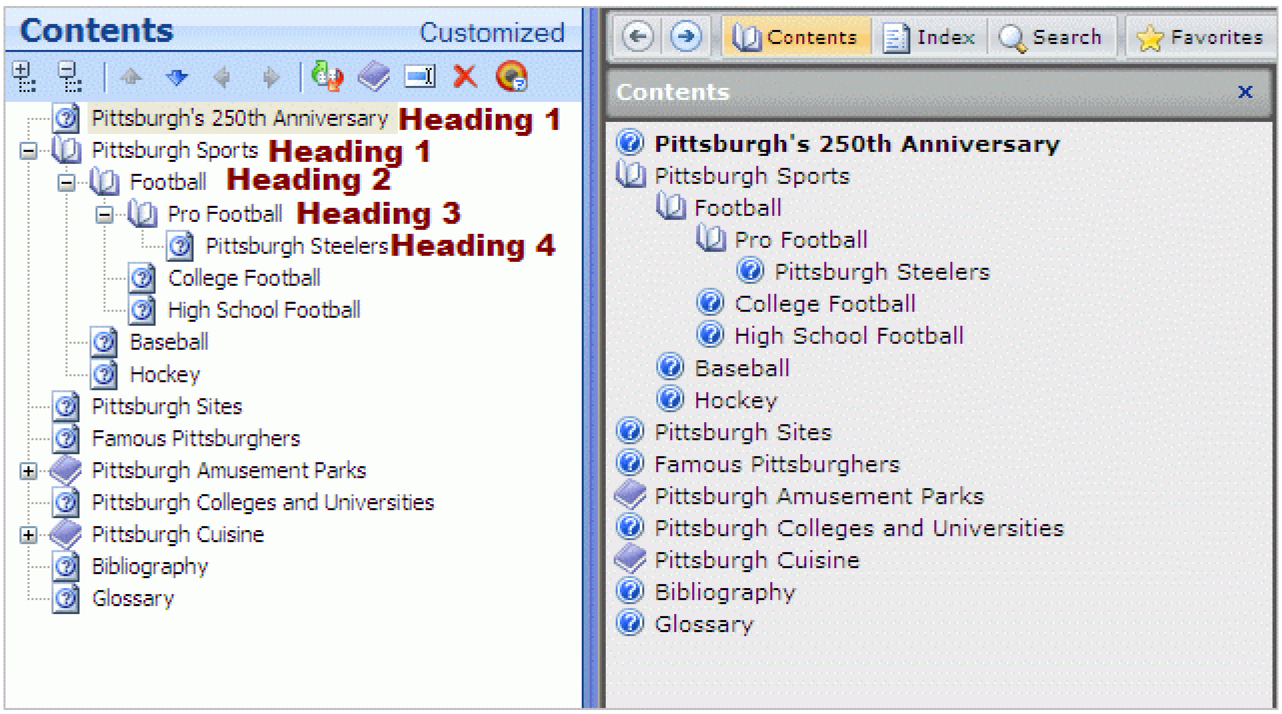
Creating a Table of Contents

Doc-To-Help automatically creates a Table of Contents (TOC) for you based on the structure of your documents. But you can create a custom TOC (you can even have different TOCs for different targets) if you wish.

An auto-generated TOC will be structured as follows:

- The order of the TOC will be determined by the order of the documents in the Documents pane (on page 108).
- Heading 1s with no Heading 2s under them will appear at the main level
- Heading 1s with Heading 2s under them will appear at the main level as "books," with the Heading 2s indented below them.
- If used, Heading 3s will be indented under Heading 2s; Heading 4s under Heading 3s.
- If a Heading 1 has no Heading 2s under it, it will not appear as a "book" in the TOC. It can be changed to one, but when the project is built it will not retain the book icon, because there are no subtopics under it.

The "book" icon visually indicates that "more information is available" about a topic.



By default, Headings 1-4 will be included in an auto-generated TOC. If you would like to add a Heading style to the TOC, clear the **Explicit access** check box for that Paragraph Style in the **Project Styles** dialog box. See Defining Character/Paragraph Styles and Topic Types on page 181 for more information.

You can see which topics are included in your table of contents by opening the Topics window (on page 112) and looking at the **TOC** column.

Please note: If your TOC is autogenerated, and you add a new topic to your project, that topic will automatically be added to the TOC the next time a Target is built.

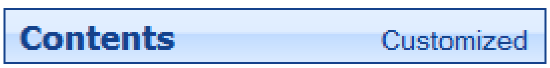
To create a custom Table of Contents

1. Open the Contents pane (on page 109).
2. Add, delete, reorder, and rename TOC items as you wish.
   - To add a topic to the TOC, drag it from the **Topics window** into the **Contents pane**.

- To move a TOC item up or down, choose it in the **Contents pane** and click the **Move Up** or **Move Down** buttons .
- To change the level of a TOC item, choose it in the **Contents pane** and click the **Move Out** or **Move In** buttons .
- To create a **new, empty book**, choose a spot in the **Contents pane** and click the **Create Book** button .
- To convert a TOC item to a **book** (it must have subtopics), choose it in the **Contents pane**, right-click and choose **Convert Topic to Book**.
- To rename a TOC item, choose it in the **Contents pane** and click the **Rename Topic** button .
- To delete a TOC item, choose it in the **Contents pane** and click the **Remove Topic** button .

As soon as you manually edit the TOC, it will be flagged as **Customized**. If you would like this table of contents to be exclusively used for the Target selected, click the **Target-Specific Table of Contents** button .

Once you have customized a TOC, any topics added to your project will not be automatically added to the TOC. They must be added manually.



Please note: In a Manual Target, topics in multi-topic documents will always appear in the TOC in the same order as they appear in the source document. If the order is changed in the TOC, Doc-To-Help lists an error in the build log and does not use the customization.

To create a target-specific Table of Contents

1. Open the Contents pane.
2. Choose the desired **Target** using the **Home tab** > **Select Target** button.
3. Manually edit the TOC.
4. Click the **Target-Specific Table of Contents** button .

Continue choosing Targets to customize each one.

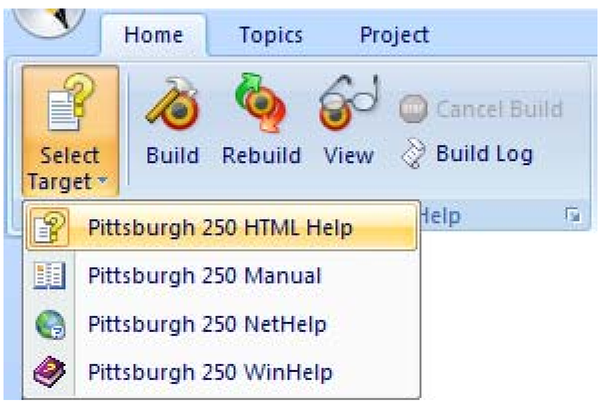
To rebuild the TOC based on document structure

1. Open the Contents pane.
2. If you have created Target-Specific TOCs, choose the desired **Target** using the **Home tab** > **Select Target** button.
3. Click the **Rebuild Table of Contents** button .

See Contents pane (on page 109) for a complete listing of TOC options and instructions.

Watch the video: How to Create a Table of Contents (1:40) [http://helpcentral.componentone.com/videos/D2H2009_Tour/CreatingATOC.html]

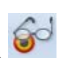
Building a Target



Before you begin a build, you should save and close your source documents.

When you are ready to build a target, you can do so using the Target ribbon group (on page 98) on the Doc-To-Help **Home** tab.

This ribbon group contains a drop-down that is used to choose the desired Target, as well the **Build Target**  and **Rebuild Target**  buttons. When rebuilding a Help target, Doc-To-Help empties its output directory and recompiles the entire project.

To view the Target, click the **View Target** button . After a build or rebuild, Doc-To-Help will also ask if you'd like to view the Target.

Use the **Cancel Build** button to stop a build, and the **Build Log**  button to view the Output window (on page 113). The **Output window** will display the build status and any unresolved links and errors.

If you have generated a Manual Target the **View PDF** button will be available. If you would like to view your PDF using this button, you should first select the **Generate PDF** check box for your Manual Target in the Help Targets (on page 155) dialog box.

Please note that when you generate a Manual target as a Word file, the resulting file will have a different toolbar than the Source documents. The **Doc-To-Help** toolbar or ribbon will contain only the basic styles and tools necessary to make final adjustments to your manual before printing or conversion to PDF. Tools include the **Cross-Reference** button (see Inserting Cross References on page 132 for more information) and the **Margin Notes** button (see Creating Margin Notes on page 131 for more information).

Watch the videos: How to Build a Target (1:53); How to Deliver a Target (1:01) [http://helpcentral.componentone.com/videos/D2H2009_Tour/BuildingATarget.html] [http://helpcentral.componentone.com/videos/D2H2009_Tour/DeliveringATarget.html]

Building Help Targets in Batch Mode

You may build a Help target in batch mode from the 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 are as follows:

Command	Description	Example
c1d2hbatch –build project target flags	Rebuilds the current Help target entirely.	c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp" -p
c1d2hbatch – make project target flags	Updates the current Help target.	c1d2hbatch -make "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual"
c1d2hbatch –compileall project target flags	Compiles all project files.	c1d2hbatch -compileall "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp"
c1d2hbatch –compile project target document_name flags	Compiles the specified document.	c1d2hbatch -compile "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide HTML Help" -p

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 to the Output window (on page 113):

```
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, etc.:

```
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 to the Output window (on page 113):

```
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
```

See Automating Doc-To-Help Project Builds in ComponentOne Help Central for more information.

Note: In Windows Vista, **C1D2HBatch.exe** must be “run as administrator," or it will always throw an exception. If you run it directly, start it with "Run as Administrator," available in Windows Vista in the context menu of the **C1D2HBatch.exe** file. If you run it from a command line prompt, open the command line window with "Run as Administrator." If you run it from a script or a program, make sure the script or program runs as administrator.

Doc-To-Help Reports

Doc-To-Help has nine reports that will help you organize your work and keep track of topics and index elements. These reports can be viewed, saved, and printed for maximum flexibility.

To run a report

1. Click the Doc-To-Help button .
2. Choose the **Reports** menu item, then the report name. The **Reports window** will open.

   From this window you can save, print, navigate, change views, and refresh a report. Use the **Two-Column Report** button (when available) to toggle a report from a two-column display to a one-column display. Once the **Reports window** is open, you can select another from the **Select Report** drop-down list.

Help Contents Listing

Displays the table of contents for the target chosen with all books expanded and all topics visible.

Help Index Listing

Displays all the project's index keywords and how many topics are assigned to each one.

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.

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.

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.

Script Listings

This report lists the code for each script defined in the project. Scripts are listed in alphabetical order.

Style Definitions

This report lists the properties of each style defined in the project. Styles are listed in alphabetical order.

Topic Detail Report

This report lists the properties for each topic in the project. Topics are listed in alphabetical order by title, and each topic starts on a new page.

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 their document name and style from which the topic was derived.

Working on a Team

Doc-To-Help Team Authoring makes it possible for Help authors to work together on a single 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 only available in Doc-To-Help Enterprise.

Team authoring is a basic source control feature in which authors work on 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. When a document is checked out and is being edited by one team member, it is locked so that others can't edit it at the same time. Until 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 viewed 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 Requirements

A Doc-To-Help project must satisfy the following team authoring requirements:

- All documents and auxiliary files, such as graphics, must be located within subfolders of the project folder. Source documents must be stored in a single folder, which is specified in the **Document folder(s)** field of the **Project Settings** dialog box (see Setting Project Properties on page 196); the default folder name is **Documents**. The graphics should be stored in a subfolder of the source documents folder. The graphics folder is specified in the **Media folder(s)** field; the default folder name is **Media**.
- 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 for setting up a team project is making it available to other users by uploading it to a central repository.

Note: Before setting up a Team Project, make sure your project satisfies all requirements described in Team Authoring Requirements (on page 241).

Doc-To-Help supports two repository types:

- **File System Repository** — A 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.
- **Web Repository** — A 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 determine the locations 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 **Doc-To-Help button** > **Team Authoring** > **Share Project for Team Authoring**. The **Share Project** wizard will open.
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. The **Respository Tests** dialog box will open.
8. Click **Close** to close the dialog box.
9. Click **Next** to continue.
10. In the **Repository Options** window, you can specify the settings regarding file backups:
    - **Limit the number of backups for each file —** When selected, 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 cleared, 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** — Select 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 and the limit of backups for that file has been exceeded.
    - **Ask user confirmation to delete backups —** Select 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** check box has been selected.

- The **Ask user confirmation to delete backups** check box is cleared by default; the team member will not be asked to confirm the backup cleanup; it is performed automatically without 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 (on page 254) for more information.

11. Click **Next** to continue.
12. Confirm your repository information and click **Next** to create the team project. The **Upload project** dialog box will open.
13. 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.

14. Click **Finish**.

Sharing a Project with a Web Repository

Sharing a project with a Web repository allows you to share your team 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 using IIS

1. To open the Computer Management utility, click the **Start** menu and select **Control Panel**.
2. Click **Performance and Maintenance** and then click **Administrative Tools**.
3. Double-click **Computer Management**.
4. Expand the **Services and Applications** node.
5. Expand the **Internet Information Services and Web Sites** nodes.
6. Right-click **Default Web Site** and select **New > Virtual Directory**. The **Virtual Directory Creation** Wizard opens.
7. Click **Next**.
8. Enter a name for your virtual directory in the **Alias** text box. This name will be part of the URL for your Web repository.
9. Browse to the physical directory on your machine that contains the content you want to share and click **Next**.
10. For the **Access Permissions**, make sure these items are checked: **Read**, **Run scripts**, **Write**, and **Browse**.

11. Click **Next** and then click **Finish** to complete the process. The virtual directory is now set up. Make sure the directory has Read, Write, and Browse privileges.

You may need to consult your network administrator to set up the virtual directory. Setting up a virtual directory will vary by operating system.

Once you have a virtual directory created, you can then 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 **Doc-To-Help button** > **Team Authoring> Share Project for Team Authoring**. The **Share Project** wizard will open.
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. 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, clear 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. Click **Close**.
9. Click **Next** to continue.
10. In the **Repository Options** window, the two checkboxes regarding file backups are selected by default:
    - **Limit the number of backups for each file** — When selected, 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 cleared, 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** — Select 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 and the limit of backups for that file has been exceeded.
    - **Ask user confirmation to delete backups** — Select 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** check box has been selected.
    - **The Ask user confirmation to delete backups** check box is cleared by default; the team member will not be asked to confirm the backup cleanup; it is performed automatically without 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 (on page 254) 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.

13. 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.

---

14. Click **Finish**.

Opening 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 to contribute to the project.

You can connect to a project that is stored in 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

1. Open Doc-To-Help.
2. Select **Doc-To-Help button** > **Team Authoring** > **Connect to Team Project**.
3. Under **Select repository type**, choose **File System Repository**. The team project is located on your organization's network.
4. Click **Next** to continue.
5. Enter the network location of the team project or click the **ellipsis** button to browse for the project.
6. Click **Next** to continue.
7. Enter a location for your working copy or click the **ellipsis** button to browse for a location. This is the project you will edit.
8. Click **Next** and confirm the working copy information.
9. 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.
10. Click **Close** and then click **Finish**. Your working copy of the project will open.

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

1. Open Doc-To-Help.

2. Select **Doc-To-Help button**  > **Team Authoring** > **Connect to Team Project**.
3. Under **Select repository type**, choose **Web Repository**. The team project is located on a Web server.
4. Click **Next** to continue.
5. 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.
6. If you are not using **Windows Integrated Authentication**, clear this check box and enter your login credentials. Note that these credentials must be entered before you try to browse the virtual directory.
7. Click **Next** and specify a location for the working project folder. This will be the location of the project in which you will work.
8. Click **Next** and confirm the working copy information.
9. 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. 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.
10. Click **Close** and then click **Finish**. Your working copy of the project will open.

Working with a Team Project

Three different file types make up a team authoring project:

- **Documents:** These are the source documents, the Word, HTML, and/or XHTML files included in your Doc-To-Help project. The file folder of these source documents is specified in the **Document folder(s)** field of the **Project Settings** dialog box (see Setting Project Properties on page 196); the default folder name is **Documents**. These files are accessible from the Team Authoring window (on page 114).
- **Auxiliary files:** These are any files (usually graphic files) located within the source file folders but are not documents. The file folder is specified in the **Media folder(s)** field of the **Project Settings** dialog box (see Setting Project Properties on page 196); the default folder name is **Media**.
- **Project file (.d2h extension):** This is the project itself, which contains all project information such as documents, topics, windows, table of contents, index, etc.

You can check out and work with all three file types. Please note that if a document or auxiliary file is checked out by an author, no one else can edit that file until those files are checked back in. If an entire project is checked out, no one else can edit the project or the files within it until the project is checked back in. You can check file status in the **Team Authoring** window (see Document and Auxiliary File Status on page 249 for more information). Click the **Refresh View** button in the Team Authoring tab, Documents ribbon group (on page 108) to update the display.

To begin working with a team project, first connect to it and download the project. See Opening a Team Project (on page 245) for details. (Once a project is downloaded, you should update your working copy before beginning work. See Synchronizing Your Team Project on page 249.)

After you have downloaded your project, there are several actions that can be performed on each file type.

Documents and Auxiliary files

- **Get Latest Version** — Use the **Get Latest Version** button of the Documents ribbon group (on page 108) of the Team Authoring tab to retrieve current versions of documents from the repository. This option is available for individual files, but you can check out the entire project (including the .d2h project file, and all documents and files) using the **Get Latest Version** button in the Project ribbon group (on page 107) of the **Team Authoring** tab. If any file was deleted from the team project by another user, getting the latest version of the project reflects that deletion in your working copy.

- **Check Out** — Checking out a file 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 files in the **Team Authoring window** are read-only, except for newly added files. To check out a document, select it in the Team Authoring window, click the **Team Authoring tab** and click the **Check Out** button from the Documents ribbon group (on page 108). The check out status is shown by an icon and status change in the Team Authoring window. If you change your mind and do not want to modify a file, you can undo the check out command by clicking the **Undo Check Out** button in the Documents ribbon group of the **Team Authoring** tab.
- **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 it is checked back into the repository.
- **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. If you would like to change the text and highlight color of new project items, from the **Doc-To-Help button**, click the **Doc-To-Help Options** button, then the **Team Authoring** tab.
- **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.
- **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 In** button in the Documents ribbon group (on page 108) of the **Team Authoring tab**, but recommended practice is to check in all of your changes to the project, including all documents and files, using the **Check In** button in the Project ribbon group (on page 107) of the **Team Authoring tab**.

Auxiliary files are checked in and out the same way as documents. Unlike documents, you don't need to add them to the project explicitly; they are added automatically once they are created in the source file folder. If you can't see your auxiliary files in the Team Authoring window, they may not be stored in the **Documents** folder within your project.

Project files

- **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. To check out a project, click the **Team Authoring tab** and click the **Check Out** button from the Project ribbon group (on page 107).
- **Check In** — Checking a project in uploads any documents that have been changed, along with the project file, into the repository. To check in a project, click the **Team Authoring tab** and click the **Check In** button from the Project ribbon group (on page 107).
- **Get Latest Version** — Getting the latest version of the project downloads the team project to your machine and updates your working copy. To get the latest version of a project, click the **Get Latest Version** button in the Project ribbon group (on page 107) of the **Team Authoring** tab

Working with Project files (.d2h files) in team authoring is more complex than working with documents. See Checking Out a Project vs. Individual Documents (on page 247) for more information.

Checking Out a Project vs. Individual Documents

When you check out Team Project document and auxiliary files, those files are unavailable to others until you check them back in. If you check out the entire project (which includes the .d2h project file, and all documents and files), then no one on your team can make any changes at all until you check the project back in.

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:

- **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. 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 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.

If you need to build your Help Target while working in a Team Project, you can do so by checking the entire project out, or by temporarily disabling team authoring control. See Temporarily Disabling Team Authoring Control (on page 248) for details.

Temporarily Disabling Team Authoring Control

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 can temporarily disable team-authoring control for the duration of a 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; 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 attempt to build a Help target without checking out the project, Doc-To-Help displays a dialog box suggesting you disable team-authoring control for the duration of the build. If you choose to do so, 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.

You can also temporarily disable team authoring control by clicking the **Team Support** button in the Project ribbon group (on page 107) of the **Team Authoring** tab. Click it again after the build to enable team support once again.

If your intention is only to see the Help target built from your project, you can use this option to 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. In that case, you will be able to build Help targets without any restrictions.

An administrator may remove team authoring support from a project permanently; see Removing Team Authoring Support from a Working Copy on page 253.

Document and Auxiliary File Status

The **Team Authoring window** will display the documents and auxiliary files in your Team Project. The information displayed for each document includes document status, size, modification date, 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; in addition, a document that has been checked out will have a checkmark icon next to it in the **Team Authoring** window and **Documents** pane.

Possible status values include:

- **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 has not yet been checked into 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, click the **Refresh View** button in the **Team Authoring** tab, Documents ribbon group (on page 108).

Synchronizing Your Team Project

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. When you choose either of the options below, the **Get Latest Version** dialog box opens. See **Get Latest Version dialog box** (below) for more information.
  - **To update individual files**, use the **Get Latest Version** button of the Documents ribbon group (on page 108) of the Team Authoring tab to retrieve current versions of documents from the repository.
  - **To check out the entire project** (including the .d2h project file, and all documents and files) use the **Get Latest Version** button in the Project ribbon group (on page 107) of the **Team Authoring** tab.
- **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.

- **To upload changes to individual files**, use the **Check In** button on the Documents ribbon group (on page 108) of the Team Authoring tab.
- **To upload changes to the entire project,** use the **Check In** button on the Project ribbon group (on page 107) of the Team Authoring tab.

Notes on uploading changes to the repository

If conflicts are encountered on upload (**Check In**), they may be resolved in the **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.

Get Latest Version dialog box

The **Get Latest Version** dialog box provides the following information about the project's files:

Local Status — This 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** check box is selected; 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 — This 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** check box is selected 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** check box is selected 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; since it can disrupt team-authoring control and cause data loss.

Merge Action — This 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 **D2HTempFolder** 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. Be aware that once you get the latest version of a project, all files are considered synchronized with the repository; you will not see the files in the list of the repository-changed files unless they were changed after you retrieved the latest version of the project from the repository. If necessary, you can retrieve these files from the repository by selecting the **Show all files** check box.

Team Authoring Administrative Functions

A special administrative utility is included when Doc-To-Help is installed. This utility, **C1D2HTeamAdmin.exe**, should be used only by a Doc-To-Help administrator when it is necessary to perform certain actions, such as removing backup files or unlocking files from the repository, upgrading a team-authoring project, or removing team-authoring support from a working copy of a project.

This **C1D2HTeamAdmin.exe** can be found in \\Program Files\ComponentOne\DocToHelp, where it is installed, by default, with Doc-To-Help.

To access the administrative utility, the administrator can locate and double-click the **C1D2HTeamAdmin.exe**. After the utility opens, use the menu and toolbar to perform the necessary actions.

Removing Backup Files from 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. 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 **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.

To delete up backup files in the repository:

1. Double-click C1D2HTeamAdmin.exe (on page 251).
2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
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. Select **Delete all backup files older than** in the lower part of the dialog box.

7. Click the drop-down arrow next to the date and select a date from the calendar. 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.

The administrator can also use this dialog box to change the backup settings specified when the project was first shared. See Changing Repository Settings on page 254.

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 during a 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 **C1D2HTeamAdmin.exe**. The administrator should run the utility from any machine that can connect to the repository.

To unlock files in the repository

1. Double-click C1D2HTeamAdmin.exe (on page 251).
2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
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 choose **Unlock** from the right-click 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 .dhv file in the repository:

1. Double-click C1D2HTeamAdmin.exe (on page 251).

2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
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. Click the **Version info** node on the left and select the **<project name>.dhv** file on the right.
6. Click the **Unlock** button on the toolbar or choose **Unlock** from the right-click 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 (on page 251).
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 (usually done temporarily - on page 248 - when building a Help target). 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 tab** and **Team Authoring window**, use **C1D2HTeamAdmin.exe**.

To remove team-authoring support from your working copy of the project

1. Double-click C1D2HTeamAdmin.exe (on page 251).

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 **C1D2HTeamAdmin.exe**.

To change the repository settings

1. Double-click C1D2HTeamAdmin.exe (on page 251).
2. Select **File** > **Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
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 will open.
   - **Limit the number of backups for each file**: Clear this check 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, select this check box and specify the **Number of backups kept** in the repository.
   - **Allow users to delete old backups in check-in**: Select this check 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**: Select this check 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** check box has been selected.
   - If the **Allow users to delete old backups in check-in** and **Ask user confirmation to delete backups** check boxes are both selected, team members will be prompted with a dialog box when checking a file into the repository once the backups limit has been exceeded.
   - The **Ask user confirmation to delete backups** check box is cleared by default, meaning the team member will not be asked to confirm the backup cleanup; it is performed automatically without any notification. Therefore, a dialog box will not appear when **Allow users to delete old backups in check-in** is selected and the **Ask user confirmation to delete backups** is not checked.

- If the **Allow users to delete old backups in check-in** checkbox is cleared, when a team member checks a file into the repository and the limit of backups for the file has been exceeded, a dialog box will appear. 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.

Creating a Modular Help System

A Modular Help system is necessary when you have a collection of several different Help files (and would like to keep them that way) but would like them to appear to the end user as a single Help system.

There are four common scenarios necessitating a Modular Help system:

- **Modular software** — Your software application is sold as separate modules with the user purchasing one or more modules at a time. A modular Help system will contain all the Help files, but only the appropriate Help files will accompany the purchased modules.
- **Large documentation sets** — If you wish, you can "chunk" information into several small Help systems and create a modular system rather than deploy one large Help system. Smaller projects enable streamlined updating and easier distribution.
- **Help systems with future Help projects planned** — If you plan to release your Help system in stages, you can pre-position placeholders for future Help projects before they are released. Instead of distributing the entire Help system each time you add to the system, you only need to distribute additional Help files. If you didn't plan for an addition, you can distribute a new hub Help file along with the new Help files.
- **Documentation teams sharing responsibilities across one large project** — Modular Help can be a solution for a large project with many authors. Each Help author is assigned one or more projects. The team leader is usually charged with the responsibility of maintaining the hub Help project and assembling the entire project. (You may want to check out Doc-To-Help's Team Authoring capabilities and use those instead. See Working on a Team on page 241 for more information.)

Modular Help systems can reference Help files that are not installed (for example, Help for a software module the end user has not purchased) and still look seamless. The table of contents and index will simply omit the missing information without displaying error messages. If the user installs the module in the future, the Help will be added to modular system in the proper position.

Note: Verify that your software can accommodate context-sensitive calls from multiple .hlp or .chm files **before** creating a modular Help project. If your software can only accommodate context-sensitive calls from one Help file, you can still create a modular system, but in that case you must place all context-sensitive topics into one Help file.

Modular Help System Deliverables

Modular Help systems are comprised of one hub (or master) Help file and any number of child Help files. The modular system appears to the user as one integrated Help system, no different than a large single Help file system. The contents and index are merged at run time, when the Help file is opened. Dynamic links are also created at run time.

The following files are distributed to end users. When you are ready for release, these files should be stored in the **Redistributables** folder of your modular help project. See File Organization (on page 259) for more information.

HTML Help

- Hub project .chm file
- All child .chm files

NetHelp

- The entire contents of the hub project NetHelp folder. Since NetHelp is uncompiled HTML Help, the number of files will vary. By default, click on the **default.htm** file to open the project. (This file name can be renamed in the **Default name** field of the **Help Targets** dialog box.)

WinHelp

- Hub project .cnt file
- Hub project .hlp file
- All child project .cnt files
- All child project .hlp files

Note: When creating a hub project, avoid including spaces in any of the file names. Using spaces in file names will disable the next/previous functionality of the modular Help system. If your .D2H project file contains spaces you can fix this without renaming the project by specifying a name with no spaces in the **Base name** field in the **Help Targets** dialog box. See Creating Help Targets (on page 155) for more information.

Standardizing Modular Help

In order for your modular project to mesh logically, it's important that all the pieces adhere to a set of standards. You may want to create standards for the following:

- Source and Target templates and/or cascading style sheets
- Styles, and how they are used
- Help window definitions (each project should use the same Windows)
- Indexing conventions (keywords and groups)
- Use of graphics and graphics format
- Tone and style of writing
- Map numbering for context-sensitive help

If you are using a centralized glossary, each project should not have terms in its local glossary; all items should be stored in a central glossary (in the hub project).

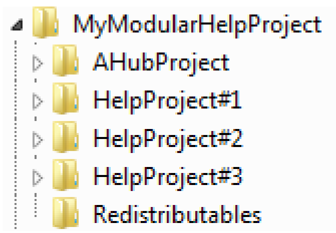
File Organization

Good file organization makes it easier to manage your modular hub project and avoid errors.

Your file system should include:

- A folder for the entire system
- A subfolder for the hub project
- A subfolder for each child help project
- A subfolder to store the end-user project redistributables

For example:



If your project has multiple authors, it is best to store the project on a network drive.

Creating a Hub Project

The hub (or master) project ties all of the child projects together. It should be a small project, but can contain information that does not change often (such as a welcome message and company contact information) if you wish.

Before creating your project, set up your file system for the entire modular system. See File Organization (on page 259) for more information.

If you create a new child project in the future that is not part of your original hub project, simply add it to the hub project, and distribute both the new hub project and child project to the end user.

To set up a hub project

1. Create your hub project. The project name should contain no spaces.
2. Open the **Project Settings** dialog box. (See Setting Project Properties on page 196 for more information.)
3. Select the **Is modular hub project** check box to make this a Modular Hub Project.
4. Close the **Project Settings** dialog box.
5. Open the hub project document and create placeholder topics for each of the child files you wish to associate with the hub. There are several ways to structure this file, but you should only choose one. (The placeholder topics will later be mapped to the child topics. This is done by editing the **Topic Properties**.) Document Structure options:
   - Heading 1 placeholder topics only
   - One Heading 1, followed by placeholder topics (Heading 2's)
   - One Heading 1, followed by content in Body Text, then the placeholder topics (Heading 2's)
   - One Heading 1, followed by one Heading 2 and Body Text content, then the placeholder topics (Heading 2's)
6. To disable any automatic related links that may be displayed in the hub project, in the Topics tab, Related Topics ribbon group (on page 105), select the **Hide Subtopic Links** check box for each topic.
7. Click the **Build Target** button to build the file.
8. Add at least two index entries (one must be a keyword) to the project.

9. Open the **Help Targets** dialog box. (See Creating Help Targets on page 155 for more information.)

10. Choose the Help Target (can be HTML Help, NetHelp, or WinHelp) on the left. The hub project, and all of the child projects, must use the same Help Target.

11. Select the **Binary Index** check box. (HTML Help only)

12. Clear the **Binary TOC** check box. (HTML Help only)

13. Click the **Build Target** button to build the file.

To map placeholder topics to child topics

1. Verify that all of your child projects are in the proper place in your modular system. See File Organization (on page 259) for more information.

2. Open your hub project.

3. Select a placeholder topic in the **Topics** window. Right-click on it and choose **Properties**. The **Topic Properties** dialog box will open.

4. Edit the **Topic Properties**. The properties changed will vary depending on the target of the entire modular project.

   - For **HTML Help** targets, you need to specify the **Module File** (.chm) and **Contents file** (.hhc) fields of the child project for the placeholder topic. (The .chm and .hhc can be found in the child project's HTMLHelp folder.)
   - For **NetHelp** targets, you need to specify the **Module file** (.npj) field of the child project for the placeholder topic. (The .npj can be found in the child project's NetHelp\ProjectInfo folder.)
   - For **WinHelp** targets, you need to specify the **Module file** (.hlp) and **Contents file** (.cnt) fields of the child project for the placeholder topic. If you wish, you can use the **Title** field to enter a different name for child Help file (useful if two projects have the same name). (The .hlp and .cnt files can be found in the child project's Help folder.)

5. Map all additional placeholder topics.

6. Rebuild the project by clicking the **Rebuild** button. Click the **View Target** button to view the project.

   Doc-To-Help automatically creates the TOC structure based on the order of the placeholders in the hub source document. Each placeholder topic will be a "book" in the hub project TOC.

Note: For **HTML Help** and **WinHelp** projects, you will need to place copies of the child project files (.chm for HTML Help; .hlp and.cnt for WinHelp) in the same folder as the hub project files to view the complete modular project. Whenever you do a full **Rebuild Target** (as opposed to a **Build Target**) those child files will be overwritten in the hub project folder and must be replaced.

Note: For **NetHelp** projects, all projects in the modular help system must be created in the same version of Doc-To-Help.

Creating a Child Project

Child projects are tied together with the hub project (see Creating a Hub Project on page 259). There are no special mappings in child projects, just a few settings and other options that need to be handled. You can create as many child projects as needed.

Before creating your child projects, set up your file system for the entire modular system. See File Organization (on page 259) for more information.

To set up a child project

1. Create your child project. The project name should contain no spaces.
2. Click the **Build Target** button to build the file.
3. Open the **Help Targets** dialog box. (See Creating Help Targets on page 155 for more information.)
4. Choose the Help Target (can be HTML Help, NetHelp, or WinHelp) on the left. The hub project, and all of the child projects, must use the same Help Target.
5. Clear the **Binary Index** check box.
6. Clear the **Binary TOC** check box.
7. Click the **Build Target** button to build the file.

Cross-project links may be created between child projects. See Cross-Project Linking on page 261.

Child projects using automatic context ID creation need to have offsets added, see Modifying Context IDs in Modular Projects on page 262.

Glossaries may be used in Modular hub projects, but HTML Help and WinHelp targets require a few adjustments to avoid duplicate items. See Modular Help System Glossaries on page 262.

Cross-Project Linking

It is possible to create links from one Help file to another in a Modular Help system. Regular topic links should be used for links within each Help file.

To create cross-module links

To create a link from **HelpProject#2** to **HelpProject#1**.

1. Open HelpProject#1.
2. Create a group named **Modules**. Assign a topic (the one you wish to link to) to that group. (See Index and Groups pane on page 110 to learn how to create a group.)
3. Build the Help Target. Close the project.
4. Open **HelpProject#2**
5. Create a group named **Modules**. This group will be empty (no topic will be assigned to it.)
6. Open a document, and select the desired link text. Create a link to the **Modules** group (see Creating Links on page 212).
7. Build the Help Target. Close the project.
8. Open the hub project.
9. Build the Help Target. The link from **HelpProject#1** to **HelpProject#2** will work.

Note: You will need to create a Group name for each cross-project link pair you create.

Modifying Context IDs in Modular Projects

To avoid duplicate context IDs when using automatically generated context IDs, you can customize the context ID numbers in each child project. This will guarantee that your context-sensitive help will work properly for all of the child projects within your Modular hub project.

To customize context IDs

1. Open a child project.
2. Open the **Project Settings** dialog box. (See Setting Project Properties on page 196 for more information.)
3. Enter a number in the **ID Offset** field. For example, you could make this 1000; then 1000 will be added to each automatic ID generated.
4. Click the **Build Target** button to build the file. Close the project.
5. Open another child project.
6. Open the **Project Settings** dialog box. (See Setting Project Properties on page 196 for more information.)
7. Enter a different number in the **ID Offset** field (for example, 2000).
8. Click the **Build Target** button to build the file. Close the project.

   Repeat for all child projects.

Modular Help System Glossaries

Glossaries may be used in Modular hub projects, but HTML Help and WinHelp targets require a few adjustments to avoid duplicate items in the hub Table of Contents and Index.

In NetHelp projects, all child project glossaries are merged in the hub and duplicate glossary entries are automatically purged.

To work with Glossaries in HTML Help and WinHelp files

1. Verify all child Help projects, and the hub project contain the same Glossary document file.
2. Open a child project.
3. Open the **Help Targets** dialog box. (See Creating Help Targets on page 155 for more information.)
4. Select the **Skip Glossary** check box.
5. Click the **Build Target** button to build the file.
6. Repeat for all child projects.

Modular TOC Utility

One issue with modular HTML Help is that the child projects do not contain the entire table of contents. Generally, this limitation is not a problem because the 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 use of 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 **Modular TOC Utility** removes the table of contents limitation by automatically adding the full table of contents to each Help file in the 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.

To create a modular TOC for HTML Help Targets

Verify that you have a fully defined modular Help system before continuing with the following steps.

1. Open the Modular TOC Utility by selecting **Start** > **All Programs** > **ComponentOne Doc-To-Help** > **Modular TOC Utility**. The **Modular TOC** Utility will open.
2. Select the **Create a new Modular TOC** file option and click **Next**.
3. After reading the dialog box regarding the location of the new Modular TOC file, click **OK**.
4. Browse to the directory where you want to store your .hub file. (You may want to use your **Redistributables** folder for this file. See File Organization on page 259 for more information.)

   Note: The Modular TOC Utility will overwrite duplicate .hub files.
5. Provide the Modular TOC Utility with a name for your .hub file and click **Save**.
6. The Modular TOC Utility prompts you for the hub .hhp file. **Browse** and select the file, then click **Open**. (This file is located by default in the **HTMLHelp** sub-directory of your hub Help file.)

   The Modular TOC Utility will list all the child **.**hhp files in your modular Help system.
7. Choose the first child .hhp file, then click **Next**. Choose the location of the appropriate .hhp file and click **Open**. (This file is located by default in the **HTMLHelp** sub-directory of your child Help file.)
8. Repeat until each of the child .hhp files has been located.
9. Click **Next**. The Modular TOC Utility will then prompt you to verify that the path for the build project.
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.

    After the files are built, the Modular TOC Utility Editor will open.
15. Close the Modular TOC Utility Editor and examine the Help files in your build directory. Note that each Help file (hub and child) will include the entire table of contents.

Documenting Your Class Library with Microsoft® Sandcastle

Microsoft's Sandcastle utility automatically creates MSDN formatted reference documentation from .NET assemblies and XML comment files. This makes it possible for you to document a class library — plus add narrative information — quickly and efficiently. Doc-To-Help integrates Sandcastle's XML output into your projects, automatically creating topics, index, TOC, and other Help elements. With this information added to your project, you can edit/add your own topics, and easily link to namespaces.

Doc-To-Help is the first Help Authoring tool to integrate with Microsoft Sandcastle. This integration extends the functionality of the Sandcastle plug-in by providing an interface for Sandcastle, and giving authors more power to connect narrative and reference content.

This version of Doc-To-Help supports Sandcastle version 2.4.10520.1, which is available for download here: http://www.microsoft.com/downloads/details.aspx?FamilyId=E82EA71D-DA89-42EE-A715-696E3A4873B2&displaylang=en.

If you have reference documentation originally created in Doc-To-Help's Documenter for .NET, you can convert those projects to Sandcastle, see Converting Documenter for .NET projects to Sandcastle Projects (on page 266).

Creating a Project with Sandcastle

There is a sample project you can test drive in **\\Users\<username>\Documents\My Doc-To-Help Projects\Samples\MSSandcastle** (Windows® Vista); **\\Documents and Settings\<username>\My Documents\My Doc-To-Help Projects\Samples\MSSandcastle** (Windows XP).

To create a new Sandcastle project

1. Create a new Doc-To-Help project using the **New Project Wizard** (**Doc-To-Help button** > **New Project**).
2. In Doc-To-Help, click the **Project tab** > **Project Styles button**. The **Project Styles** dialog box will open. For **Paragraph Styles** Heading 1-Heading 5, change the **Topic Type** to **Conceptual**. (See Defining Character/Paragraph Styles and Topic Types on page 181 for more information.)
3. Click the **Project tab** > **Plugins button** and choose **Create New Plugin Document** to add a new Sandcastle plug-in document.
4. Enter a name, such as **Reference**, for the .xmlprop file in the **XMLDocuments** subdirectory.
5. From the **Choose Plugin Type** dialog box, select the **Sandcastle plugin type**. Click **OK**, and a new document will be created, **XMLDocuments\Reference**.

6. Double-click on **Reference**. The Properties for Generating Reference dialog box (on page 267) will open. Select the assemblies you want to document and the source XML comment file(s) for them. You can also specify the framework version, the types and members to include, and the syntax to include.
7. Right-click on **Reference** in the Documents pane and choose **Generate**. This will generate the Reference.xml file.
8. Open the file your narrative will reside in. (If it is not using the **C1_sandcastle_src.dot** template, you should apply it now. See Using styles to create links from narrative to reference text (on page 281) for more information. Add text. If you would like to create a link from the narrative document to the reference document, highlight the text to link and select the appropriate toolbar button or style.
9. In Doc-To-Help, click the **Home tab > Select Target button** to choose the Help target. Click the **Rebuild Target** button.

   The target will be formatted with the MSDN reference style.

Converting Documenter for .NET projects to Sandcastle projects

To convert a legacy project

1. Make a backup copy of your entire project before conversion.
2. Open the Doc-to-Help project (.d2h)
3. Right-click the Word document generated by Documenter for .NET and choose **Remove Document** to remove it from the project. Delete the Word document file from your project folder.
4. Using the **Project Styles** dialog box (see Defining Character/Paragraph Styles and Topic Types on page 181), delete the character styles that are used for creating links from narrative to reference. They are: all Link ... styles (Link Class, Link Constructor,..., Link Type - 15 styles) and two additional styles: "MSDN Link style" and "Block Key." (17 character styles total). Be careful not to delete any other styles.
5. In all narrative Word source documents change the attached template from C1H_dotnet_src.dot to C1H_sandcastle_src.dot.
6. Add the plugin document: in Doc-To-Help choose **Project tab > Plugins button** and select **Create New Plugin Document**. Enter its name in the **New Plugin Document** dialog box in the **XMLDocuments** folder ("Reference" is the suggested name, but any name is OK).
7. Double-click the plugin document, which will open the **Properties for Generating Reference** dialog box.
8. Select **Import**, click the **Browse** button, and select the **Documenter for .NET** project file (.d2hdotnet) that you are converting.
9. This is the main point in the conversion process: the converter scans your Documenter for .NET project (.d2hdotnet) and converts its settings to properties of the plugin document (.xmlprop).
10. When the conversion is finished, check that everything you need is converted (double-click on the plugin document to open the **Properties for Generating Reference** dialog box):
    - List of assemblies and XML files, see the **Assemblies** node
    - XSLT transformation applied to the XML file, see the **Generation** node, **XSLT file to transform source code XML** field**.**
    - Elements excluded from documentation, see unchecked nodes in the tree in the **Element filter** node.
11. Right-click on the plugin document and choose **Generate** from the menu.

12. In Doc-To-Help, click the **Home tab > Select Target button** to choose the Help target. Click the **Rebuild Target** button.

    The target will be formatted with the MSDN reference style.

Note: Links from narrative to reference created using the "Link ..." styles, such as Link Class, Link Method, Link Property, and so on, should work after this conversion. But links created with C1H Jump and Link Tag styles will probably not work, because the titles of generated topics and therefore their linktags have changed (and these two styles have a linktag explicitly specified in the hotspot itself).Those links must be fixed manually.

Configuring the Sandcastle Reference Section with the Properties for Generating Reference dialog box

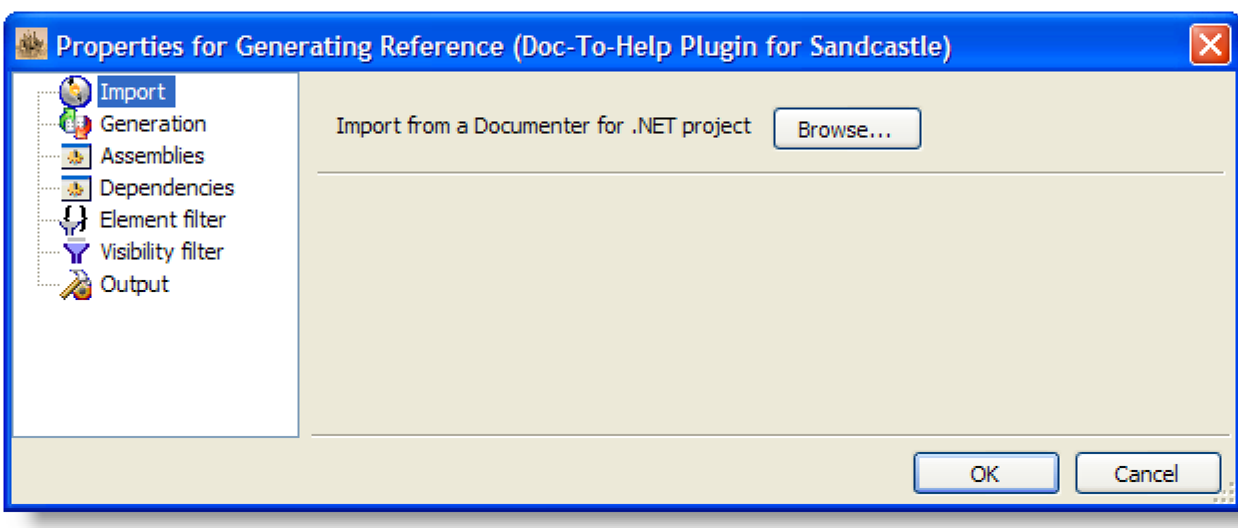
In the **Properties for Generating Reference** dialog box, you can import a **Documenter for .NET** project file and/or configure your Sandcastle reference section.

To access the **Properties for Generating Reference** dialog box, navigate to the **Documents** pane and double-click the reference document.

The dialog box is broken into seven nodes: **Import**, **Generation**, **Assemblies**, **Dependencies**, **Element filter**, **Visibility filter**, and **Output**.

Import node

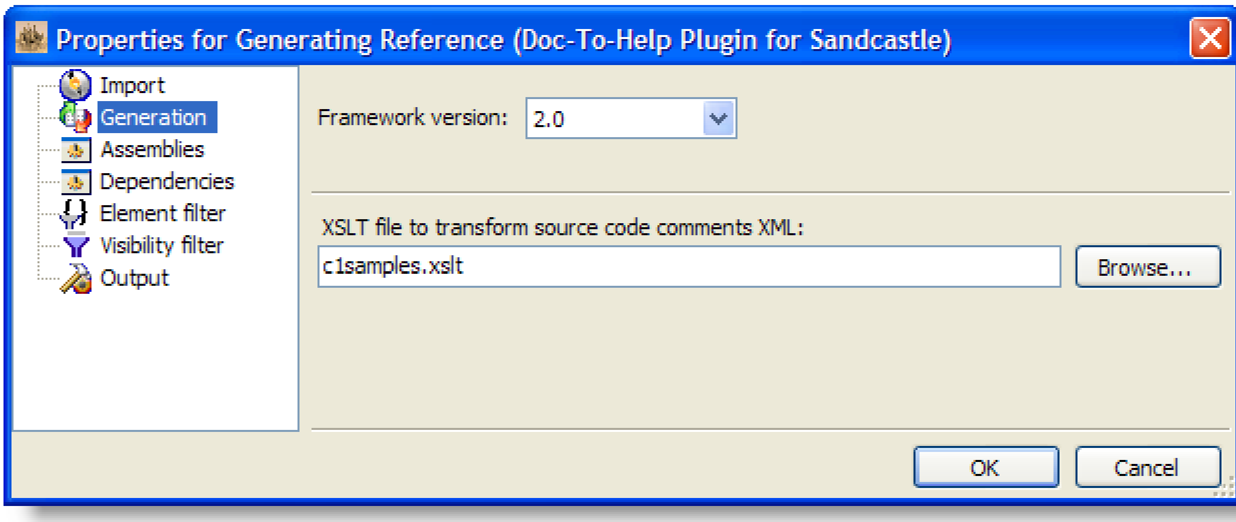
The **Import** node allows you to import a Documenter for .NET project — including all of its properties — which will overwrite the default properties for the Sandcastle plug-in document.



To import a Documenter for .NET project, click **Browse**. The **Open** dialog box will appear, allowing you to navigate to your project folder and select the **.d2hdotnet file** you wish to convert to Sandcastle.

Generation node

In the **Generation** node, you can choose the framework version for your system type documentation links and add an .xslt stylesheet to your project.

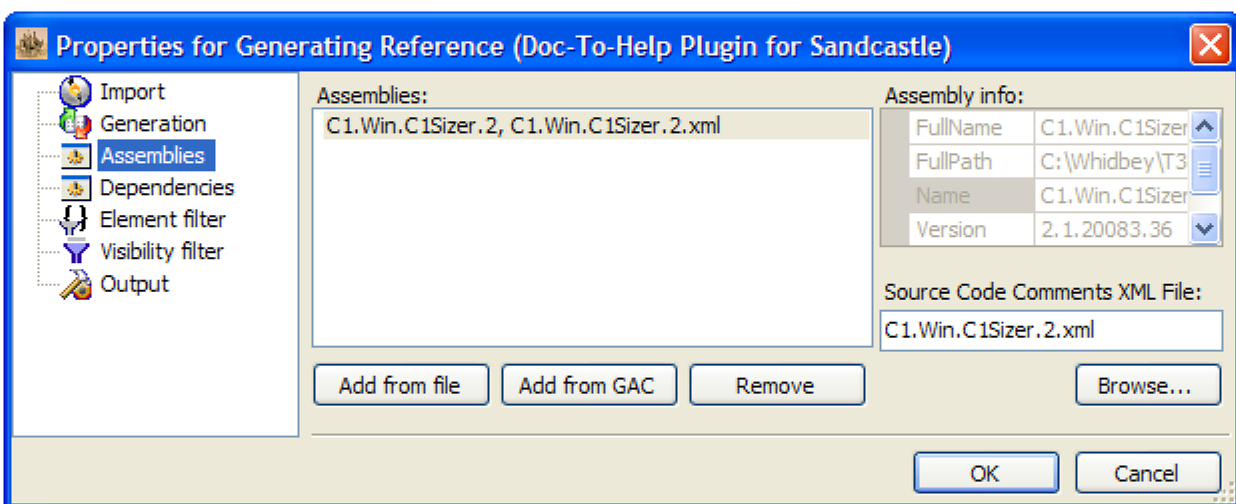


Framework version: Select the .NET Framework version you want to link to when a system type appears in your documentation. Select version 1.0, 1.1, or 2.0.

XSLT file to transform source code comments XML: Click **Browse** to select the .xslt file containing the transforms that are used to manipulate your .XML source code comments after a build has already been performed and topics have been created.

Assemblies node

Use the **Assemblies** node to add assembly files and source code comment files to your project.



Add from file: Click this button to add an assembly from a .dll on your machine.

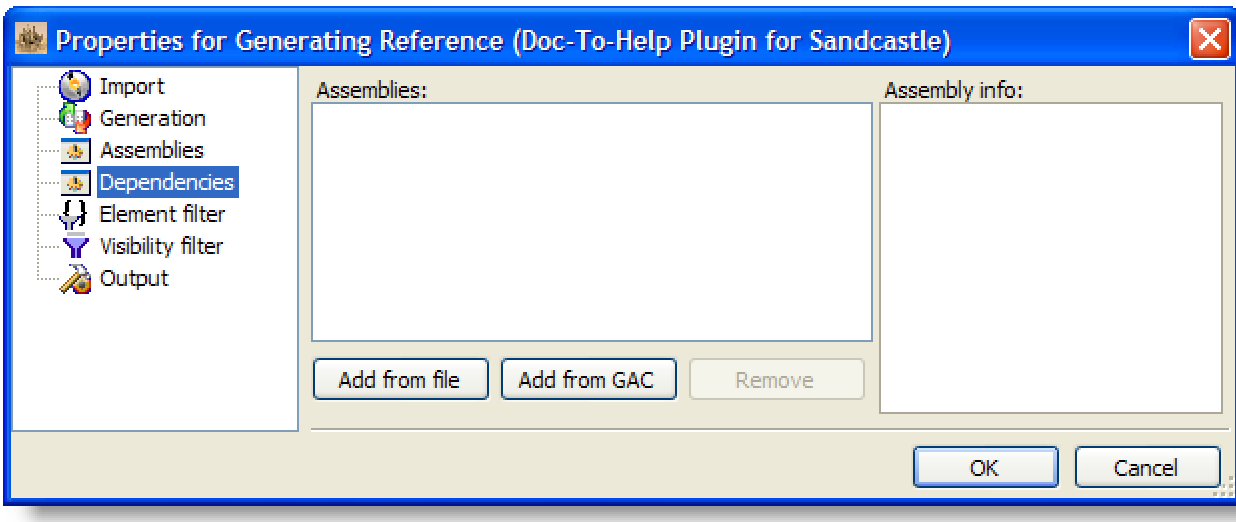
Add from GAC: Click this button to add an assembly from the Global Assemblies Cache (GAC). Strong-named assemblies must reside in the GAC in order to be documented.

Remove: Click **Remove** to remove an assembly and its corresponding .xml file from the project.

Source Code Comments XML File: Click **Browse** to locate the .xml file containing source code comments, or the actual content, for the reference section of your documentation.

Dependencies node

The dependency assemblies are required by the assemblies specified in the **Assemblies** node.



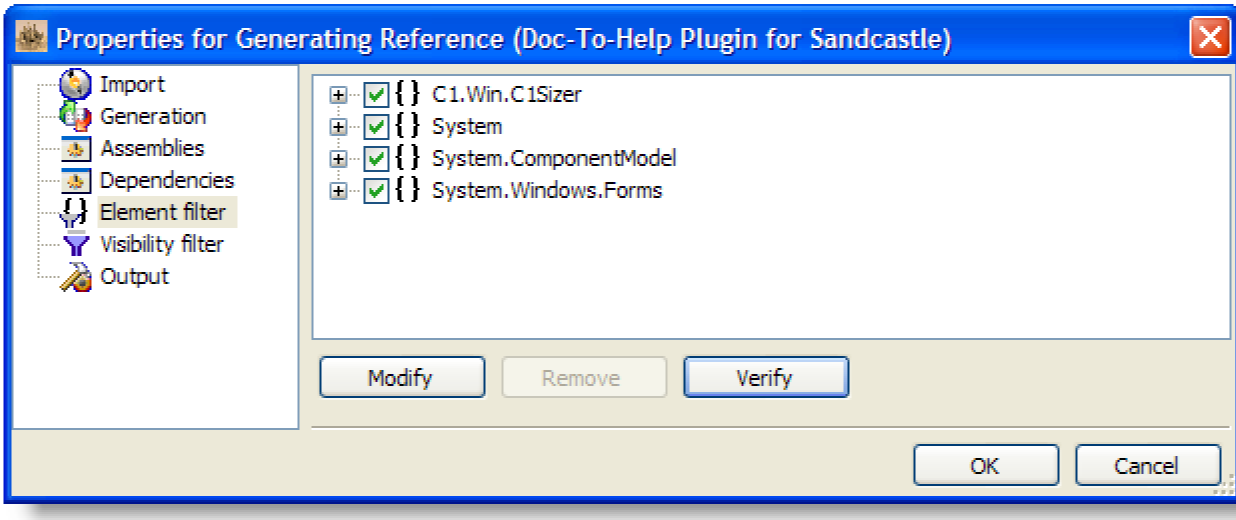
Add from file: Click this button to add an assembly from a .dll on your machine.

Add from GAC: Click this button to add an assembly from the Global Assemblies Cache (GAC). Strong-named assemblies must reside in the GAC in order to be documented.

Remove: Click **Remove** to remove an assembly and its corresponding .xml file from the project.

Element filter node

The **Element filter** allows you to select which elements you want to appear in the documentation. To include an element in your reference section, simply select the check box next to it.

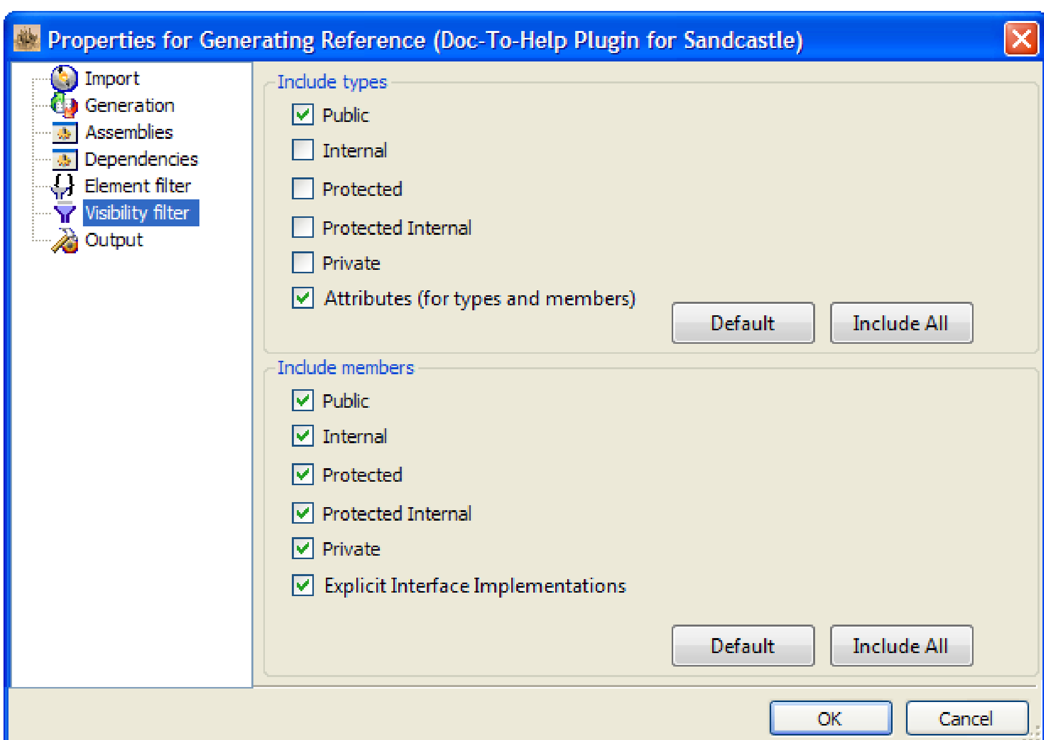


Modify: Click this button to access the **All Elements** dialog box. From the **All Elements** dialog box, you can easily add elements to the **Element filter** by clearing their check boxes. You can also click the **Exclude Elements Without Description** button to remove any elements that don't have descriptions from your project.

Verify: Clicking this button will initialize a verification test, which will check whether or not all of the filter elements specified exist in the assemblies.

Visibility Filter node

The **Visibility filter** determines which types of classes and members will appear in the Sandcastle reference section. The Visibility filter is separated into two sections: **Include types** and **Include members**.



Include Types Section

In the **Include types** section, select the check boxes for the types of classes you would like to appear in your reference section. You can include the following class types in your reference section: Public, Internal, Protected, Protected Internal, Private, and Attributes.

To revert to the default selection, which only includes Public and Protected classes, click the **Default** button

To include all types in the reference, click the **Include All** button.

Include Members Section

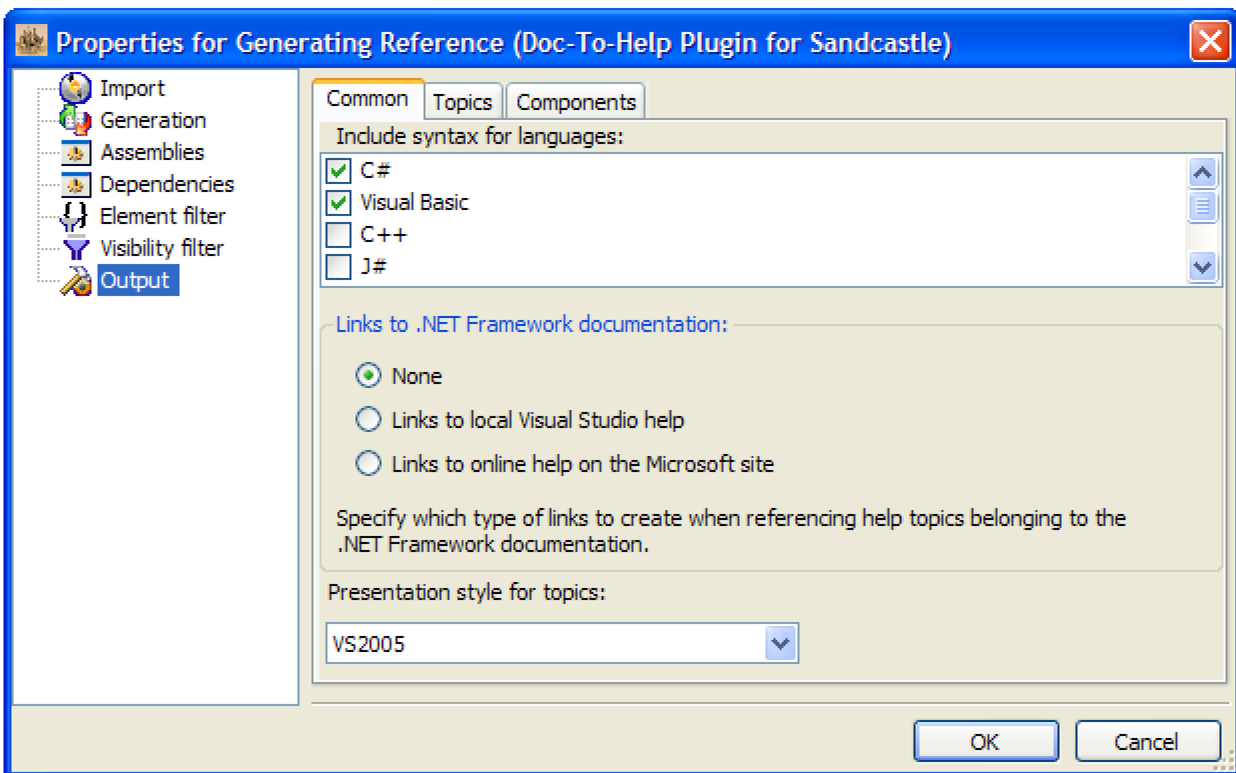
In the **Include members** section, select the check boxes for the members that you would like to appear in your reference section. You can include the following types of members in your reference section: Public, Internal, Protected, Protected Internal, Private, and Explicit Interface Implementations.

To revert to the default selection, which only includes Public and Protected members, click the **Default** button

To include all members in the reference, click the **Include All** button.

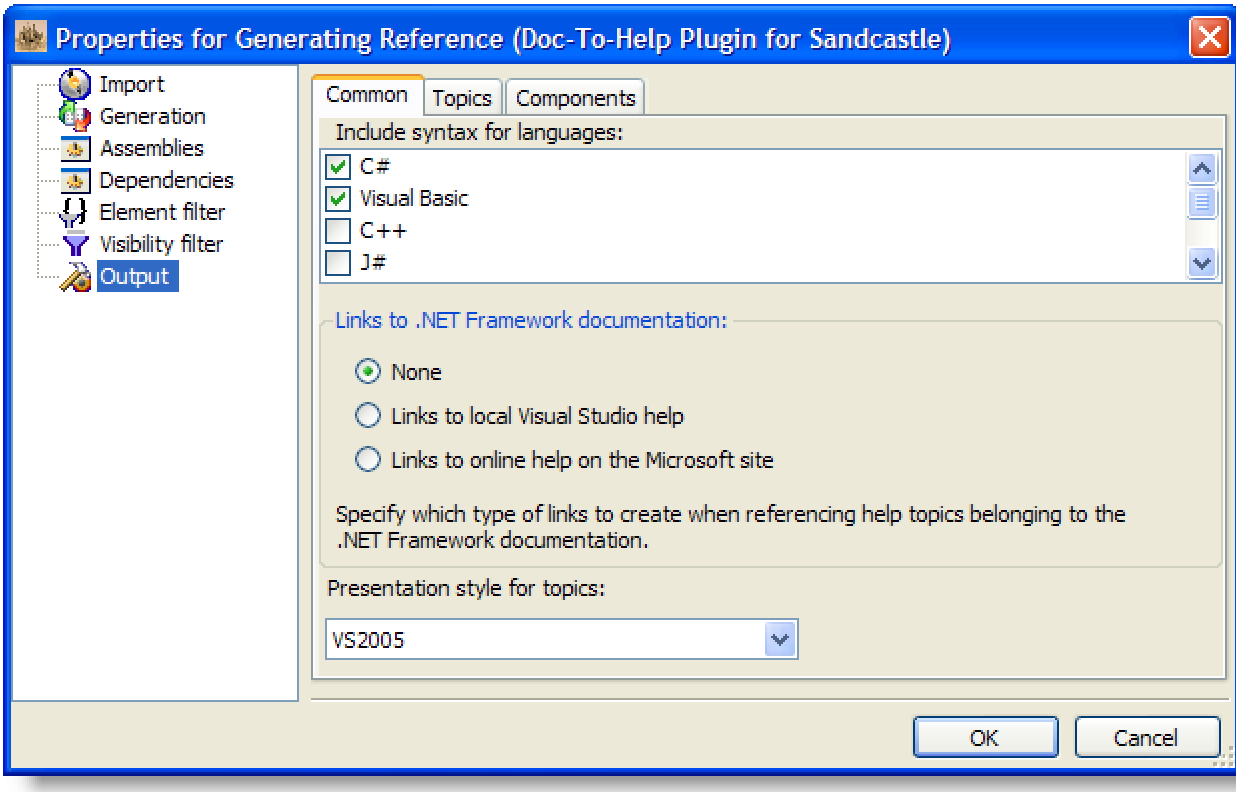
Output node

The **Output** node of the **Properties for Generating Reference** dialog box has three tabs: **Common**, **Topics**, and **Components**.



Common tab

Under the **Common** tab, you can choose syntax for languages, specify documentation links, and select a presentation style.



Include syntax for languages: To add syntax for a programming language to your reference section, simply check the box next to that reference section. You can include syntax for C#, Visual Basic, C++, J#, JavaScript, XAML, and ASP.NET.

Links to .NET Framework documentation: Select the check box for the type of links Sandcastle should create when linking to the .NET Framework documentation.

- None
- Links to local Visual Studio help
- Links to online help on the Microsoft site

Presentation style for topics: Select the check box next to one of the built-in themes provided by Sandcastle.

- **Prototype:**

Reference Library

CheckCRC32 Method

Namespaces ► C1.C1Zip ► C1ZipEntry ► **CheckCRC32()**

C#

Calculates a checksum value for the entry and compares it to the checksum that was stored when the entry was created.

Declaration Syntax

C#	Visual Basic

```
public bool CheckCRC32()
```

Return Value

True if the checksum values match, false otherwise.

Remarks

This method is used to check the integrity of the entries in the zip file. If the calculated checksum does not match the stored checksum, then either the zip file is corrupted or the program used to create the zip file is incompatible with **C1Zip**.

Assembly: C1.C1Zip.2 (Module: C1.C1Zip.2)

- **VS2005:**

Collapse All Code: All

.NET Framework Class Library

C1ZipEntry.CheckCRC32 Method

C1ZipEntry Class See Also

Calculates a checksum value for the entry and compares it to the checksum that was stored when the entry was created.

Namespace: C1.C1Zip
Assembly: C1.C1Zip.2 (in C1.C1Zip.2.dll)

Syntax

C#

```
public bool CheckCRC32()
```

Visual Basic (Declaration)

```
Public Function CheckCRC32 As Boolean
```

Return Value

True if the checksum values match, false otherwise.

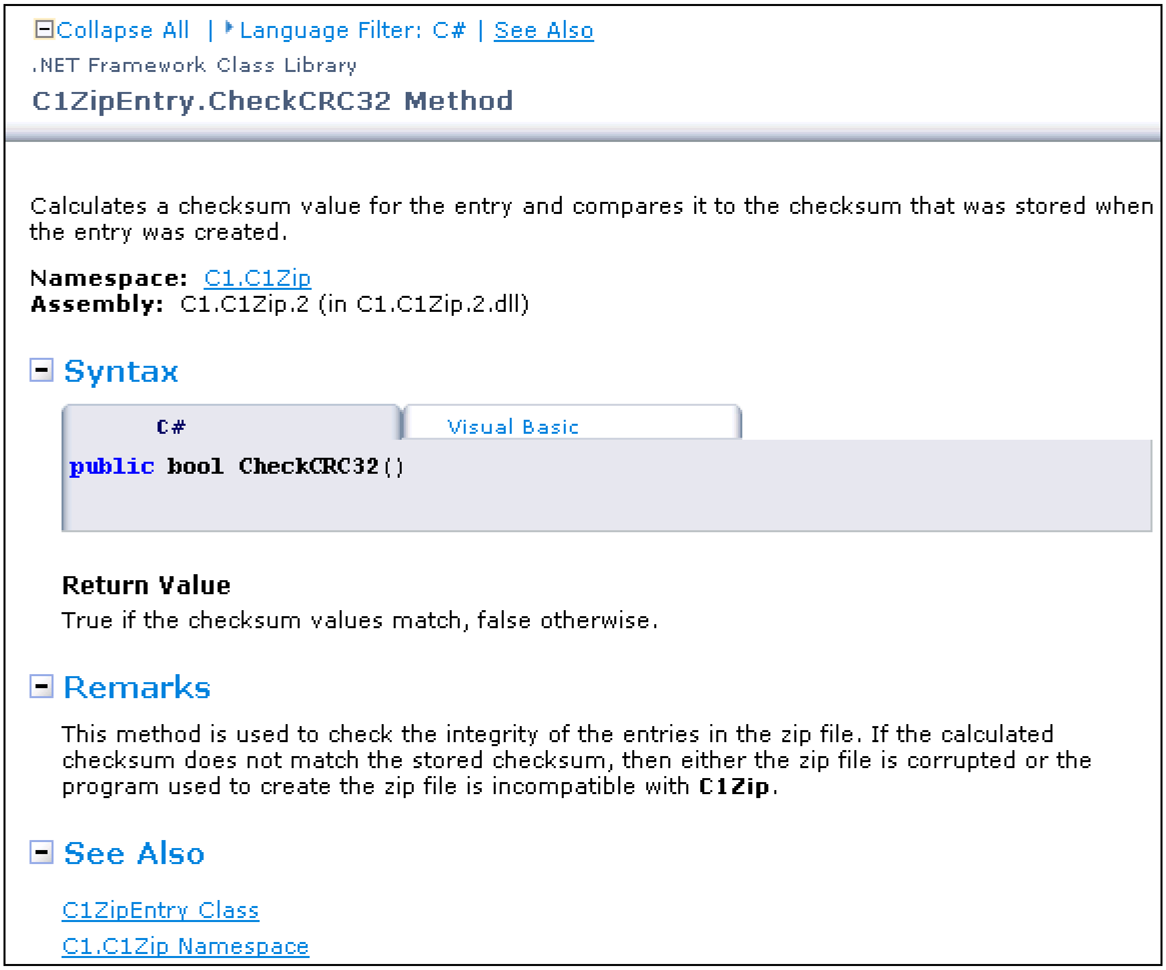
Remarks

This method is used to check the integrity of the entries in the zip file. If the calculated checksum does not match the stored checksum, then either the zip file is corrupted or the program used to create the zip file is incompatible with **C1Zip**.

See Also

C1ZipEntry Class
C1.C1Zip Namespace

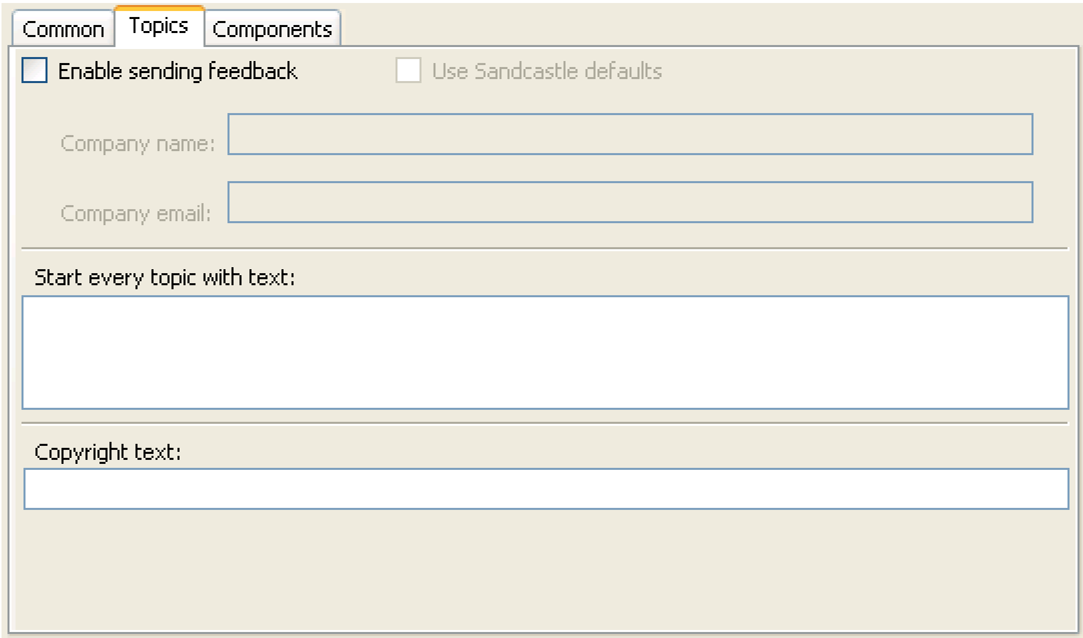
- **Hana**:



Collapse All | Language Filter: C# | See Also
.NET Framework Class Library
C1ZipEntry.CheckCRC32 Method
Calculates a checksum value for the entry and compares it to the checksum that was stored when the entry was created.
Namespace: C1.C1Zip
Assembly: C1.C1Zip.2 (in C1.C1Zip.2.dll)
Syntax
C#
Visual Basic
public bool CheckCRC32()
Return Value
True if the checksum values match, false otherwise.
Remarks
This method is used to check the integrity of the entries in the zip file. If the calculated checksum does not match the stored checksum, then either the zip file is corrupted or the program used to create the zip file is incompatible with C1Zip.
See Also
C1ZipEntry Class
C1.C1Zip Namespace

Topics tab

Under the **Topics** tab, you can enter feedback text, introductory sentences, and copyright information. Any text entered in this section will appear on every page of the reference section.



Enable sending feedback: Enable sending feedback allows you to add a footer with a feedback link to each page of the Sandcastle reference section. Selecting this check box inserts the generic statement "Send feedback on this topic to" into the footer of each page and enables the **Use Sandcastle defaults** check box and the **Company name** and **Company email** text boxes.

The following elements will affect the content of the feedback footer:

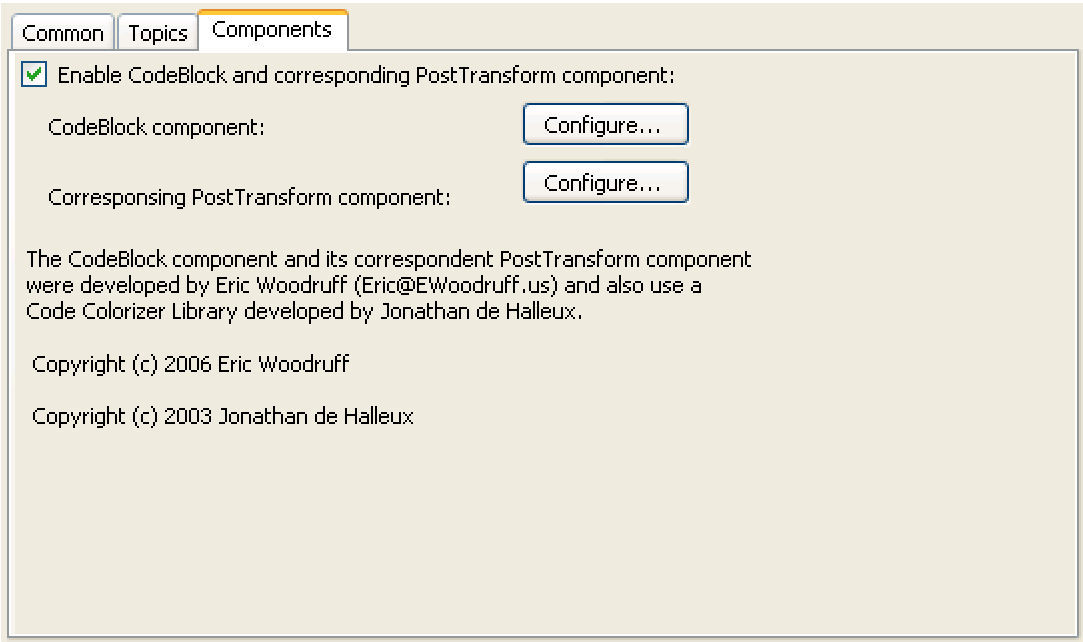
- **Use Sandcastle defaults**: Select this check box to insert the default Sandcastle contact information into the footer of the reference section. If you choose this option, the Company name and Company e-mail boxes will be disabled.
- **Company name:** Enter your company's name into this text box to add it to the feedback footer statement. If you typed "ComponentOne" into this field, the feedback footer statement would read "Send feedback on this topic to ComponentOne."
- **Company email:** Enter your company's contact e-mail address into this text box to add it, as a mailto link, to the feedback footer statement.

Start every topic with text: Text entered into this box will appear as the first sentence or paragraph of all reference topics.

Copyright text: Enter copyright information into this box to have it appear in the footer of your reference pages.

Components tab

The **Components** tab allows you to enable and configure the **CodeBlock** and **PostTranform** components.



Enable CodeBlock and corresponding PostTransform component: Select this check box to enable the **CodeBlock** and **PostTransform** components. These components, which were developed by Eric Woodruff, complement Sandcastle's functionality for better display of code blocks in various languages. For more information on the background and functionality of these components, see CodeBlock and PostTransform components on page 280.

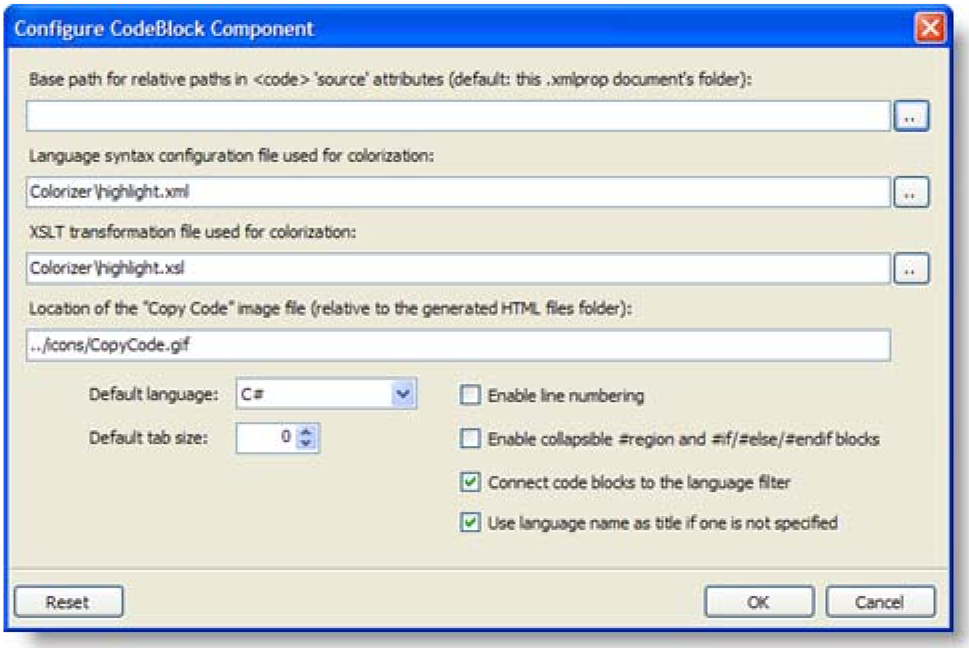
Click the **Configure** button next to its corresponding component to configure that component's settings. The **CodeBlock** component's **Configure** button opens the Configure CodeBlock Component (on page 277) dialog box, while the **PostTransform** component's **Configure** button opens the Configure PostTransform Component (on page 278) dialog box.

Configure CodeBlock Component dialog box

The **Configure CodeBlock Component** dialog box is used to configure the elements of the **CodeBlock** component.

To access the **Configure CodeBlock Component** dialog box, open the **Properties for Generating Reference** dialog box, select the **Components** tab of the **Output** node and click the **Configure** button next to "CodeBlock component." See Configuring the Sandcastle Reference Section with the Properties for Generating Reference dialog box (on page 267) for more information.

Also see CodeBlock and PostTransform components on page 280.



Base path for relative paths in <code>'source' attributes: Sets the base path to use for relative paths in the source attributes of code blocks. If this field is left blank, the current folder at the time of build is used as the base path.

Language syntax configuration file used for colorization: Sets the path to the .xml file that holds the syntax definitions for the languages supported by the colorizer.

XSLT transformation file used for colorization: Sets the path to the .xslt file that defines the XSLT transformation used to convert the parsed code block into colorized HTML.

Location of the "Copy Code" image file: Sets the path to the image file used for the "Copy Code" link.

Default language drop-down: Choose a language from the drop-down list to override the default language specified in the component's configuration.

Default tab size: Use this element to override the default setting of a language's tab size. If this is set to zero, the default syntax file setting will be used.

Enable line numbering: Check this box to add line numbers to your code examples.

Enable collapsible #region and #if/#else/#endifblocks: When checked, this feature makes #if, #else, and #endifblocks expandable and collapsible in your example code blocks.

Connect code blocks to the language filter: When this option is enabled, the code blocks in your reference section can be shown or hidden based on the user's language filter setting.

Reset: Click the **Reset** button to return to the **CodeBlock** component's default settings.

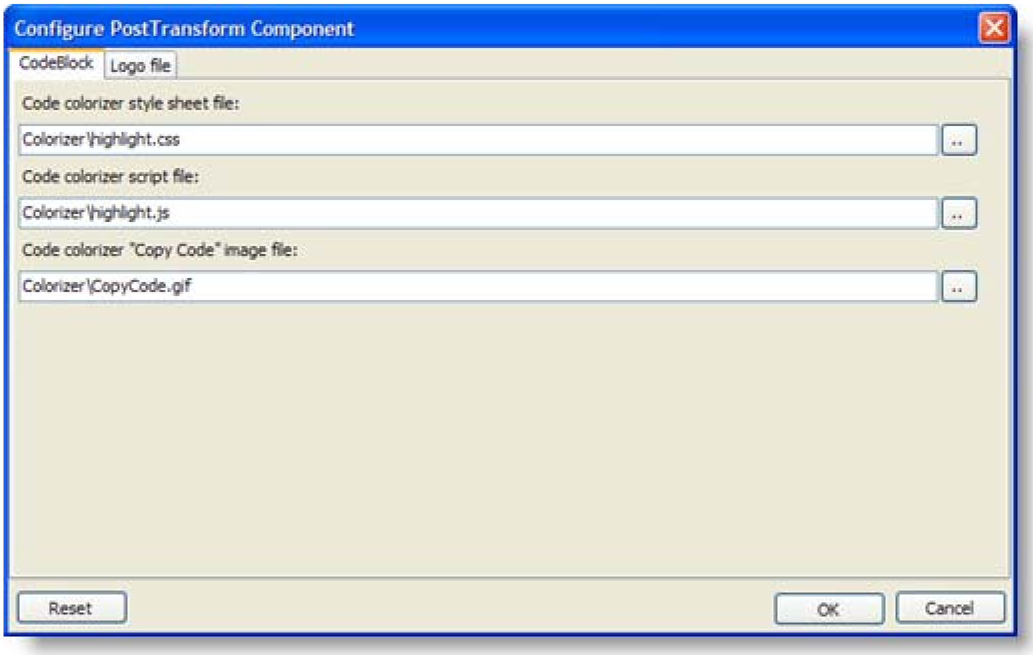
Configure PostTransform Component dialog box

The **Configure PostTransform Component** dialog box is used to configure the elements of the **PostTransform** component. The **Configure PostTransform Component** dialog box consists of two tabs: **CodeBlock** and **Logo file**.

To access the **Configure CodeBlock Component** dialog box, open the **Properties for Generating Reference** dialog box, select the **Components** tab of the **Output** node and click the **Configure** button next to "Corresponding PostTransform component." See Configuring the Sandcastle Reference Section with the Properties for Generating Reference dialog box (on page 267) for more information.

Also see CodeBlock and PostTransform components on page 280.

CodeBlock tab

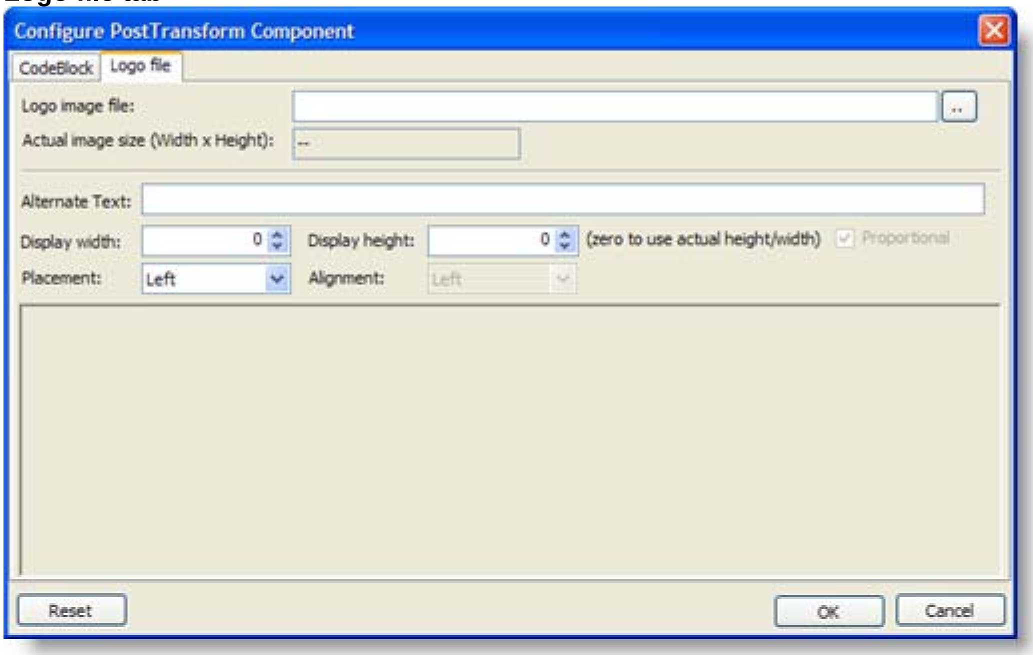


Code colorizer style sheet file: This text box sets the path to the stylesheet that will be used to define the colorized code.

Code colorizer script file: This text box sets the path to the script file containing the JavaScript code used to show and hide the collapsible sections of code examples.

Code colorizer "Copy Mode" image file: This text box defines the graphic that will be used for the "Copy Code" link that appears next to your code examples. The path of this image is defined in the **CodeBlock** component configurations under "Location of 'Copy Code' image file."

Logo file tab



Logo image file: Sets the path to the logo image file.

Actual image size: This field shows the original width and height of the logo file that you specified in the Log image file field. This can not be edited.

Alternate text: The text that, in the event that the image link is broken, will appear in place of the image.

Display Width: The width, in pixels, of the image as it will appear in the project.

Display Height: The height, in pixels, of the logo image as it will appear in the project.

Proportional: When selected, this will maintain the scale of your image upon resize.

Placement: This attribute allows you to specify whether the logo image will appear to the right, left, or above the topic title.

Alignment: This attribute allows you to specify whether the alignment of the log image is right, left, or center. Please note that this property is disabled unless the **Placement** property is set to **Above**.

CodeBlock and PostTransform components

The Doc-To-Help Sandcastle plugin includes Eric Woodruff's components (with his permission) that complement Sandcastle functionality for better display of code blocks in various languages, including syntax highlighting and connecting code blocks to the language filter. Normally, you don't need to change the settings for those components, but the XML source document's **Properties for Generating Reference** allows you to do so (double-click on the plugin document to open the Properties for Generating Reference on page 267 dialog box).

For a complete description of those components, see Eric Woodruff's article "Creating Custom Build Components for Sandcastle" at http://www.codeproject.com/csharp/SandcastleComponents.asp.

Here is an excerpt from that article listing the features of the **CodeBlock** component:

- Excess leading whitespace is stripped from the <code> blocks to left-align them correctly.
- Support for optional line numbering.
- Support for optional collapsible regions for code in #region and #if/#else/#endif blocks with support for nested collapsible regions. The VB.NET equivalents are supported also.
- A default title can be added based on the language that the code represents.
- A "Copy" link appears to the right on the title line that allows you to copy the code sample to the clipboard (Internet Explorer only). The code is copied as plain text without the highlighting and line numbers if used.
- Adds support for reading in an external source file or a defined region of an external source file so that you can keep code samples in a buildable project to test them for correctness and to do away with managing the code samples in the XML comments or include files.
- C#, VB.NET, C++, and J# code blocks can be associated with the language filter to show or hide them based on the language filter setting.
- Syntax highlighting of code blocks in <code> tags. Languages supported include C#, VB.NET, C++, J#, C, JavaScript, VBScript, and XML. An external configuration file is used so that it is possible to extend the colorizer to support other languages. The stylesheet is also replaceable.

XSLT transformation of the source code comments XML

The XSLT file is specified in the **Generation** node of the **Properties for Generating Reference** dialog box (double-click on the plugin document to open the **Properties for Generating Reference** dialog box).

Excluding elements without description

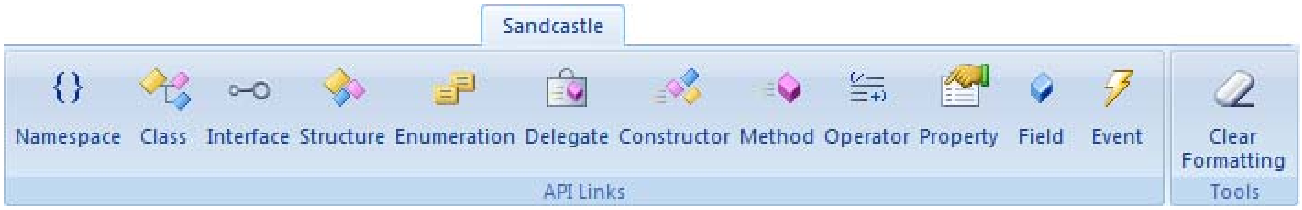
To exclude elements that do not have descriptions (to add them to the filter), go to the Element filter node of the **Properties for Generating Reference** dialog box (double-click on the plugin document to open the **Properties for Generating Reference** dialog box).

Click the **Modify** button (the **All Elements** dialog box will open) and click **Exclude elements without description**.

Using styles to create links from narrative text to reference text

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, and so on) with an appropriate style. 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.

The template **C1H_sandcastle_src.dot** is recommended for Source narrative documents in Sandcastle plugin projects, because it contains the **Doc-To-Help - Sandcastle** toolbar or ribbon, which has a button for each special **Link …** style for creating links from narrative text to reference text (styles such as Link Class, Link Property, and so on).



You can use your own template if you add the "Link ..." styles to it (or if you don't need those styles), but this template has these styles already defined for you with a toolbar that makes applying these styles a matter of a single click.

The following styles are supported:

- **Link Class**
- **Link Constructor**
- **Link Delegate**
- **Link Enumeration**
- **Link Event**
- **Link Field**
- **Link Interface**
- **Link Method**
- **Link Namespace**
- **Link Operator**
- **Link Property**
- **Link Structure**

The following styles are not supported:

- **Link Tag**, **Link Topic** - superseded by the D2HML style (on page 210) **C1H Link** (You may find these styles in legacy Documenter for .NET projects; D2HML makes these two styles unnecessary)
- **Link Type** - obsolete.

Link Class Style

The **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 click the **Link Class** button on the **Doc-To-Help Sandcastle 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 on page 289, 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 on page 289.

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

Link Constructor Style

The **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 click the **Link Constructor** button on the **Doc-To-Help Sandcastle** toolbar.

By default, Doc-To-Help applies the nested type separator substitution to the link text, if applicable, see Special Characters in Link Text on page 289. 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 on page 289. You can also use the argument list for constructor overloads, see Argument List in Links on page 290.

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

Link Delegate Style

The **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 click the **Link Delegate** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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

Link Enumeration Style

The **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 click the **Link Enumeration** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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
(System.Web.)TraceMode	System.Web.TraceMode	TraceMode Enumeration

Link Event Style

The **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 click the **Link Event** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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

Link Field Style

The **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 click the **Link Field** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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

Link Interface Style

The **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 click the **Link Interface** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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

Link Method Style

The **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 click the **Link Method** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289. You can also use the argument list for method overloads, see Argument List in Links on page 290.

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

Link Namespace Style

The **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 click the **Link Namespace** button on the **Doc-To-Help Sandcastle** 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

Link Operator Style

The **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 click the **Link Operator** button on the **Doc-To-Help Sandcastle** toolbar.

By default, Doc-To-Help applies the nested type separator substitution to the link text, if applicable; see Special Characters in Link Text on page 289. 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 on page 289. You can also use the argument list for operator overloads, see Argument List in Links on page 290.

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
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

Link Property Style

The **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 click the **Link Property** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289. You can also use the argument list for property overloads with arguments, see Argument List in Links on page 290.

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

Link Structure Style

The **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 click the **Link Structure** button on the **Doc-To-Help Sandcastle** 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 on page 289, 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 on page 289.

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

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 Sandcastle). 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**, 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 (on page 289), 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 **Target** template used by Doc-To-Help. These templates are installed by Doc-To-Help in the following locations:

- **In Word 2007** at C:\Users\(user name)\AppData\Roaming\Microsoft\Templates
- **In Word 2003** at C:\Documents and Settings\(user name)\Application Data\Microsoft\Templates.

If you uninstall Doc-To-Help and reinstall a newer version, the templates in this folder (including customized ones) will be saved to the \\Program Files\ComponentOne\DocToHelp\Backup\Templates folder. The Doc-To-Help installation program will inform you during installation. The Application Data or App Data folder is a hidden folder. To show hidden folders, open the **Folder Options** dialog box in your operating system (**Control Panel** > **Folder Options**). Click the **View** tab, under **Advanced Settings > Files and folders > Hidden files and folders**, select the **Show hidden files and folders** radio button. Click **OK**.

To see what template is used in building a help target click the **Home tab > Select Target button** to choose the Help target. Click the **Target Template** drop-down list.

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 target template. Here is the list of standard Sandcastle 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 the default in Sandcastle. 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, verify that the **Affects appearance** check box in the **Project Styles** dialog box is cleared. (See Defining Paragraph/Character Styles and Topic Types on page 181.)

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 by the Internet Explorer settings **Internet Options > General > Colors**. For WinHelp, the standard link color is green with an underline.

Using D2HML hot spots in XML comments to create links from reference text to narrative text

It is possible to use D2HML in your XML source code comments, for example, to create links from reference topics (generated by Sandcastle) to narrative topics (written manually by the author).

Writing comments in your source code that are included in the documentation through the XML comment file, you can use special tag <span> supported by the Doc-to-Help Sandcastle plugin, to include any links and other help hot spots supported by D2HML.

To format a range of text in your XML comment with any D2HML style, use the <span> 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 Using D2HML (on page 210) for details. Note that although the 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 <span> tag you can easily overcome this limitation; in fact, the <span> 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>
```

Glossary

ALinks

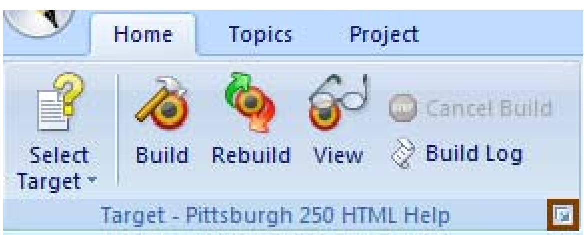
Associative topics or related topics. See Creating an Index or Groups on page 231.

Conditional Text

Marking text, documents, and/or topics as Conditional means that those items will display only in specific instances. This gives you a great deal of control for single sourcing. To add even more flexibility, Doc-To-Help has three types of conditions: platform, target, and attribute. See Utilizing Conditions (on page 174) for more information.

Dialog Box Launchers

The ribbons in Doc-To-Help feature dialog box launchers; small arrows in the bottom right corners of some ribbon groups. These launchers open dialog boxes.



KLinks

Keyword or Index links. See Creating an Index or Groups on page 231.

Live Links

When converting a Manual target (.doc) to PDF, Doc-To-Help can make all the hyperlinks live in the PDF (Live Links). To do so, in the Help Targets dialog box (on page 155), choose the **Manual Target** and select the **Live Links** and **Create Master Document** check boxes. Please note that pop-ups will not become live links in manual or PDF formats because those formats do not have a pop-up feature.

Natural Search

ComponentOne Natural Search provides a natural language search capability for WinHelp (.hlp), compiled HTML Help (.chm) and NETHelp Help (.htm) files. It is enabled and configured in the Help Targets dialog box (on page 155). Also see Natural Search Requirements (on page 37) for a list of project deliverables.

Related Topics

Topics related by hierarchy, keyword, or group. See Managing Related Topics on page 229.

Sandcastle

Microsoft's Sandcastle utility automatically creates MSDN formatted reference documentation from .NET assemblies and XML comment files. See Documenting Your Class Library with Microsoft Sandcastle on page 265.

Subtopic Links

Subtopic links are automatic links that are displayed at the bottom of Help topics. By default, the related subtopics are the children of the main topic (Heading 1s will display Heading 2 as subtopics; Heading 2s will display Heading 3s.) See Managing Related Topics (on page 229) for more information.

You may add custom related topics using the Related Topics pane (on page 111).

By default, the **Related Topics label** (the title displayed above the related topics) is **More:**. You may change it for an entire Help Target using the Help Targets dialog box (on page 155), or for individual topics using Topic Properties dialog box (on page 225).

Target

Doc-To-Help has six default Help Targets: Help 2.0, HTML Help, JavaHelp, Manual, NetHelp, and WinHelp. They are all specific output types.

You may create multiple Help Targets of the same type, or rename the default Target names. You can also remove Targets from your project to streamline it. See Creating Help Targets (on page 155) for more information. See Doc-To-Help Outputs and Deliverables (on page 30) for a complete guide to targets and deliverables.

Variable

Variables allow you to manage content in one place for reuse across your project. Variables are commonly used for text that can change frequently, such as product or company names — or larger chunks of text, such as tables.

Variables are developed in two steps; first, you create the variable — second, insert it into your project. When you build your target, the variable value is automatically inserted in the target.

With Doc-To-Help, you can create two different types of Variables — text or rich content. Text variables take on the same style and formatting (font, etc.) as the location they are inserted. Rich content variables retain the formatting they were given when created.

See Creating Variables (on page 194) for more information on creating and using variables.

Index

P

Q

R

S

T

U

V

W

X