Doc-To-Help 2014 User Guide

By MadCap Software, Inc.

Corporate Headquarters
MadCap Software, Inc.
7777 Fay Avenue
La Jolla, California 92037

Email:
Info@MadCapSoftware.com

Web site:
http://www.madcapsoftware.com

Sales Email:
Sales@MadCapSoftware.com

Telephone:
1-888-MadCap1 or
+1 858-320-0387

Technical Support Web site:
http://www.madcapsoftware.com/support

This manual was produced using *Doc-To-Help.*™

Contents

Welcome to Doc-To-Help 2014!

Doc-To-Help 2014 is a single-source authoring tool that makes it possible to write once and publish to many different deliverables. Technical communicators, Help authors, policy writers, and other content creators can author any type of information in Microsoft Word, or in the built-in HTML5 content editor and publish to the Web, Help systems, EPUB, or printed manuals. It is the perfect choice for creating many types of user assistance, converting existing Word documents to online materials, and producing professional quality printed manuals.

With Doc-To-Help 2014 you can:

- **Choose your work environment**

  Doc-To-Help features a built-in HTML5 editor, or you can work in Microsoft Word and use the integrated toolbars and dialogs.

- **Create many types of user assistance**

  With Doc-To-Help, you can build compiled HTML Help, NetHelp (a browser-based, platform-independent Help for Web deployment), EPUB, Microsoft Help Viewer, Microsoft Help 2.0, JavaHelp, Eclipse Help, and WinHelp.

- **Create responsive help**

  When you choose NetHelp, you can also choose a Responsive theme. This theme was especially designed to work well on both desktop and mobile platforms; it will automatically adapt to the current window size. It also includes a Favorites tab so that end users can easily mark favorite topics.

- **Include widgets in your help**

  The HTML5 content editor includes seven widgets you can use to improve your Targets and your efficiency. They are: LightBox, Gallery, Carousel, Topic Contents, Code Highlighter, Tabs, and Note.

- **Collaborate on projects**

  The SharePoint® collaboration feature makes it possible to share your project documents across your team or company, import documents from SharePoint, and create a translation workflow. You can also publish Help to SharePoint Libraries and SharePoint Wikis.

Quick Start Resources

See Doc-To-Help's Learning Resources for a complete list of videos, guides, and tutorials.

- Doc-To-Help Quick Tour on page 41
- How-To Video Library
- Free Introductory Webcasts
- Training Information
- Learn How Doc-To-Help Works in 5 Lunch Breaks tutorial series (with video)
- Introduction to Single Sourcing on page 3
- Doc-To-Help Outputs and Deliverables on page 13
- Doc-To-Help Overview video
- Tour of the Doc-To-Help Interface video
- Doc-To-Help Workflow video
- Learn Doc-To-Help with the Sample Projects

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 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**
  - **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.
  - **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.
  - **EPUB** (Target designed specifically for EPUB readers.)
  - **Eclipse Help** (Deliverable: Help for the Eclipse environment.)
  - **JavaHelp** (Deliverable: Help for Java applications.)
  - **Microsoft Help Viewer** (Only used to create Help for .NET components; projects are integrated with Microsoft® Visual Studio 2010 and above)
  - **Microsoft Help 2.0** (Only used to create Help for .NET components; projects are integrated with Microsoft® Visual Studio 2002-2008)
  - **WinHelp** (Deliverables: .cnt and .hlp files) (Older Help format being phased out. For more, read this article.)
- **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 2007/2010/2013]), 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 133.

How Doc-To-Help Makes Single Sourcing Easier

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

1. Automatically structures content and navigation of online Help.
   1. Heading 1’s automatically become parent topics, and all of the Heading 2’s under it become subtopics.
   2. Parent topics automatically include “See Also” links to subtopics. See Managing Related Topics on page 321 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 160 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.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**

    3. Standardize on a consistent heading (such as "Using the… Screen") which makes sense in both the book and help.
    4. 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 160 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.

    5. 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 124).

    6. 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 7. 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.
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 125, add glossary items, add links, etc.
7. If you would like to change the way your online Help "skin" looks, use the Theme Designer 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.
- If producing Responsive NetHelp, keep an eye on desktop-specific terminology ("Click" vs. "tap" for example). Substituting "choose" is one solution. You can also create text variables with conditions so that terminology switches automatically, based on the Target you are building.

Additional Tip: Create a variable 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, the Content Editor, or an HTML 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 .dotx files). See Word File Templates on page 8.
- For Doc-To-Help projects containing HTML files, the look of their Source/Target is controlled by separate cascading style sheets (.css files). (This is for online outputs; manuals are transformed using Word templates.) See HTML File Style Sheets on page 11.

Note: A Doc-To-Help project can include Word files, HTML5 files, HTML 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 303 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 168 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 124 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, Microsoft Help 2.0, and JavaHelp targets specifically) is configured in the Theme Designer. (The skin includes the items in the Help window that surround the content — icons, breadcrumbs, tabs, buttons, colors, etc.)

To apply a source template

In existing Word documents:
The document must be open in Word to apply a template.

- Word 2007/2010/2013: **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 109, 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 182 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 Content Editor: **Home tab > Source ribbon group**_D2HLink_617419, click the **Source CSS** button.

Setting the default for new HTML5 Documents:
You may set the default Source Style Sheet for all new HTML5 documents created in your project. In the **Home tab >** Source ribbon group on page 109, 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 182 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 8 and HTML File Style Sheets on page 11 for details.)

- To change the **Target** Template: In the **Home** tab, Target Design ribbon group on page 109, 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 109, 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 10.

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 176 and HTML File Style Sheets on page 11.

Applying Styles
To learn more about applying styles, see:

- Editing Word Documents on page 285
- Editing HTML5 Documents
- Editing HTML Documents on page 293

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/EPUB/Microsoft Help Viewer/Microsoft Help 2.0/Eclipse Help/JavaHelp 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 285 (**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_NOMARGIN_A4.DOT: This is the source template for A4 sized paper. In the New Project Wizard called "Normal A4 Template (no left indent)"	**C1H_PRNOMARGIN_A4.DOT**: This is the target template used to format the printed manual target (A4 sized paper).	**C1H_HTML.DOT**: This is the target template used to format NetHelp, HTML Help, EPUB, Microsoft Help Viewer, Microsoft Help 2.0, JavaHelp, and Eclipse Help Targets.
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 has a two inch wide left margin.	
C1H_NORM_A4.DOT: This is the source template for A4 sized paper with a two-inch left margin. 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).	
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	**C1H_PRSIDE_A4.DOT**: This is the target template used to format the	

A4 sized paper. In the New Project Wizard called “Sidehead A4 Template”	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 10 for more information.

Please Note: Templates are stored in Windows® 7 and Vista at: \\Users\ (user name)\AppData\Roaming\Microsoft\Templates

If you uninstall Doc-To-Help and reinstall a newer version, the templates will remain in that folder. However, if one of your customized templates uses the same name as one of Doc-To-Help’s default templates, and there is a newer version of that template in the installation, Doc-To-Help will save your version of the template to the \\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp\Backup\Templates folder.

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**.

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 8 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.

Watch the video: Editing Templates (6:36)

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 your machine.

To edit a Template

1. After the template is open, the process will vary depending upon your version of Microsoft Word.
   - In Word 2007/2010/2013: Click the **Home ribbon** > **Styles ribbon group** dialog box launcher. The **Styles** 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 2007/2010/2013).
3. Save the template (.dot) 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 168.

Please Note: Templates are stored in Windows® 7 and Vista at: \\Users\ (user name)\AppData\Roaming\Microsoft\Templates

If you uninstall Doc-To-Help and reinstall a newer version, the templates will remain in that folder. However, if one of your customized templates uses the same name as one of Doc-To-Help's default templates, and there is a newer version of that template in the installation, Doc-To-Help will save your version of the template to the \\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp\Backup\Templates folder.

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 File Style Sheets

When working with HTML5 or HTML Source Files

If you want to …

- Change the look of your **Source HTML5 or HTML files** — Edit the **Source Style Sheet** (Column 1)
- Change the look of your **Printed Manual Target** — Edit the **Target Word template** (see Word File Templates on page 8)
- Change the look of your **NetHelp/EPUB/HTML Help Target** — Edit the **Online Help Target Style Sheet** (Column 3 — C1H_HTML_full or short.css; or NetHelp.css)
- Change the look of your **WinHelp Target** — Edit the **Online Help Target Style Sheet** (Column 3 — C1H_HELP_full or short.css)

Predefined Source Style Sheets	Predefined Target Style Sheets and templates	
Pick One	Printed Manuals	Online Help Pick One for HTML5 or HTML/One for WinHelp
C1H_Source_full.css: Source style	Choose a **Target Word template** (see	**C1H_HTML_full.css**: Target style

sheet with the full set of styles. In the New Project Wizard called “Full Set of Styles”	Word File Templates on page 8) In most cases, you would choose **C1H_PRNOMARGIN.DOT**	sheet for all HTML-based targets with the full set of styles.
C1H_Source_short.css: Source style sheet with the minimum set of styles. In the New Project Wizard called “Minimal Set of Styles”		**C1H_HTML_short.css**: Target style sheet for all HTML-based targets with the minimum set of styles.
		NetHelp.css: Target style sheet for NetHelp Targets.
		C1H_Help_full.css: Target style sheet for WinHelp target with the full set of styles.
		C1H_Help_short.css: Target style sheet for WinHelp target with the minimum set of styles.

Watch the video: Editing Style Sheets (3:44)

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 176 for details.

- **In new projects:** When you create a new project in Doc-To-Help 2014, 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 178 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 178 if you wish.
- Copies of the original Style Sheets are stored in C:\Program Files [or Program Files (x86)]\ \MadCap Software\DocToHelp\DefaultCSSFiles.

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.

Watch the video: All About Doc-To-Help Outputs (3:28)

Deliverable	Details	Pros	Issues	Deliverable File(s)	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.
NetHelp (Browser-based Help) This release of Doc-To-Help features a new Responsive Theme.	Uncompiled HTML; deliverable is multiple html files. NetHelp can be Section 508 compliant, see Creating Section 508 Compliant Help on page 19.	Can be delivered on the Web, installed locally, or both. See Installing NetHelp on page 17 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. Security issues can cause NetHelp installed locally to display an ActiveX warning, which may be solved by using the "Mark of the Web." See NetHelp Local Installation on page 19.	Multiple .htm/.html files. Default home page is **index.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 or posted on your server. Server side search may be enabled for NetHelp, see Search Options for NetHelp on page 21.
Notes on NetHelp	Local 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. If you open or install NetHelp locally, it will not display in Chrome or Opera because of limitations with those browsers, but it will display in other browsers. NetHelp deployed on a web server will open in Chrome and Opera.				
HTML Help	Compiled HTML; deliverable is .chm file.	One file; tri-pane interface is easy to use.	Format is 10+ years old. Locally installed HTML Help	.chm	By default, the **HTML Help** folder of your project. _D2HLink_617450

Deliverable	Details	Pros	Issues	Deliverable File(s)	File location
			files (those installed on the end user's client machine, not their server), work without issues. If you plan to install HTML Help on a server, there are security issues, see http://support.microsoft.com/kb/902225		
Eclipse Help	Help Target for the Eclipse development environment. Eclipse was released by IBM in the early 2000s.	An Eclipse Help system is a plug-in for Eclipse. This plug-in supports a table of contents, index, search, and context-sensitive help.	Eclipse Help is compatible with all browsers, but the Eclipse SDK must be installed on the same server as the Help system. Eclipse Help does not support opening topic links in secondary windows.	Multiple files; see http://help.eclipse.org/indigo/index.jsp (search on "User Assistance Support").	By default, the **Eclipse** folder of your project. To view Eclipse Help, you must have the Java Runtime Environment (JRE) and Eclipse installed (either the 32 or 64-bit versions). To build it, you must have 32-bit Eclipse installed. Eclipse is available for download from http://eclipse.org/downloads/. After installation, the location of the eclipse.exe file should be set in the Doc-To-Help Options on page 34 dialog box.
EPUB EPUB stands for "electronic publication" and is standard of the International Digital Publishing Forum (IDPF).	EPUBs are readable on a variety of devices, including Nooks. The content display adjusts for the device, as well as the device settings.	You can create interactive, portable books that can be read on multiple devices. Doc-To-Help supports the latest specification, EPUB 3.0, as well as 2.0.1.	Dynamic content is not supported, but the content will be displayed. See **Notes on EPUB** (below) for complete information.	.epub	By default, the **EPUB** folder of your project. You must add your EPUB to your reader (such as Calibre) in order to view it. To set the default EPUB reader in Doc-To-Help, go to **File** tab > **Tools** > **Options**. In the **Options** dialog box, select the **Viewers** button and choose the EPUB viewer executable.
Notes on EPUB	Please note that many devices don't support Cascading Style Sheets or only support them partially, so your EPUB display can vary depending on the device. Doc-To-Help's EPUB target was tested with the following readers: • Calibre E-book management (http://calibre-ebook.com/) • Sony Reader for PC (http://ebookstore.sony.com/download/) • NOOK for PC (http://www.barnesandnoble.com/u/free-nook-apps/379002321/) • Adobe Digital Editions (http://www.adobe.com/products/digitaleditions/) Dynamic content is not supported in EPUBs, but the content will be displayed as follows: • Popups and Glossary Terms will be opened in the current window. • All content in Collapsible Sections (created with the **Collapsible Section** button) will be displayed (just as it would in Manual Targets).				

<table>
<tr><th>Deliverable</th><th>Details</th><th>Pros</th><th>Issues</th><th>Deliverable File(s)</th><th>File location</th></tr>
<tr><td></td><td colspan="5">Topic links that were specified to open in secondary windows will open in the current window instead.
Keywords with multiple entries and Groups don’t open in a popup when clicked, but instead link to the Index, where those Keywords and Groups are displayed, along with the list of topics included in each. (The Index can be renamed in the Theme Designer.)
Expanding Text and Dropdown Text (both created with the Inline Text button) may be displayed or hidden in EPUBs. The option you prefer can be set in the Help Targets dialog box using the Show expanding text and Show dropdown text check boxes. The hidden portion of inline text will always remain hidden.</td></tr>
<tr><td>JavaHelp</td><td>Java Help 1.1.3 is supported, as well as JavaHelp 2.0</td><td>JavaHelp software was developed to provide a standard Help solution for pure Java applications.
JavaHelp software was released in April 1999, and is currently in release 2.0.</td><td>While there is no “standard” viewer, JavaHelp uses components from the HotJava browser for its display.</td><td>Multiple files; deliver the entire JavaHelp folder (its default name) in your project directory.</td><td>By Default, the JavaHelp folder of your project.
To view a JavaHelp HelpSet, you must have the necessary files installed on your machine. See Notes on JavaHelp below.</td></tr>
<tr><td>Notes on JavaHelp</td><td colspan="5">To build and view JavaHelp, you must first install the necessary files from Oracle.

Do the following:
Install the Java Developer Kit (JDK).

The JDK6 for Windows x86 and Windows x64 are available at:
http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u26-download-400750.html
Download and unzip the JavaHelp files from http://download.java.net/javadesktop/javahelp/javahelp2_0_05.zip
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 [or Program Files (x86)]\Java\jdk1.6.0 (use the location of your JDK installation)
JAVAHELP_HOME
Value: C:\Program Files [or Program Files (x86)]\Java\jh2.0 (use the location of your jh2.0 directory)
JHHOME
Value: C:\Program Files [or Program Files (x86)]\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 [or Program Files (x86)]\Java\jdk1.6.0\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 the JDK installed on your machine.</td></tr>
<tr><td>WinHelp</td><td>Compiled RTF files; deliverable is .hlp and .cnt</td><td>None. If your product is a legacy one with no plans to</td><td>Original Help format. Look and navigation not as user-</td><td>.hlp and .cnt</td><td>By default, the Help folder of your project.</td></tr>
</table>

Deliverable	Details	Pros	Issues	Deliverable File(s)	File location
	file.	change Help format, you will be able to continue producing it.	friendly as newer formats. The WinHelp viewer is not included with Windows 7/8 and Vista, but those users can download the WinHelp viewer, see http://support.microsoft.com/kb/917607 WinHelp will not build on 64-bit machines.		
Microsoft Help Viewer	Help Target for those developing Help for Visual Studio 2010 and above 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 2010 and above.	HelpContentSetup.msha and <target base name>.mshc The MSHC is a zip file containing all of the Help content.	By default, the **MSHelpViewer** folder of your project. To specify the version of the Microsoft Help Viewer to use when viewing Microsoft Help Viewer Targets, in Doc-To-Help, go to **File** tab > **Tools** > **Options**. In the **Options** dialog box, select the **Viewers** button and choose the correct version of the viewer. (Options available depend on your Visual Studio installation.)
Notes on Microsoft Help Viewer	In order to create and view Microsoft Help Viewer files you must install Visual Studio 2010. Other properties are set in the Doc-To-Help **Help Targets** dialog box.				
Microsoft Help 2.0	Help Target for those developing Help for Visual Studio 2002 - 2008 only. Deliverable can be the reference documentation only, or Doc-To-Help can be used to add narrative text. Microsoft Sandcastle is used to	Resembles Microsoft MSDN collection and uses some of the same underlying technology	Format can only be included in Help for Visual Studio 2002 - 2008. 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	.HxS (topic files), .HxC (project file), .HxF (include file), .HxT (table of contents). .HxA (attributes), .HxK (index)Your deliverables may vary based on your	By default, the **MSHelp** folder of your project.

Deliverable	Details	Pros	Issues	Deliverable File(s)	File location
	generate reference documentation.		Help systems.	application. For more on Microsoft Help 2.0 redistributables, see http://msdn.microsoft.com/en-us/library/bb165722(VS.80).aspx	
Notes on Microsoft 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 for more information. To build Help 2.0, Doc-To-Help also needs to know the locations of the Help 2.0 executable files. By default, Doc-To-Help assumes the following locations: C:\Program Files\Microsoft Help 2.0 SDK\HxComp.exe C:\Program Files\Microsoft Help 2.0 SDK\HxReg.exe C:\Program Files\Common Files\Microsoft Shared\Help\dexplore.exe				

Watch the video: How to Deliver a Target (1:01)

Installing NetHelp

NetHelp can be deployed on a web server, or installed locally (on the end user's machine). See NetHelp Local Installation on page 19.

NetHelp 2.0 Targets have no special installation requirements for web deployment, simply copy the files to the server. The instructions below are for NetHelp Classic Targets.

Supported Browsers:

NetHelp 2.0 supported browsers:

- Internet Explorer 8 or higher
- Firefox 3 or higher
- Opera 9.6 or higher*
- Apple Safari 3.1 or higher
- Google Chrome (all versions)*

NetHelp 2.0 Responsive Theme supported browsers:

- Desktop: Internet Explorer 8 or higher
- Mobile platforms: Android 4.0 or higher, iOS

NetHelp Classic supported browsers:

- Internet Explorer 8 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*

*NetHelp installed locally will not display in Chrome or Opera because of limitations with those browsers, but it will display in other browsers. NetHelp deployed on a web server will open in Chrome and Opera.

NetHelp Classic Server Installation

NetHelp Classic 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 Classic. See **Installing NetHelp Classic on a Java servlet-enabled web server** below.

NetHelp Classic has two search options: **JavaScript Client** and **JavaSearch**, see NetHelp Classic Search Options on page 24 for more information.

Note: These instructions are for NetHelp Classic Targets; there are no special installation requirements for NetHelp 2.0, simply copy the files to the server.

IIS Requirements:

- Windows 2000 Server or later
- IIS 5.0 or later
- For NetHelp Classic projects using **Java** search: Sun Java Runtime Environment (JRE)
  If the project is using **JavaScript** search this is not necessary.

To install NetHelp Classic on Microsoft Internet Information Server

1. Create an IIS virtual directory for each NetHelp Classic 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. For NetHelp Classic projects, 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.
5. To view the Help, open the **index.htm** file.

   You can rename the NetHelp destination folder using the **Folder** field of the Help Targets on page 133 dialog box. The default file name can be renamed using the **Default file** field.

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 Classic on a Java servlet-enabled web server

Deploying NetHelp on any Java-enabled Web server (such as Tomcat, Resin, etc.) requires publishing web pages to 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/index.htm** on your server to view NetHelp.

NetHelp Local Installation

NetHelp (2.0 or Classic) installed locally can be given the Mark of the Web (MOTW). This means that anyone viewing your NetHelp locally will not receive a browser security warning first (this ActiveX security warning is dependent on the security settings of the machine). The Mark of the Web adds MOTW commentary text to every HTML file in the NetHelp Target. It is turned on using the **Mark of the Web** check box in the Help Targets on page 133 dialog box, see NetHelp Target for more information. There are some issues with the MOTW; for example, links to PDFs will not open from NetHelp files using the MOTW. To learn more see:

- http://jtfassociates.com/using-the-mark-of-the-web-motw/
- http://msdn.microsoft.com/en-us/library/ms537628(VS.85).aspx

Installing NetHelp on a client machine

1. Build the NetHelp target with Doc-To-Help.
2. Copy the contents of the **NetHelp** folder to the desired directory.
3. To view the Help, open the **index.htm** file.

   You can rename the NetHelp destination folder using the **Folder** field of the Help Targets on page 133 dialog box. The default file name can be renamed using the **Default file** field.

Please note: NetHelp installed locally will not open in Chrome or Opera because of limitations with those browsers, but it will display in all other browsers. NetHelp deployed on a web server will open in Chrome and Opera.

Creating Section 508 Compliant Help

Doc-To-Help **NetHelp** (2.0 and Classic) projects 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://access-board.gov/508.htm.

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 133).
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, see Customizing Themes with the Theme Designer on page 199.
- 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 133 (NetHelp Targets only) and therefore the icon is a book, this title indicates whether it is open or closed. These strings can be changed, see Customizing Themes with the Theme Designer on page 199.
- 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, see Customizing Themes with the Theme Designer on page 199.

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

- All links and buttons are accessible from the keyboard using the **Tab** key.
- When the **Dynamic table of contents** check box is selected and a 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, see Customizing Themes with the Theme Designer on page 199 (**Accessibility** properties).

Note: For more information about editing Themes, see Customizing Themes with the Theme Designer on page 199.

To create alternative text for images

In order for your project to be Section 508 compliant, all images in your source 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 Content 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. Click the **Alt Text** tab and enter the text in the **Alternative text** text box
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

Tables must have a caption 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** field when inserting the table in the Content 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. Select **Insert Caption** from the menu. 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. Click the **Review** tab, and then 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 in the Target if text is not provided.

Search Options for NetHelp

Doc-To-Help's NetHelp Target include powerful search features that you can customize for your needs, such as Server Side search.

- **NetHelp 2.0**

  By default, **JavaScript Client** search is used in NetHelp 2.0 Targets and requires no server setup. **JavaScript Server** search requires setup, but will improve the search speed of NetHelp 2.0 Targets installed on a web server. See Setting up Server Side Search for NetHelp 2.0 Targets on page 22.

- **NetHelp Classic**

  By default, **JavaScript Client** search is used in NetHelp Classic Targets, but you also have the option of **Java Server** search. See NetHelp Classic Search Options on page 24.

JavaScript search supports exact phrase, Boolean, and fuzzy searches.

- **Exact phrase** search means that if you enclose a phrase in double quotes, the search will be limited to that exact phrase in the Help file, for example: "sports teams".
- **Boolean** search means that you can use "AND" or "OR" (no quotes) between words or exact phrases when searching. You can also use "AND NOT" or "NOT" before a word or exact phrase to exclude topics containing that phrase from the results. By default, if there is no "AND" or "OR" between words, "AND" is assumed. For example: football or hockey, sports and not baseball.
- **Fuzzy** search will display alternative search options (and results) if the user enters a search term that is close to the term entered. For example: sorts. Search will return "No topics found. Did you mean: sports."

Setting up Server Side Search for NetHelp 2.0 Targets

General information:

- The Search server is JavaScript-based and runs inside the Node.js server.
- The Search server uses HTTP protocol to handle search queries, so it requires a free port number on the server. For example, if the NetHelp 2.0 Target on the server is handled by a web server (IIS, Apache, etc.) and by default uses port 80, the search server will use 8256 by default. You must verify that the 8256 port for the search server is free and is not blocked by a firewall.
- The Search server can be used for either a single Target or for multiple Targets. Using a separate search server for each target can improve search performance, while using the single search server for multiple targets makes the setup and support processes easier. If you run several search servers, each of them requires a separate port number.

Setting up the Search Type

1. Open the Help Targets on page 133 dialog box (**Home** tab > click the dialog box launcher on the **Target** ribbon group).
2. In the **Search Type** field, choose **JavaScript Server** search.
3. Build the Target.

After the target is built, click the **View** button or view it when Doc-To-Help asks if you would like to view the Target. This will start the search server automatically and will open the NetHelp 2.0 target in a browser where you can test the search server. (This starts the server locally and is only for testing.)

When you close the Target, the search server will be automatically shut down. When you build/rebuild this or another target and click on the **View** button again, the search server is restarted automatically.

Note: You can test only one Target at a time. If you have two Doc-To-Help instances running and build a targets with the **JavaScript Server** search type, only one of them can be tested with the **View** button at a time. If you have one Target open already, and try to open another, you will receive an error message that says “you must shutdown the search server.” To do so, close the open Target and try again. A similar problem can occur if you run the server manually (as described below) and haven't stopped it. In that case you will need to stop the search server manually (instructions below).

Setting up a server for a single target

Prerequisite: Node.js must be installed on the server. It can be downloaded here: http://nodejs.org

1. Copy your built target to the web server, for example, to "c:\d2h\targets\Pittsburgh250XMLSource".
2. Depending on the web server that you use (IIS, Apache, etc.), set up your server to serve the target files (for example, in IIS, set up a virtual directory). For example, http://localhost/d2h/Pittsburgh250XMLSource can point to "c:\d2h\targets\Pittsburgh250XMLSource".
3. Run the search server in node.js. For Windows:
   - Open the command line (Start > All Programs > Accessories > Command Prompt).
   - Verify that the current directory is "C:\". If it is not so, type "c:" and press **Enter**, this will change current directory to "C:\"
   - Change the current directory and set it to the "js\nodejs" subfolder in the built Target folder, e.g., cd "c:\d2h\targets\Pittsburgh250XMLSource\js\nodejs", and press **Enter**
   - Type "node index.js" and press **Enter**.

You should see the message "The server has started" in the command line window. You will also see log and error messages in this window when the search server is used.

To close the search server, you can press **Ctrl+C** in the command line window or just close it.

Viewing and testing server side search

1. Open the built Target in a browser (not from the local file system) i.e., the URL in the browser must start with the "http://" prefix, for example, http://localhost/d2h/Pittsburgh250XMLSource.
2. Try searching for a word.

Note: When you perform a search, you can see requests and responses from the server in the open command line window.

Setting up a single search server for multiple targets (for Windows)

1. Repeat steps 1-2 from Setting up a server for a single target on page 22 for all targets.
2. Copy the search server files to a separate folder, for example, "c:\d2h\nodejs".
3. Go to the folder containing the search server files, "c:\d2h\nodejs" and open the "settings.json" file. This is a configuration file for the server, in JSON format. By default it has two settings:
   - "port" is the port number that the search server will use. The default value is 8256. Make sure that this port isn't used by another application and isn't blocked by firewalls, otherwise you must change this value to any available port number.
   - "host" is the IP address on which the search server will wait for requests. The default value is "null" and will work in most cases. You need to change this setting only if the system has several network interfaces and you want the search server to use a specific network interface.
4. To handle multiple targets by a single search server, you need to add the "targets" option to this file with information about your targets. After modifying the "settings.json" file, it could look like this:

```
{
  "port": 8256,
  "host": null,
        "targets": [
                {
                        "path": "/d2h/Pittsburgh250XMLSource/",
                        "index": "c:\\d2h\\targets\\Pittsburgh250XMLSource\\searchindex.js"
                },
                {
                        "path": "/d2h/Pittsburgh250WordSource/",
                        "index": "c:\\d2h\\targets\\Pittsburgh250WordSource\\searchindex.js"
                }
        ]
}
```

   Each target is described by two values:
   - "path" is the virtual path you set up on the web server (IIS, Apache, etc.)
   - "index" is a physical path to the search data of the target, the "searchinddex.js" file in the Target folder. Note that the character '\' must be expressed as '\\' in these strings.
5. Make sure you save all changes made to the "settings.json" file.
6. Repeat step 3 from Setting up a server for a single target on page 22 to start the search server.

Note: The folder structure described in these steps is only an example; you can use any folder structure you wish.

Updating target configurations

If you change the port number setting in the server configuration, you must update this setting in Target(s).

1. Open the command line window and go to the folder containing the search server files, for example, "c:\d2h\nodejs" or "c:\d2h\targets\Pittsburgh250XMLSource\js\nodejs" (See step 3 of Setting up a server for a single target on page 22).
2. Execute "node searchserverconfig.js"; this will update port number in the configuration files in Target(s) folder.

   The utility "searchserverconfig.js" uses the "settings.json" file to update target configurations.

NetHelp Classic Search Options

You can specify either a **JavaScript Client** or a **Java Server** search for NetHelp Classic. This option is set using the **Search Type** field in the Help Targets on page 133 dialog box (**Home** tab > click the dialog box launcher on the **Target** ribbon group).

If your NetHelp Classic Target will be installed locally, **JavaScript Client** search is recommended to avoid dependence on Java being installed on end-user machines.

If your NetHelp Classic Target will be installed on a server, **Java Server** search is recommended for large projects. NetHelp Classic with **Java Server** search deployed on a server does not require that Java be installed on end-user machines, but Java does need to be installed on the server, along with a few other setup requirements. See NetHelp Classic Server Installation on page 18 for instructions.

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® 7/8 and Vista at \\Users\<username>\Documents\My Doc-To-Help Projects.

To change the default location for Doc-To-Help projects, use the **Options** dialog box. Choose the **File** tab > **Tools** > **Options**. The **Options** dialog box will open. (Or, click the **Doc-To-Help Options** button at the bottom of the **File** tab menu.) Select the **Files** button. See Setting Doc-To-Help Options on page 34 for more information.

Within your Doc-To-Help project folder:

- Word, HTML5 (.xml) and/or HTML 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 182.

Template and Style Sheet locations

Doc-To-Help Source and Target **Templates** are stored in:

- Windows® 7/8 and Vista at C:\Users\ (user name)\AppData\Roaming\Microsoft\Templates

If you uninstall Doc-To-Help and reinstall a newer version, the templates will remain in that folder. However, if one of your customized templates uses the same name as one of Doc-To-Help's default templates, and there is a newer version of that template in the installation, Doc-To-Help will save your version of the template to the C:\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp\Backup\Templates folder.

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 11.

Theme locations

Customized Themes are stored by default in \\My Doc-To-Help Projects\Doc-To-Help\Themes. You can change this location using the Doc-To-Help **Options** dialog box, **Files** button. (To open the **Options** dialog box, choose the **File** tab > **Tools** > **Options**.)

Doc-To-Help Quick Reference Page

Creating and Converting Projects on page 87	Guide to Templates and Styles on page 7
Setting Project Properties on page 182	Introduction to Single Sourcing on page 3
Creating Help Targets on page 133	Working with Source Documents on page 263
Defining Character/Paragraph Styles and Topic Types on page 168	Using D2HML (Doc-To-Help Markup Language) on page 303
Setting the Help Window Display on page 163	Variables Window on page 129
Marking Text as Conditional on page 313	Managing Topics on page 321
Creating an Index or Groups on page 327	Creating a Table of Contents on page 329
Creating a Glossary on page 299	Implementing Context Sensitive Help on page 186
Building a Target on page 341	Doc-To-Help Outputs and Deliverables on page 13
Setting Document Properties on page 297	Customizing Themes
Editing a Template on page 10	Editing a CSS on page 176

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 105 for a quick overview of the interface.

Watch the video: Doc-To-Help's Intuitive Workflow (3:37)

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 **File** tab 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 **File** tab 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 87 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 7 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 182 for information on default source folder locations, context ID settings, keeping page breaks in manual targets and more.
- See Creating Help Targets on page 133 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 168 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 163 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 **Insert** 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 3
- Working with Source Documents on page 263

- Using D2HML (Doc-To-Help Markup Language) on page 303
- Variables Window on page 129
- Marking Text as Conditional on page 313

4. Manage Topics

Once you have created content, you can check out the Topics window on page 127 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 321
- Creating an Index or Groups on page 327
- Creating a Table of Contents on page 329
- Creating a Glossary on page 299
- Implementing Context Sensitive Help on page 186

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 341 for a quick overview of choosing and creating a Target.

See Doc-To-Help Outputs and Deliverables on page 13 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, creating Modular Help systems, and documenting class libraries with Microsoft® Sandcastle.

For more information, see:

- Working on a Team on page 381
- Creating a Modular Help System on page 397
- Documenting Your Class Library with Microsoft Sandcastle on page 409

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.

License and Redistributable Files

Doc-To-Help 2014 is developed by MadCap Software, Inc. 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 2014 is developed and supported by MadCap Software, Inc.

To contact support, please go to http://www.madcapsoftware.com/support.

System Requirements

Computer/Processor	PC with Intel® Pentium® II 500 MHz or greater processor
Memory	512 MB of RAM Minimum
Hard Disk Space	200 MB of hard disk space for the Doc-To-Help application and related files.
Operating System	Microsoft® Vista, Microsoft® Windows 7, Microsoft® Windows 8/8.1
Microsoft Word	Word 2007, Word 2010, or Word 2013
Microsoft FrontPage	Microsoft® FrontPage® 2003 SP2
Adobe Dreamweaver	Adobe® Dreamweaver® MX2004, 8.0, 9.0, CS3, CS4, CS5
.NET Framework	Microsoft® .NET Framework version 4.0 or later The Doc-To-Help installation will check for the presence of the Microsoft .NET 4.0 Framework. If not found, it will be installed automatically.
Browser	Microsoft® Internet Explorer® 8 or greater (for use by the Theme Designer), Google Chrome,

	Mozilla Firefox, or Safari

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

If using Doc-To-Help's SharePoint integration, Doc-To-Help collaboration features are compatible with Microsoft® SharePoint® 2007, 2010, and 2013, as well as Office 365 (SharePoint Online). SharePoint 2010 or 2013 is required when publishing Doc-To-Help Targets to a SharePoint Wiki Library. SharePoint 2010 is recommended when working with Translation Libraries (SharePoint 2013 does not include this feature). Translation Libraries cannot be managed in Windows SharePoint Services 3.0, because WSS 3.0 does not support them.

Prerequisite for Office 365 (SharePoint Online):

- Windows Identity Foundation http://www.microsoft.com/download/en/details.aspx?id=17331

Supported operating systems for Office 365:
Windows 7, Windows 8, Windows Server 2008 R2, Windows Server 2008 Service Pack 2, Windows Vista Service Pack 2, Windows Server 2003 Service Pack 2.

If storing your Team Project in a Team Foundation Server repository, Doc-To-Help supports Microsoft Team Foundation Server 2008, 2010, and 2012.

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.

Installing Doc-To-Help

Prerequisites and installation notes

See System Requirements on page 31 for a complete list of software and hardware requirements.

- Close Microsoft® Word before beginning Doc-To-Help installation. Microsoft® Outlook should also be closed if Microsoft Word is your default email editor.
- Turn off your anti-virus software. (You can turn it back on after installation is complete, although you may need to modify it later.
- 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.
- To install Doc-To-Help, you must have administrative privileges.
- Doc-To-Help requires the Microsoft .NET Framework 4.0, and will install it for you automatically if it is not found during installation. If Doc-To-Help installs the .NET Framework, you may be required to restart your machine before continuing the Doc-To-Help install. After the restart, you must log in as the same user who began the install, or the installation won't continue.
- Silent installs of Doc-To-Help are supported.

Installation Instructions

1. Go to http://www.doctohelp.com/download to download the latest version of Doc-To-Help. Save the .exe file to your computer.
2. Double-click the .exe file to launch the installer.
3. Follow the Wizard to complete the installation.

See the Doc-To-Help Installation Guide for additional information.

Silent Installation

Note: To install Doc-To-Help silently, Microsoft .NET Framework 4.0 must be already installed on those machines (because the automatic .NET Framework installation requires a restart).

Doc-To-Help supports all standard installer command line options for Windows, see http://msdn.microsoft.com/en-us/library/windows/desktop/aa367988(v=vs.85).aspx

To install DocToHelp silently into a specific folder with log, use the following command:
"pathtexeifile" /qn /l*v "pathtologfile" INSTALLLOCATION="pathtofolder" /i
Note: The /i switch must be the last switch in the command line.

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 **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 **File** tab > **Tools** > **Options**. The **Options** dialog box will open.
2. Click the **Updates** button.
3. Select the **Do not check for updates automatically** radio button.
4. Click **OK**.

To check for updates manually

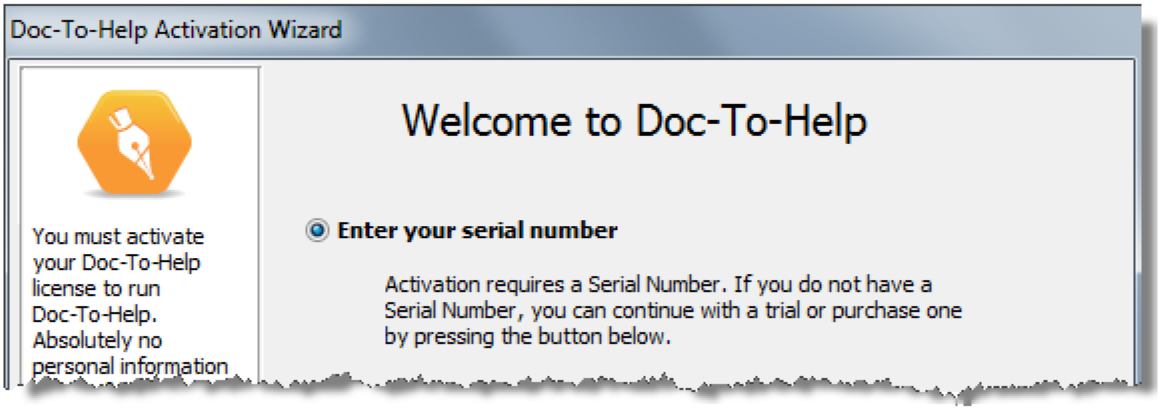
Choose the **File** tab > **Tools** > **Check for Updates**.

Activating Doc-To-Help

Activation is used by Doc-To-Help to verify the authenticity of the product key used to license the software and also to ensure that keys are only used to license an approved number of machines. The activation process does not collect any customer-specific information and protects the end user's privacy completely.

To activate Doc-To-Help

You can activate Doc-To-Help after installation using the **Doc-To-Help Activation Wizard**, which will open automatically.



In the **Doc-To-Help Activation Wizard**, choose the **Enter your Serial Number** radio button and click **Next**. Enter your serial number in the field provided and click **Next** to activate.

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.

Please note if using Windows 7/8 or Vista, you may need to log into Doc-To-Help as an administrator to activate or deactivate Doc-To-Help. To log in as an administrator, right-click on any Doc-To-Help shortcut (from the **Start** menu or on your desktop) and choose **Run as administrator** from the menu. Doc-To-Help will open.

To deactivate Doc-To-Help

Choose the **File** tab > **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.

To set Doc-To-Help Options

1. Choose the **File** tab > **Tools** > **Options**. The **Options** dialog box will open. (Or, click the **Doc-To-Help Options** button at the bottom of the **File** tab menu.)
2. Select the **General**, **Files**, **Editors**, **Team**, **SharePoint**, **Updates**, **Spelling**, or **Eclipse Help** button.
3. Select the desired options.
4. Click **OK**.

General button:

- 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 documents 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 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 installing Microsoft Help Viewer target after build**. If you built a Microsoft Help Viewer target, Doc-To-Help will ask if you would like to install the target. If yes, it will open the Help Library Manager so you can add the project to the Microsoft Help Viewer and look at it.
- **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 making a copy of a document with Save As**. If saving the currently open XML document under another name using **File** tab > **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.

Files button:

- Change the default location for Doc-To-Help projects and Doc-To-Help custom Themes (See Customizing Themes with the Theme Designer on page 199.)

HTML Editors button:

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

Team button:

- Change the default “new project” colors for your Team Projects.
- Set **Check Out** and **Compare and Merge** options. Also choose your version of Microsoft Visual Studio Team Explorer. See Working on a Team on page 381 for more information about team authoring support.

Please note that you must log into Doc-To-Help as an administrator to change your version of Microsoft Visual Studio Team Explorer if using Windows 7/8 or Vista. To log in as an administrator, right-click on any Doc-To-Help shortcut (from the **Start** menu or on your desktop) and choose **Run as administrator** from the menu. Doc-To-Help will open.

SharePoint button:

- Change the defaults for opening and synchronizing documents to a SharePoint Library. When the **Check out Word documents when editing local copy** check box is selected, Word documents will be checked out of the SharePoint Library automatically when you open them in Doc-To-Help. For more on sharing documents to a SharePoint Library, see Collaborating with SharePoint on page 353.

Updates button:

- Turn the Doc-To-Help **Live Update** feature on and off. See Using Live Update on page 33 for more information.

Spelling button:

- Set the spell check options for the Doc-To-Help **Content** editor. (The **Spelling** button is on the **Editor** tab. See Editor tab on page 110 for more information.)

Viewers button:

- Specify the location of the Eclipse executable file (eclipse.exe) on your machine. Doc-To-Help requires this information to view Eclipse Help.
- Specify the location of the EPUB reader executable on your machine. Doc-To-Help requires this information to view EPUBs.
- Specify the version of the Microsoft Help Viewer to use when viewing Microsoft Help Viewer Targets. (Options available depend on your Visual Studio installation.)

Doc-To-Help Learning Library

There are a number of resources available to get you up-to-speed with Doc-To-Help quickly, including videos, blog posts, free webcasts and more. Doc-To-Help also includes 11 sample projects that demonstrate the editors, plus are examples of software documentation, an employee handbook, training materials, Responsive Help, and API/SDK documentation with Microsoft Sandcastle. There is even a sample in German.

See Doc-To-Help's Learning Library for complete information.

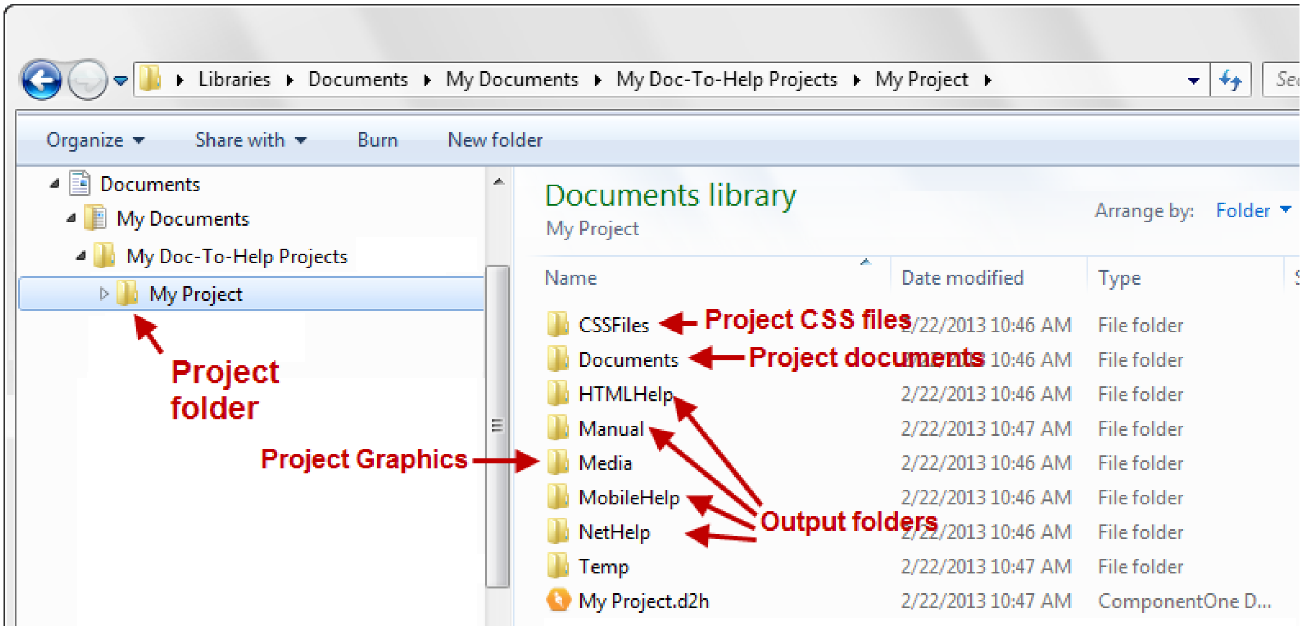
Trademark Statement

Doc-To-Help is a registered trademark of MadCap Software, Inc. Microsoft, FrontPage, Vista, NT, Visual Studio, Team Foundation Server, SharePoint, Sandcastle, Office 365, 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, Google Translate, Google Custom Search, and Google Analytics are trademarks of Google. Safari is a registered trademark of Apple, Inc. Opera is a trademark of Opera Software ASA. Disqus is a trademark of Disqus. All other product and company names herein are the property of their respective owners.

Migrating a Project

Many of us collaborate on Doc-To-Help projects using Team Authoring, TFS, or SharePoint, but sometimes we simply need to transfer a project to a coworker or a client. This is done by sending them the entire project folder, but here are a few tips to make the transition easier.

Here is a typical project folder:



Transfer the project

By default, Doc-To-Help projects are stored in the following folder:

- C:\Users\YourName\Documents\My Doc-To-Help Projects

So when you give a colleague a project, they should drop the entire folder in that location. (This default location can be changed in Doc-To-Help by choosing **File** > **Doc-To-Help Options** > **Files**.)

Tip: If the file size of the project is an issue, you can delete the contents of all the Target folders before transferring the project (because your colleague can always build the projects again).

Transfer the Custom Templates and Themes

If a project has custom Word templates, and/or Themes, those need to be transferred also — just drop them into the appropriate folders.

Drop custom Word templates here (they will have the file extension of .dot):

- Windows® 7/8 and Vista at: \\Users\ (user name)\AppData\Roaming\Microsoft\Templates

Themes are stored here, so drop them in the appropriate folder:

- C:\Users\YourName\Documents\My Doc-To-Help Projects**Doc-To-Help**\Themes\(name of output)\Themes

To make things quicker, you could just transfer the entire "Doc-To-Help" folder (bolded above) to your colleague. Then all they need to do is drop that folder into the "My Doc-To-Help Projects" folder (C:\Users\YourName\Documents\My Doc-To-Help Projects).

Note: If your project uses Doc-To-Help's built-in Editor and you edited the CSS to create a custom one, those changes are stored in the Project folder (in the **CSSFiles** folder), so they transfer along with the project.

Get Started

Once transferred, open the project by choosing **File** > **Open Project**. The file extension of all Doc-To-Help projects is .D2H.

Double-click on any document in the **Documents Pane** to begin editing.

Doc-To-Help Quick Tour

This mini, self-paced tutorial will show you how easy it is to create a project in Doc-To-Help, as well as build your output, apply styles, create links, and more. You will build a project about Pittsburgh using two existing documents. This Quick Tour uses Microsoft® Word as the document editor, but the same principles apply when you are using Doc-To-Help's built-in Content Editor or an HTML editor.

The *Doc-To-Help in 5 Lunches Tutorial Series* is another quick start resource you should check out. It has Word and Built-In Editor versions, plus accompanying videos. See Learn How Doc-To-Help Works in Five Lunch Breaks.

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, or click the **Help** button to open Doc-To-Help online Help.

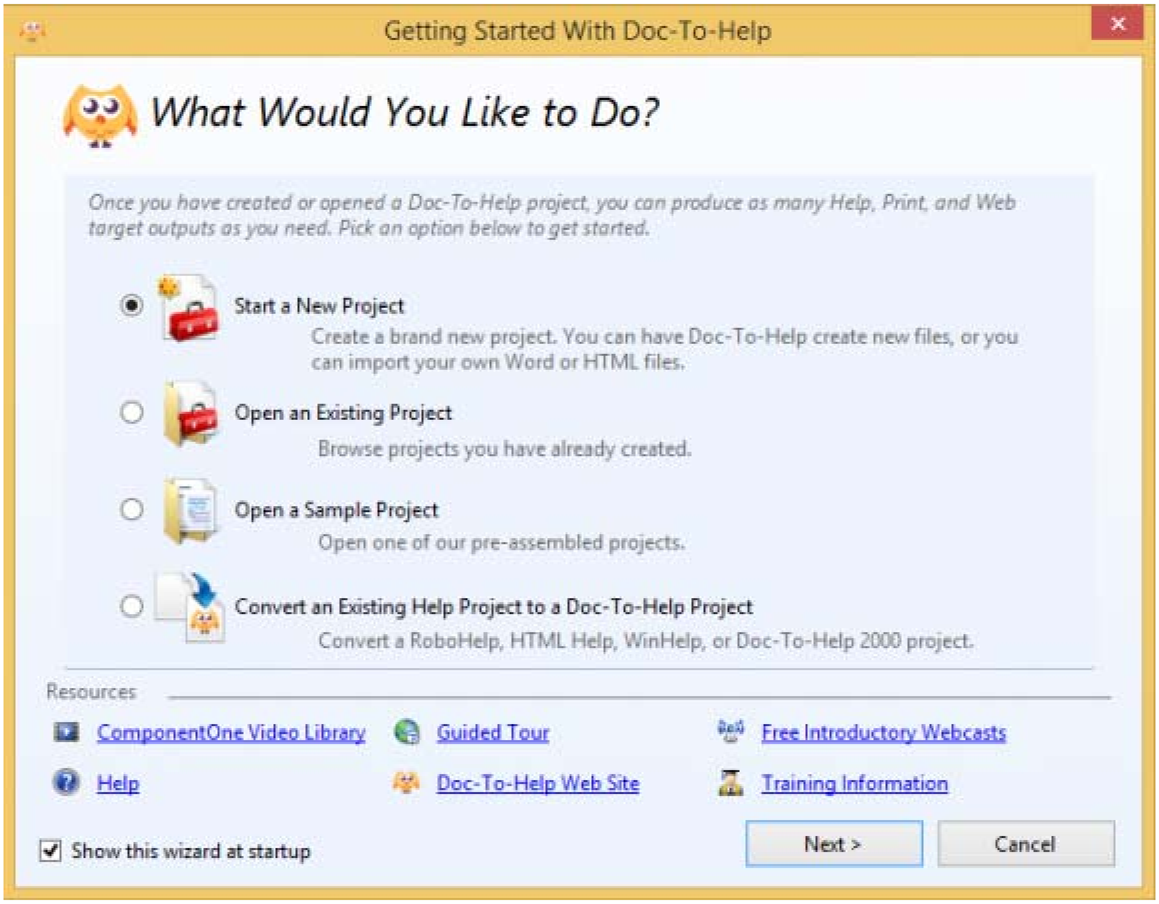
Watch the videos: Starting a New Project in Doc-To-Help (3:48) Tour of the Doc-To-Help Interface (5:26)

Let's get started …

Creating a Project

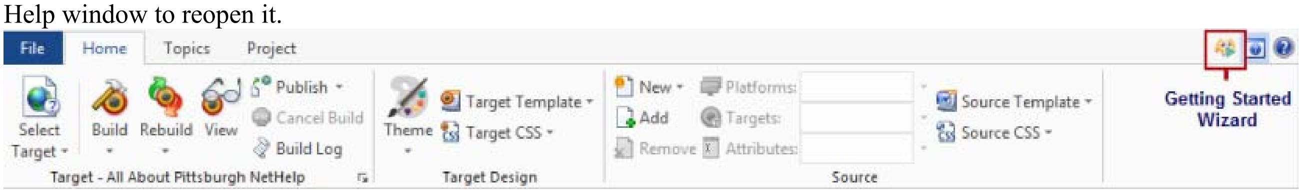
Creating a new project is quick and easy with the **Getting Started Wizard**.

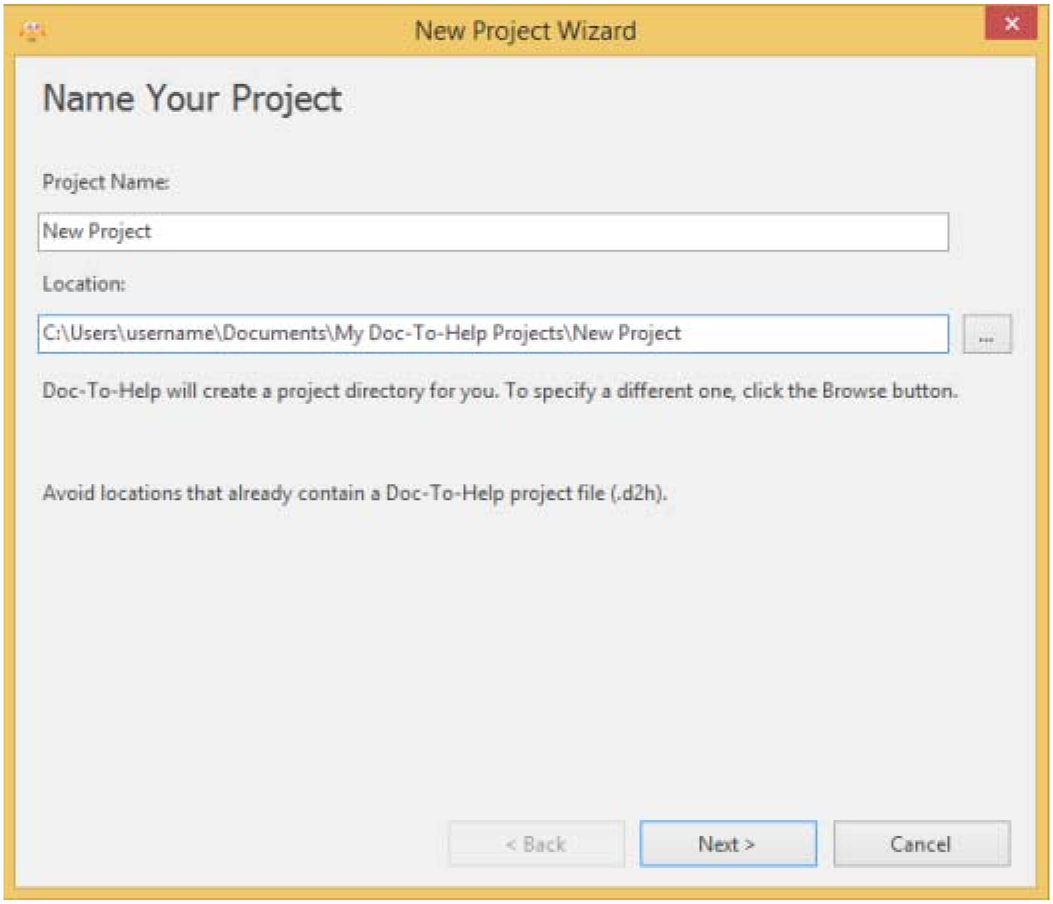
This wizard opens when you open Doc-To-Help. It can also be used to open an existing project, open a sample project (a great way to test drive a Word or HTML5 project), or convert an older existing project to a Doc-To-Help project.



1. Choose **Start a New Project** and click **Next**. The **New Project Wizard** appears.

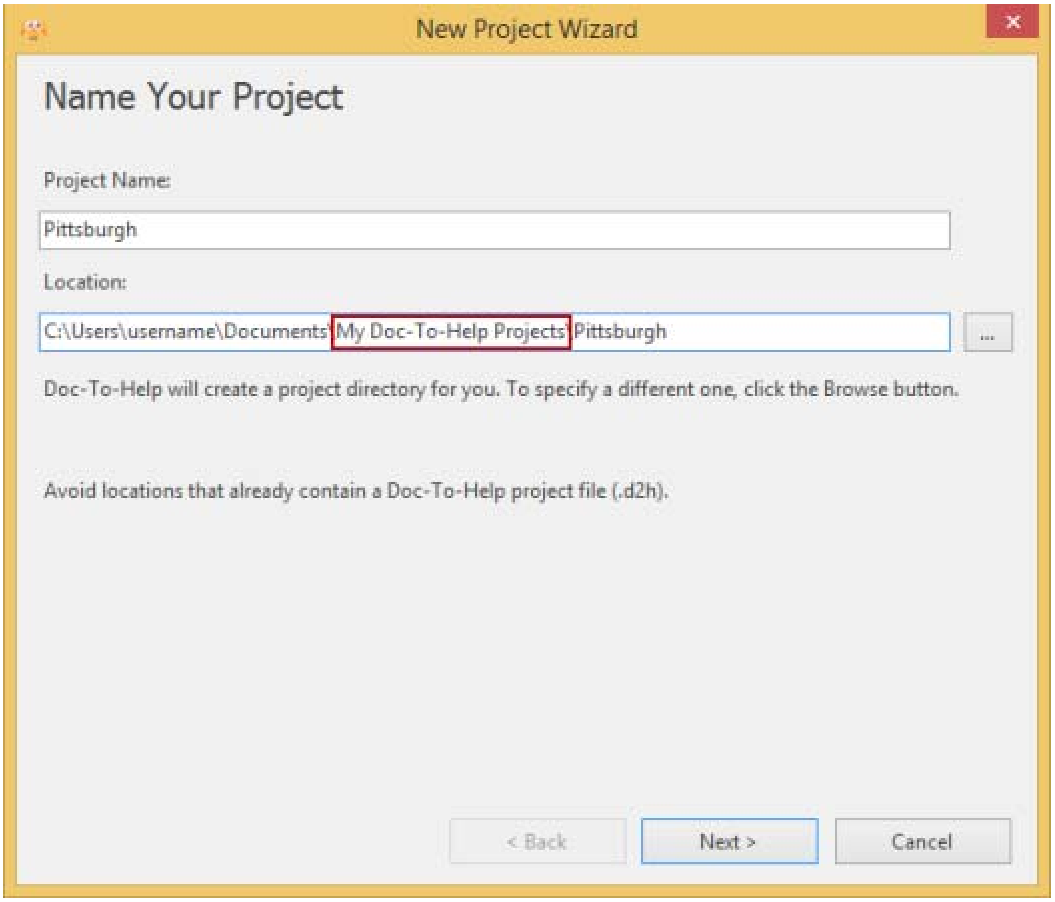
Note: If you have closed the Wizard, click the **Getting Started Wizard** button in the upper-right corner of the Doc-To-Help window to reopen it.





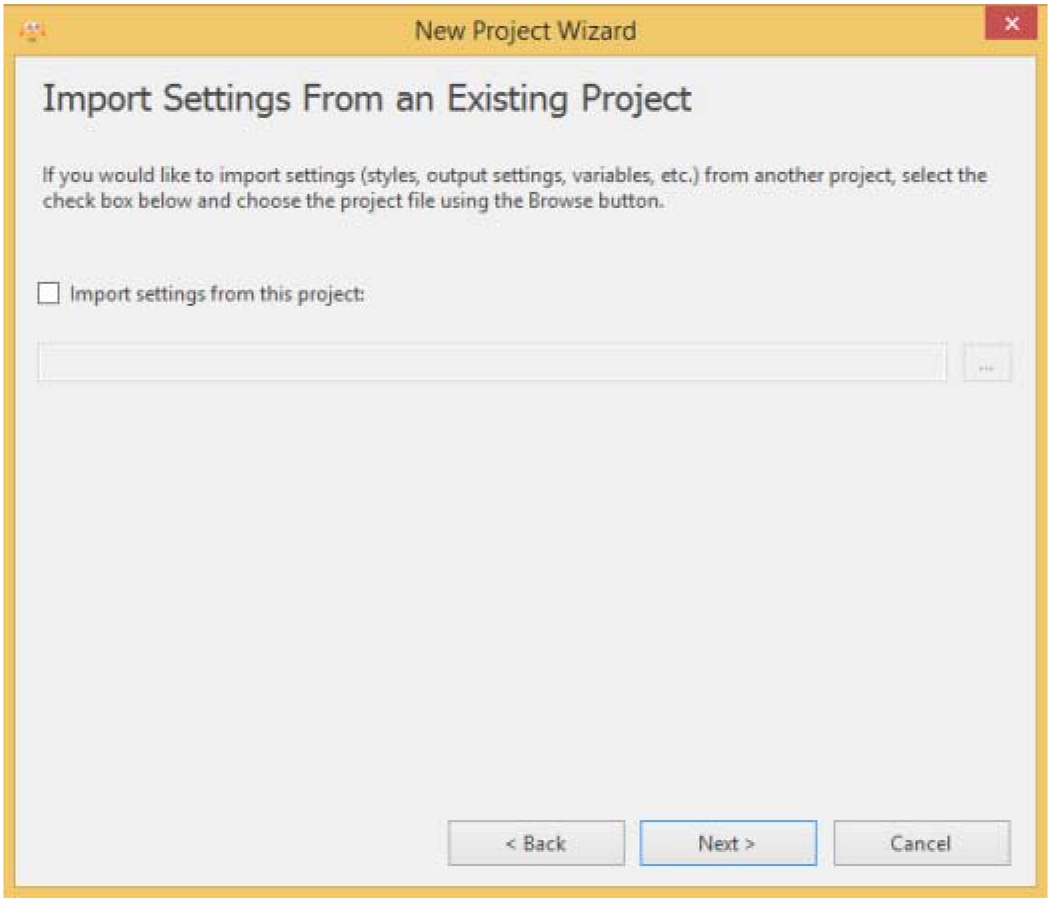
2. Name your project **Pittsburgh**. Doc-To-Help saves the project in a folder with the same name within the default **My Doc-To-Help Projects** folder. This makes it easy to keep your projects organized.

Note: You can change the default folder by clicking the **File** tab > **Doc-To-Help Options** button. Choose the **Files** button.



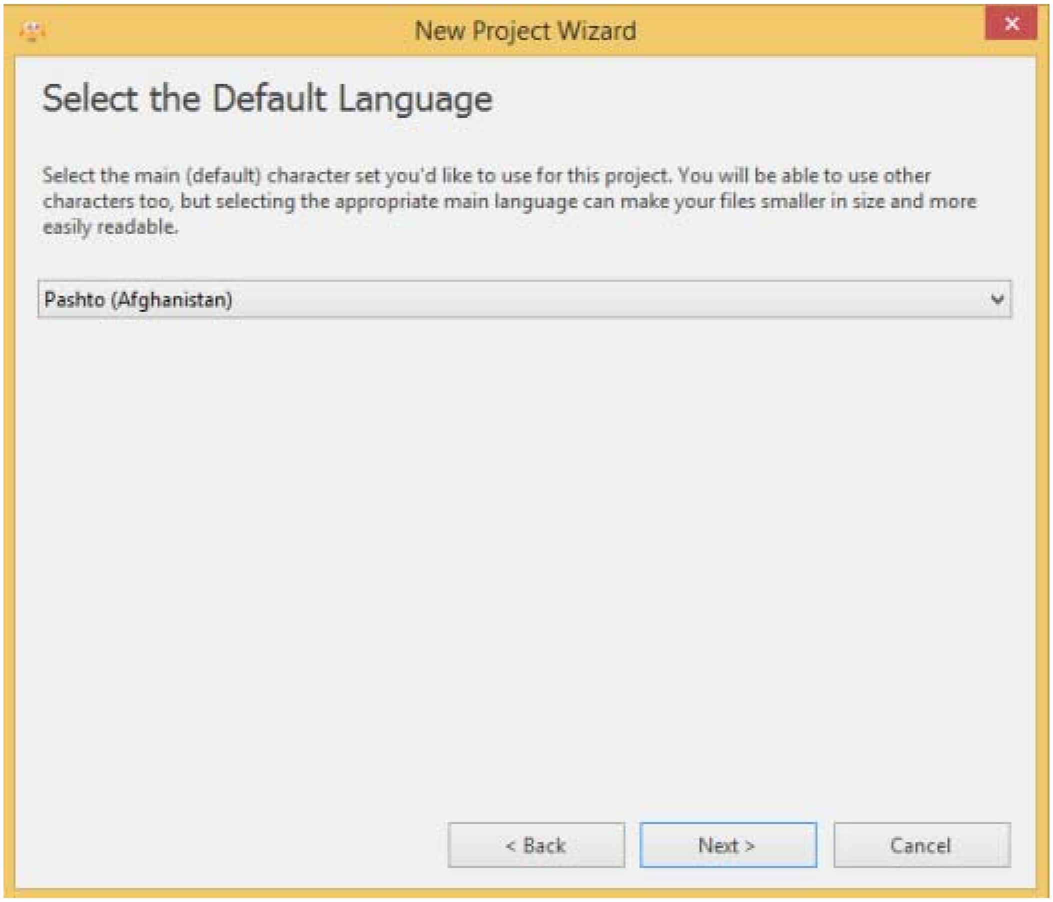
3. Click **Next**.

   The next screen gives 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.



4. Click **Next**.

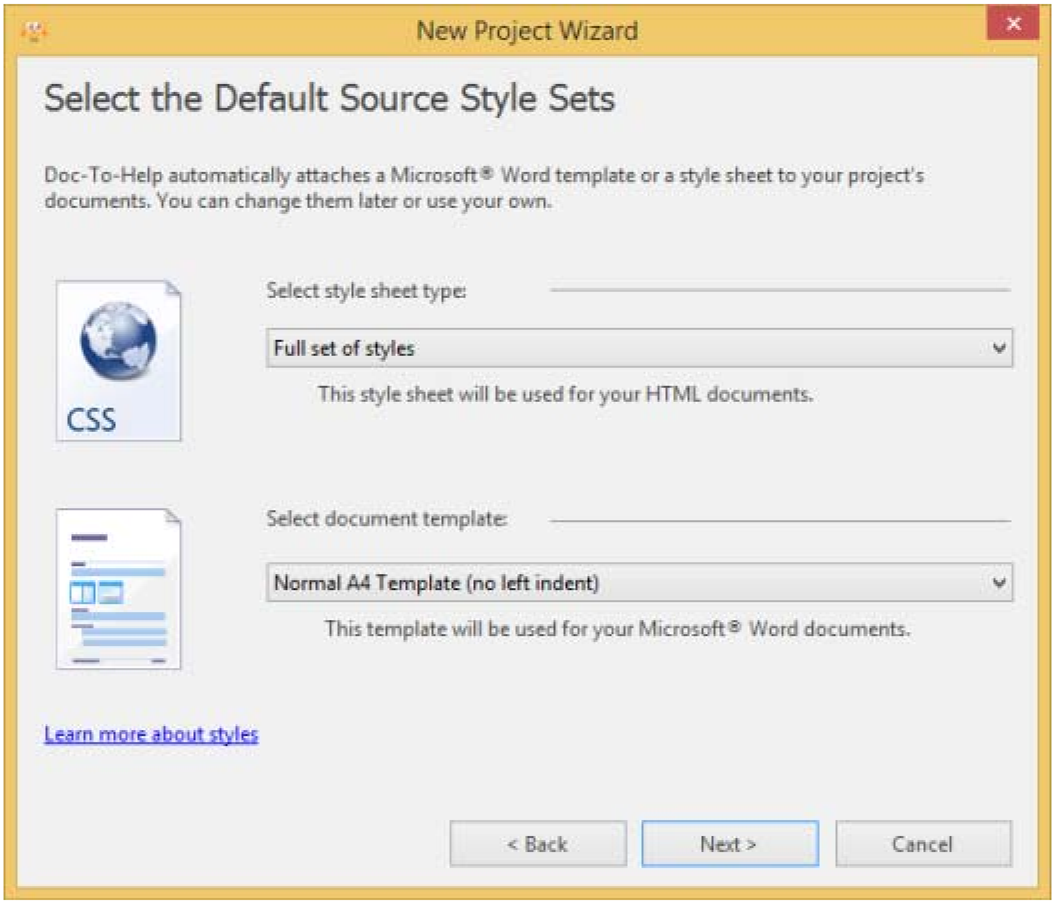
   Here you can select the default language for your project. By default, it is U.S. English, but you can choose another from the drop-down list. Doc-To-Help chooses the correct character set based on the language chosen.



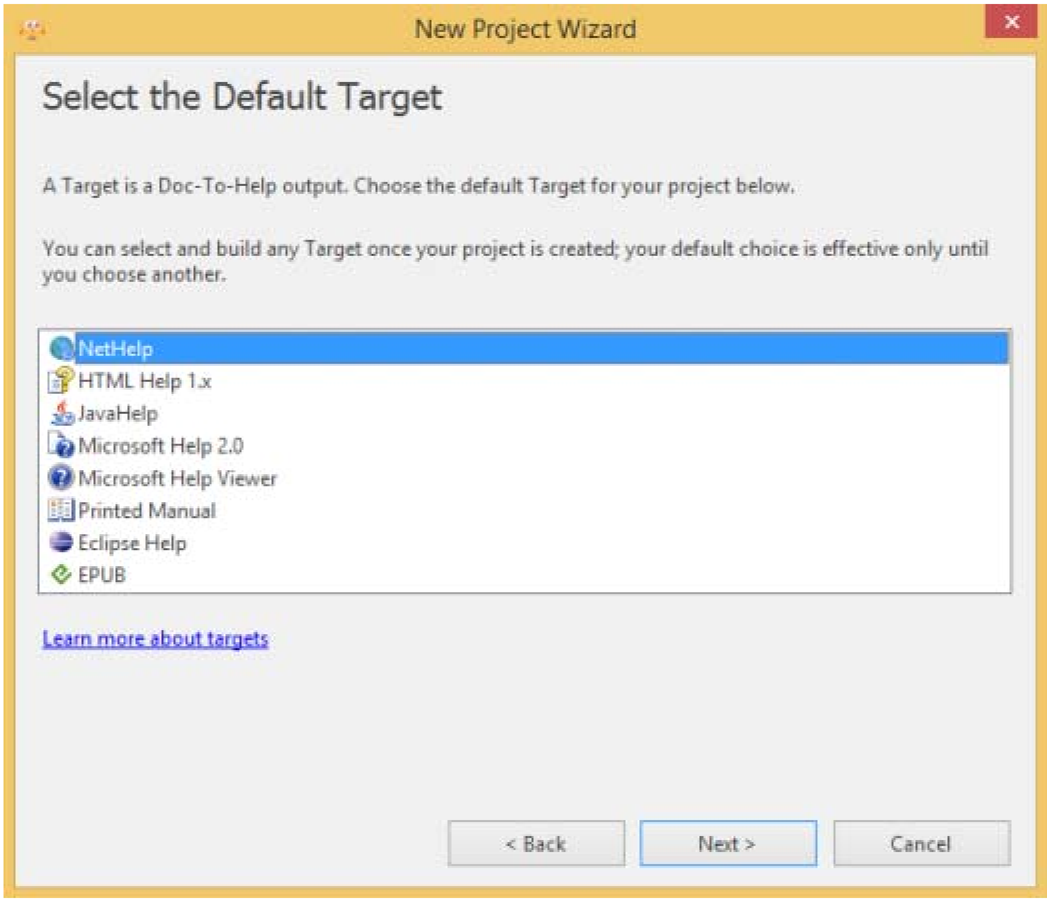
Here you can use the wizard to select the style sheet (.css) for your HTML5 or HTML Source documents and the template (.dot) file for your Microsoft® Word Source documents.

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 Content Editor, or an HTML editor) **and** your final targets, or the online Help and printed manual outputs.

Doc-To-Help automatically chooses your target style sheet and template based on your selections.

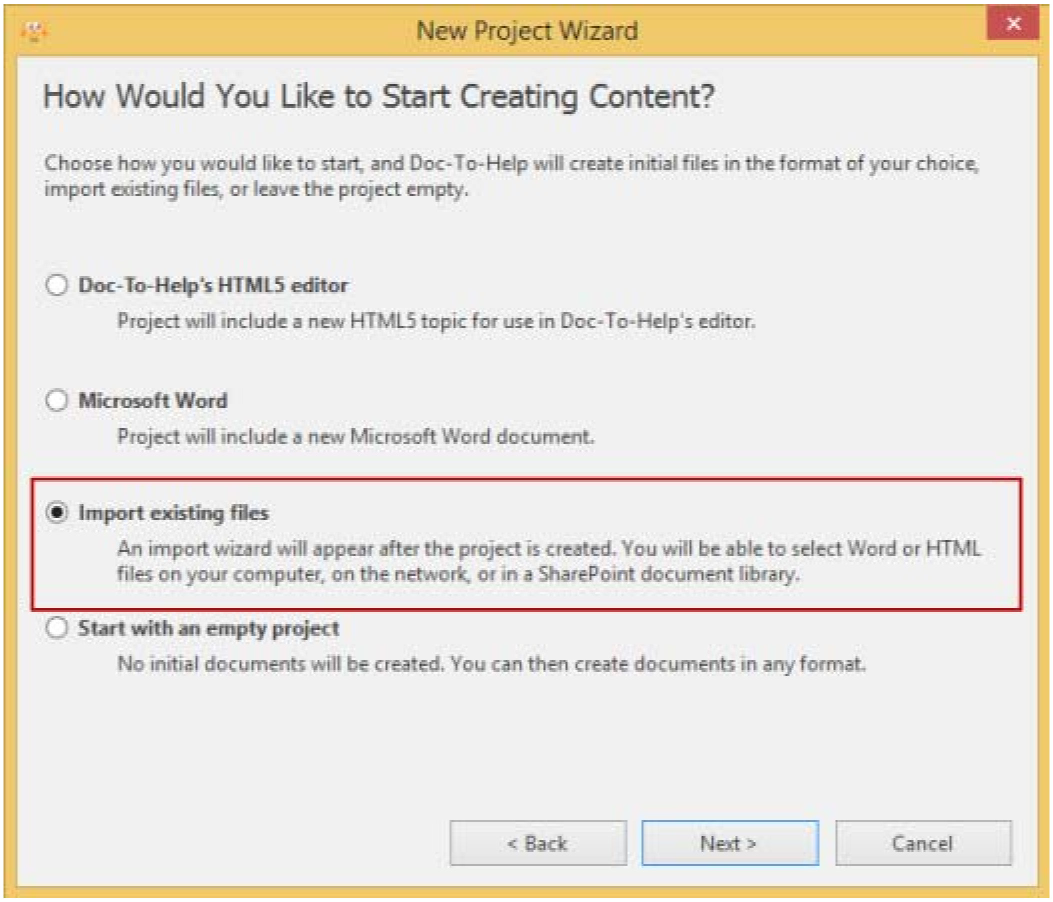


5. Click **Next**.
1. Leave the default settings of **Full set of styles** and **Normal 8-1/2" x 11" Template (no left indent)** and click **Next**.
2. In the next window, you can select your default Help target. You can generate any of these targets from your project at any time, but the default chosen here will be the one displayed when you open your project.
6. 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.



7. Click **Next**.

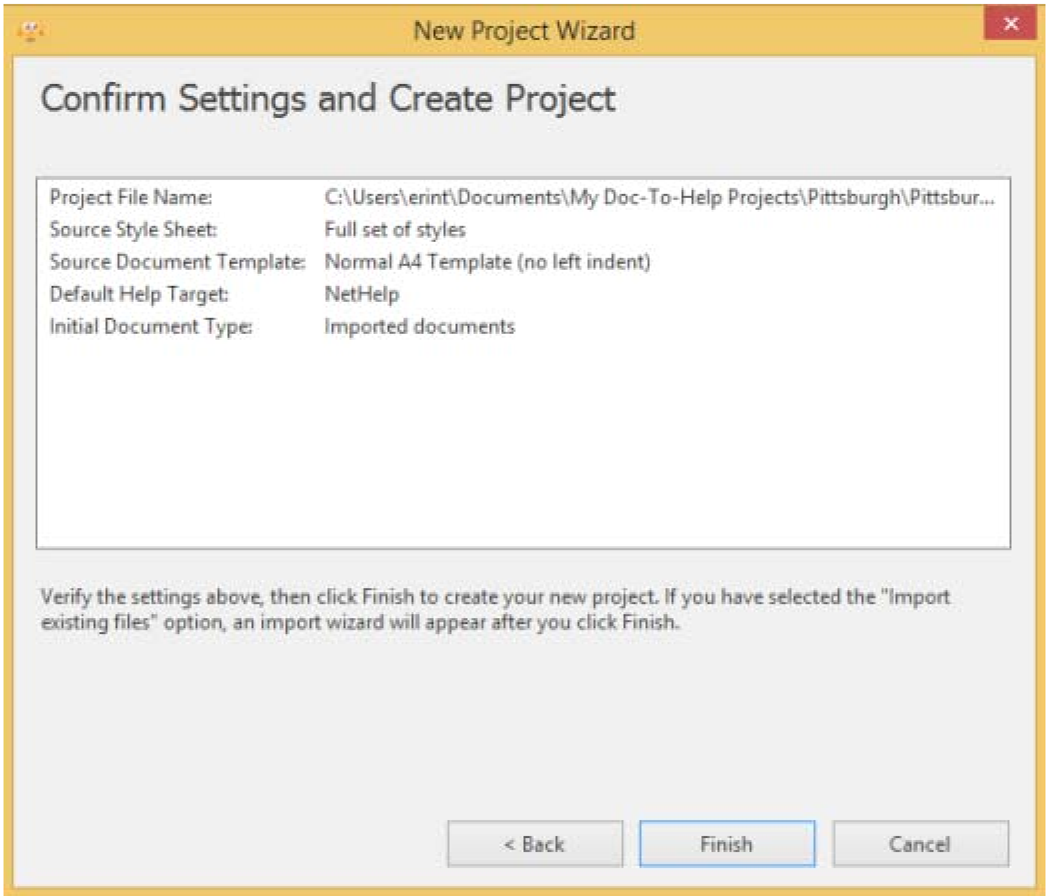
   Here you can select the type of document you would like your first document to be. We are going to use an existing document to start our project.



8. Choose **Import existing files** and click **Next**.

Note: Doc-To-Help will launch an import wizard after the project is created to make the import process simple.

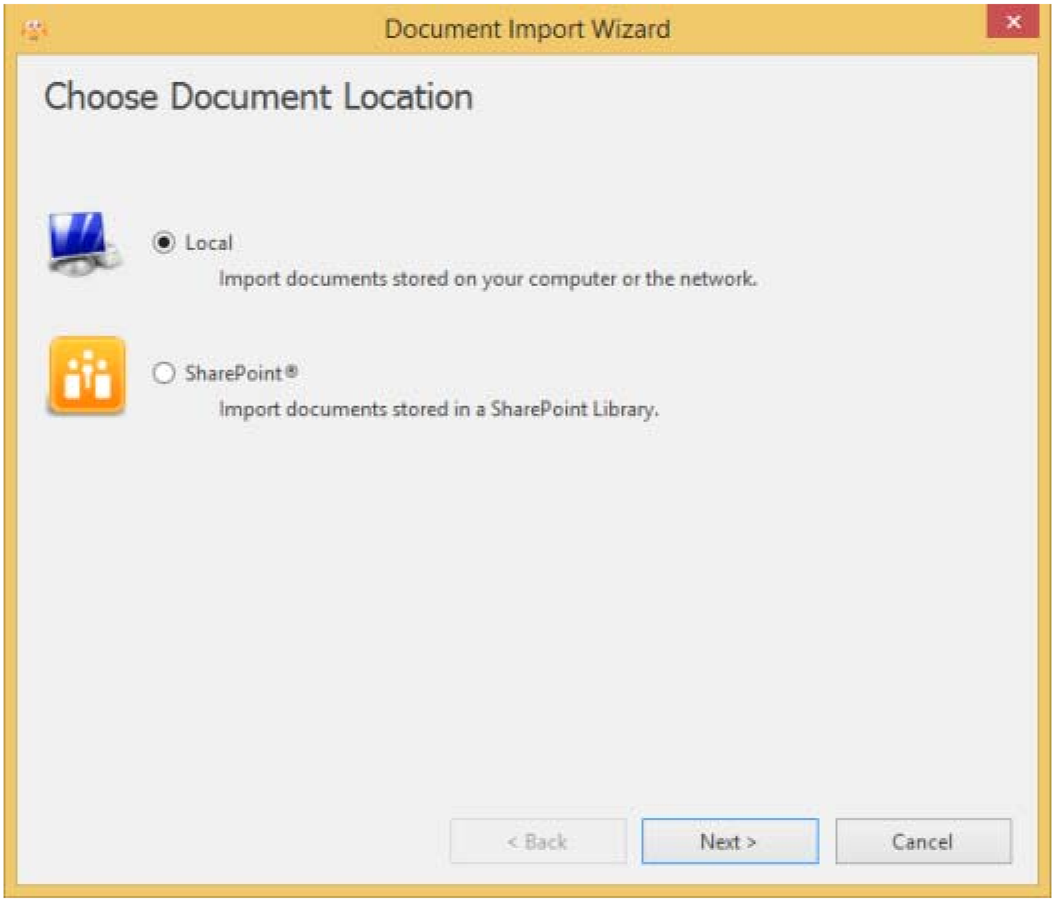
9. Review the new project information and click **Finish**.



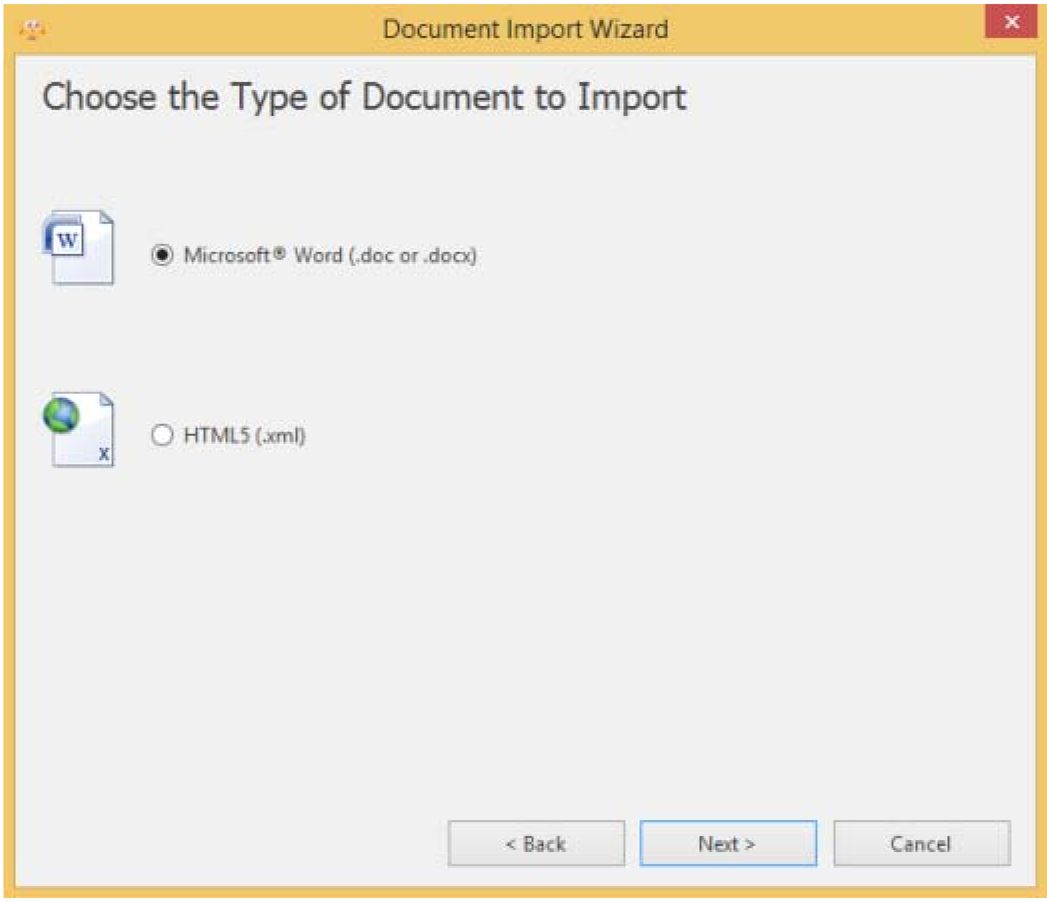
Importing a Document

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

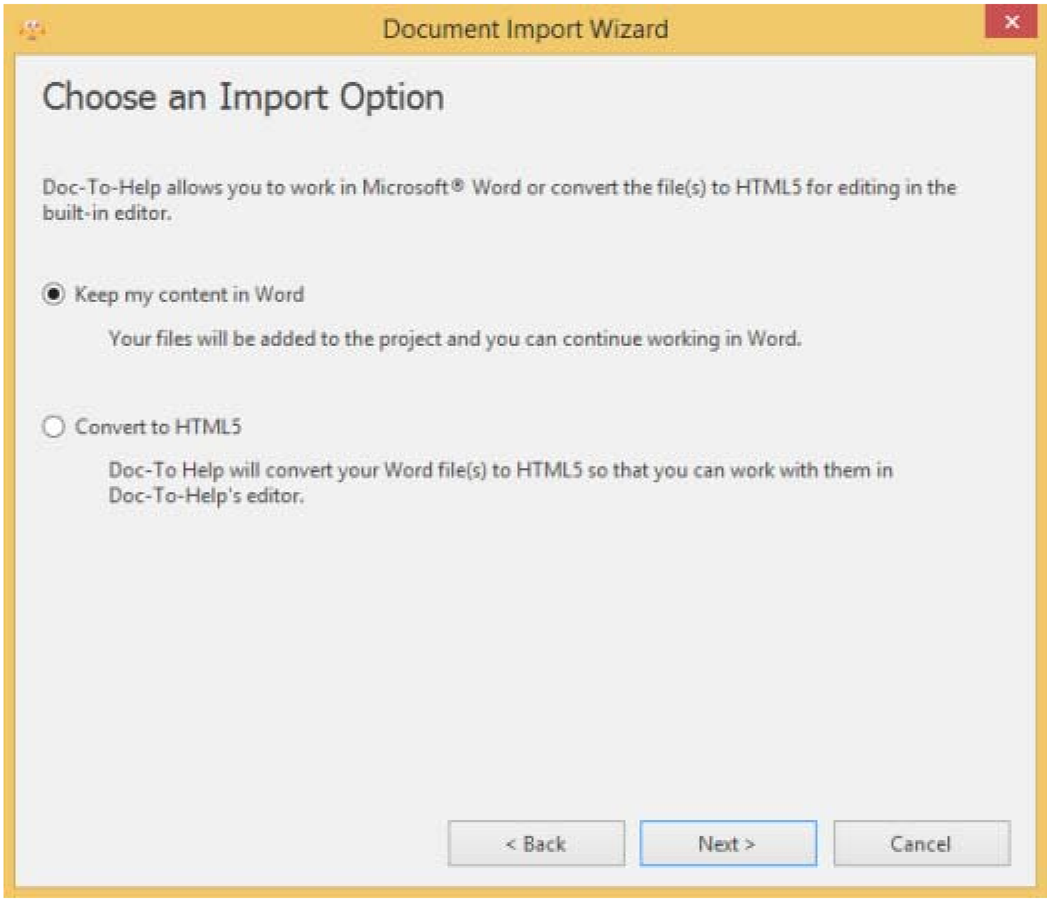
1. Choose **Local** to import documents stored on your computer and then click **Next**.



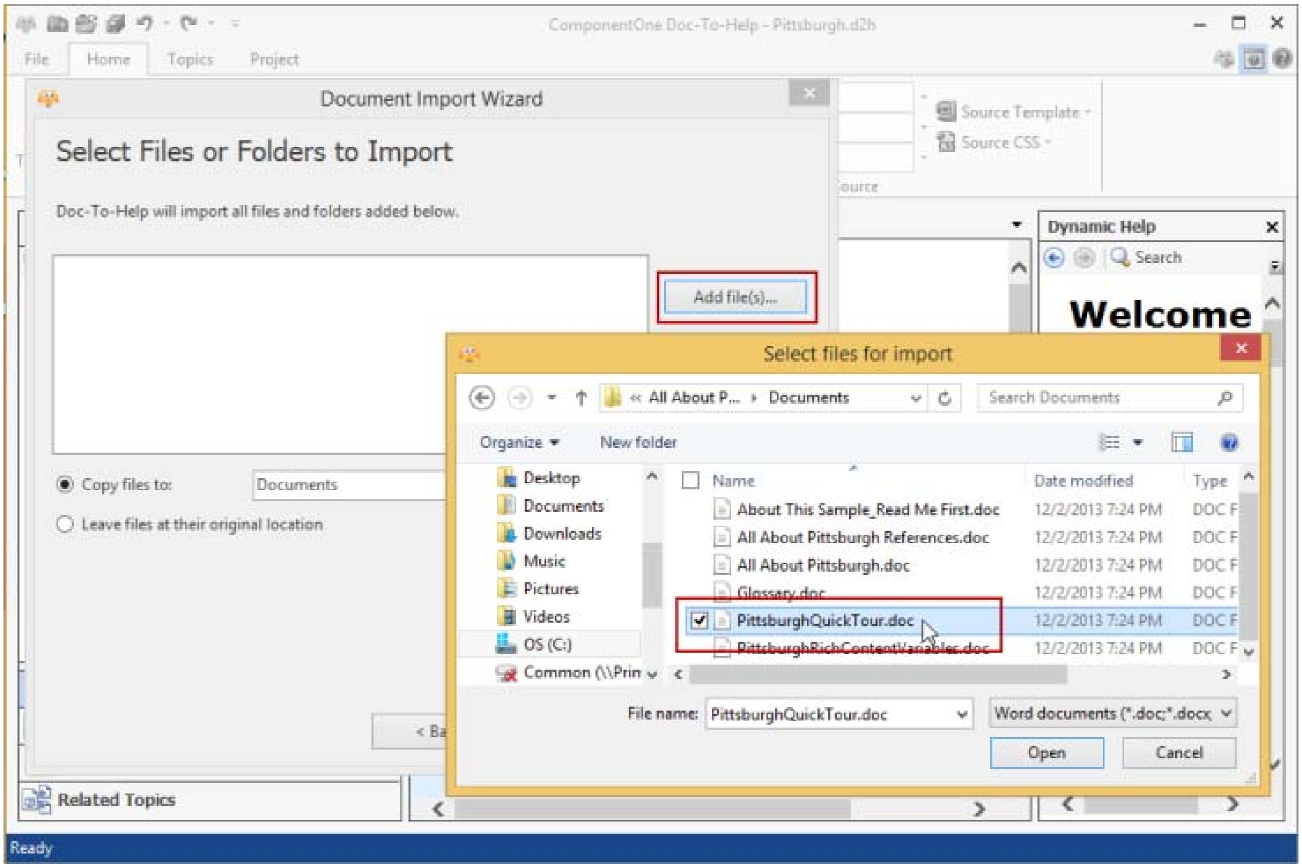
2. Choose **Microsoft® Word** and click **Next**.



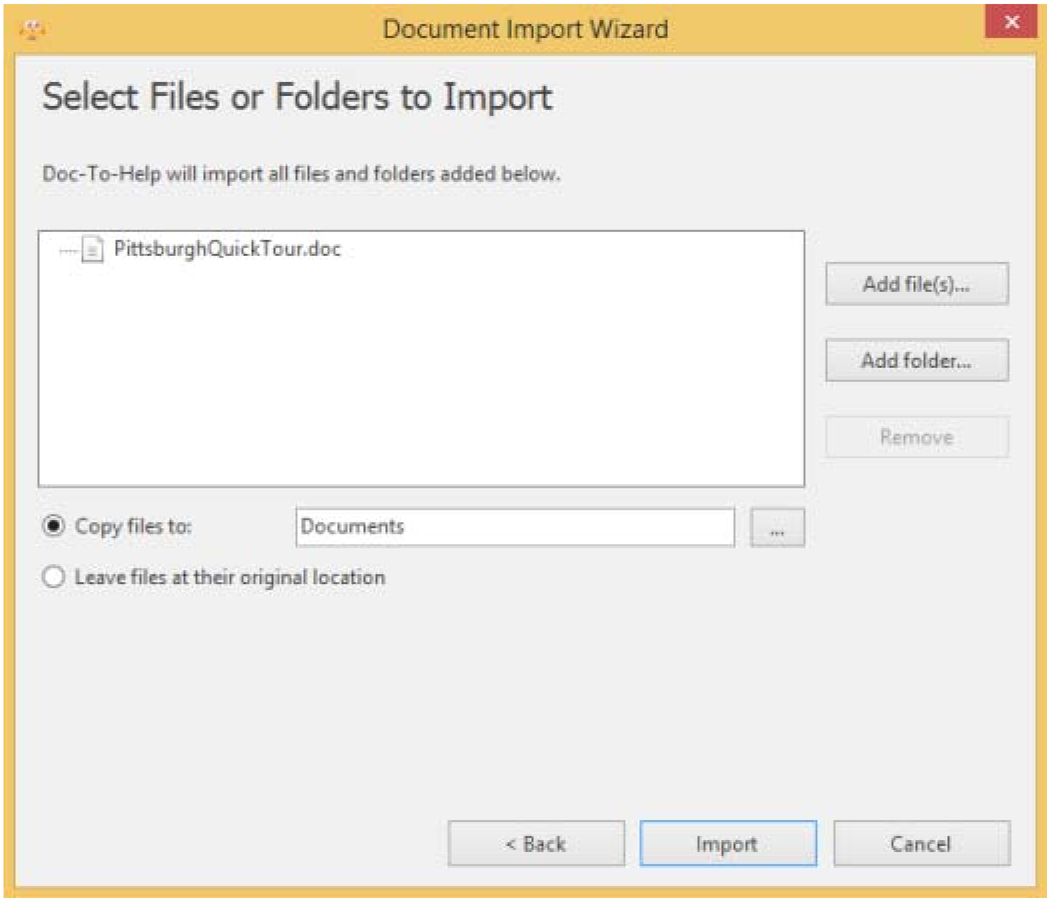
3. Choose **Keep my content in Word** and click **Next**.



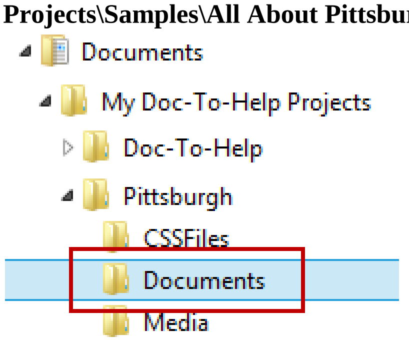
4. Click the **Add file(s)** button to select the file to add to the project. The **Select Files for Import** dialog box automatically opens to the **..\Users\<username>\Documents\My Doc-To-Help Projects** folder.
5. Double-click through the **Samples**, **All About Pittsburgh Sample Word**, and **Documents** folders and select the **PittsburghQuickTour.doc**.



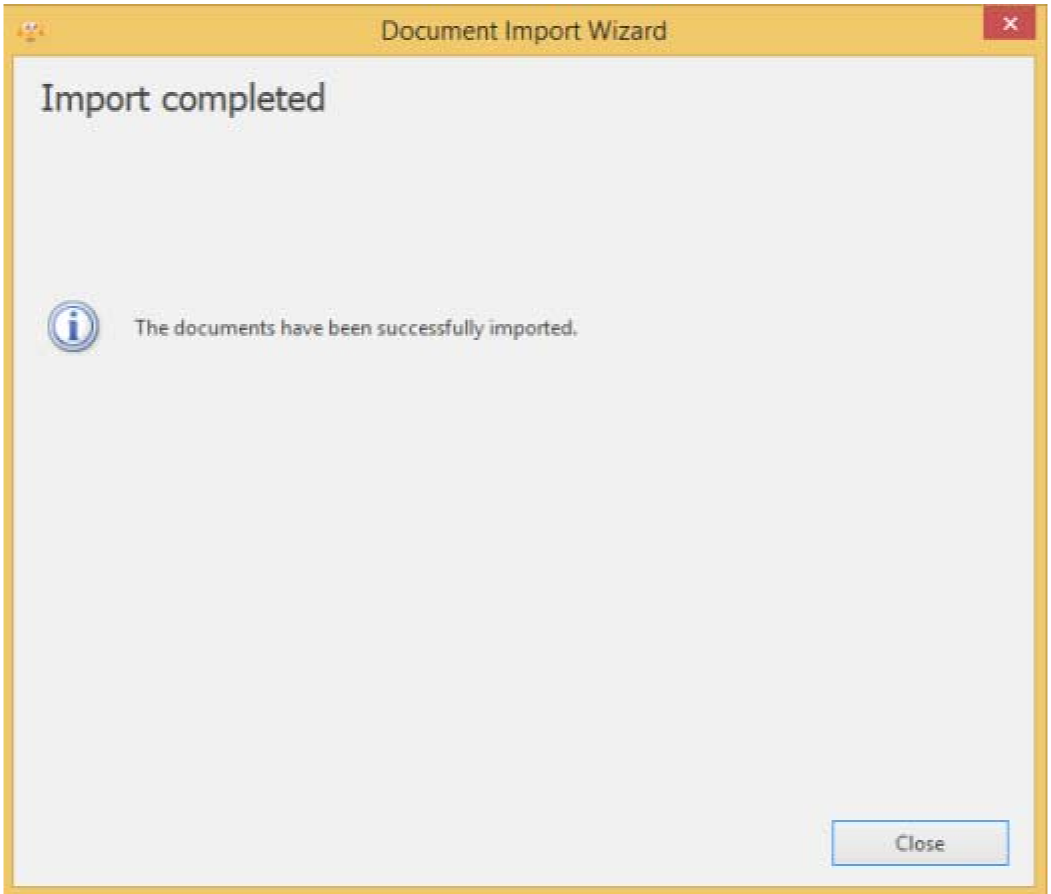
6. Click the **Open** button in the **Select files for import** dialog box. **PittsburghQuickTour.doc** appears in the wizard.



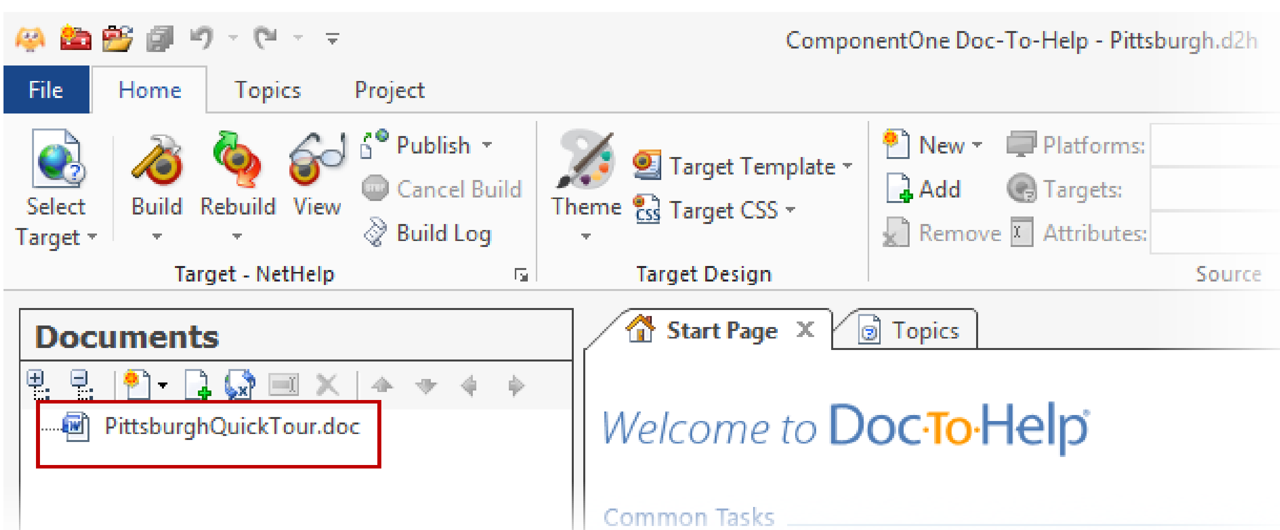
Note: Word source documents are stored in the **Documents** folder of the project. In this example, Doc-To-Help will automatically copy the **PittsburghQuickTour.doc** to the **..\Users\<username>\Documents\My Doc-To-Help Projects\Samples\All About Pittsburgh Sample Word\Documents** folder.



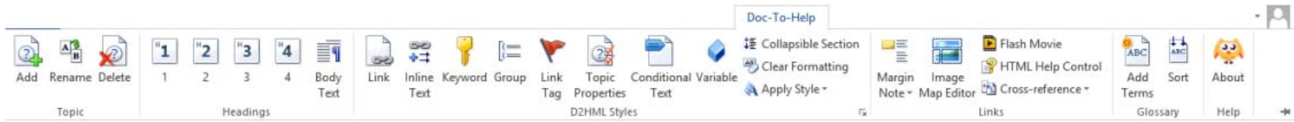
7. Click **Import** and the document is imported.



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



9. From the **Documents** pane, double-click the **PittsburghQuickTour.doc** to open it. In Microsoft® Word, take a look at the **Doc-To-Help** tab of the ribbon. With it, you can create links, add index entries, and more. 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.

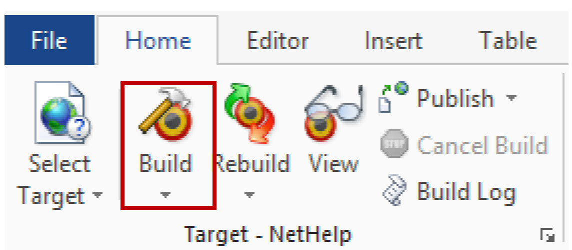
Building and Examining a Target

So far, we have done very little, but we have 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, let's take a quick look at our Word document. If it is not already open, double-click the **PittsburghQuickTour.doc** to open it.

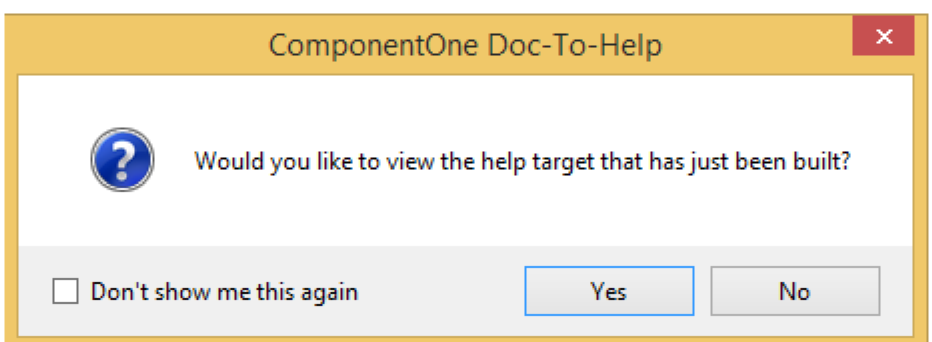
It is structured like this:
Heading 1 — All About Pittsburgh
Heading 1 — Pittsburgh Sports
Heading 2 — Football
Heading 2 — Baseball
Heading 2 — Hockey
Heading 1 — Pittsburgh Sites

Doc-To-Help will use this structure to create topics, the Table of Contents (TOC), and automatic links. Close the document so we can build a target to see how it looks.
In Doc-To-Help:

1. On the **Home** tab, in the **Target** group, click the **Build** button. The default **NetHelp Target** (chosen earlier) is built.



2. Doc-To-Help prompts you to view the target. Click **Yes**.



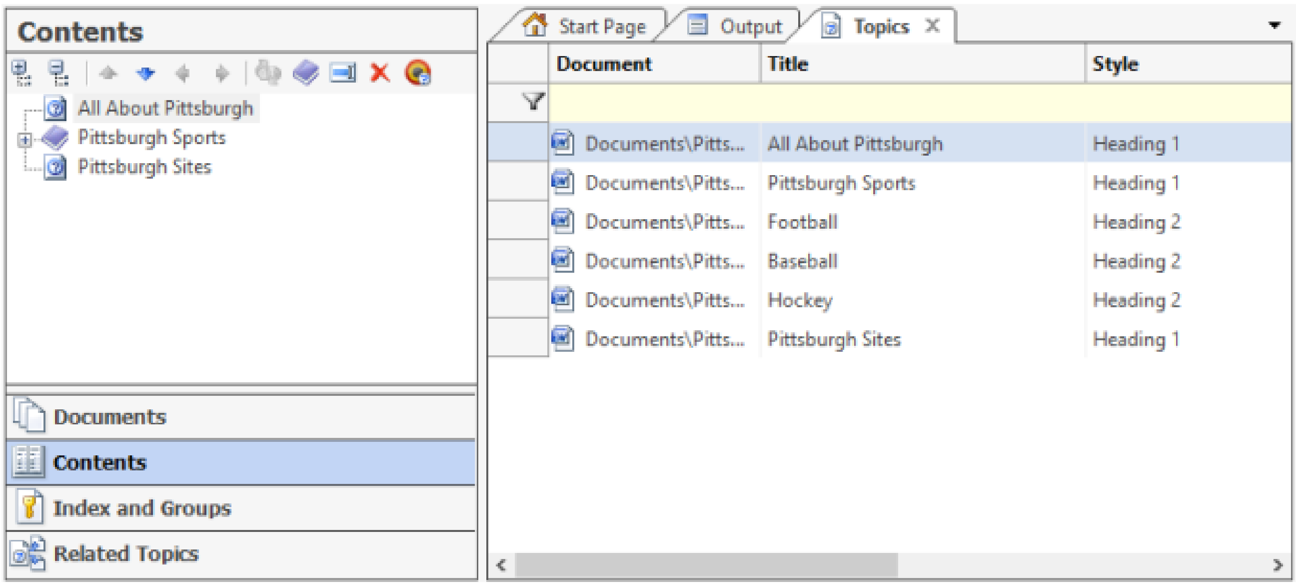
Warning: Your browser's security settings may block display of NetHelp. Click "Allow blocked content" to view your target.

Take a look at the target first. In the following example, we show Responsive NetHelp, which adjusts for any device.
The TOC 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.

3. Click the *Pittsburgh Sports* topic in the TOC of the NetHelp target. The topic has three automatic subtopic links — *Football*, *Baseball*, and *Hockey* — which were created by Doc-To-Help based on the document structure. We will learn later how to add custom subtopics, but these are created automatically.

In Doc-To-Help:

1. Click the **Contents** pane button. The TOC is displayed. In this pane, you can rearrange the TOC any way you want and add/delete topics, but the automatically generated TOC is logical and can be used as-is if you like. The TOC in the **Contents** pane looks exactly like the TOC in the NetHelp target.
2. 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 have seen our baseline, let's add more to this project.

Inserting a Graphic

We are going to add a graphic to our source document, but first we must copy an existing graphic into our project.

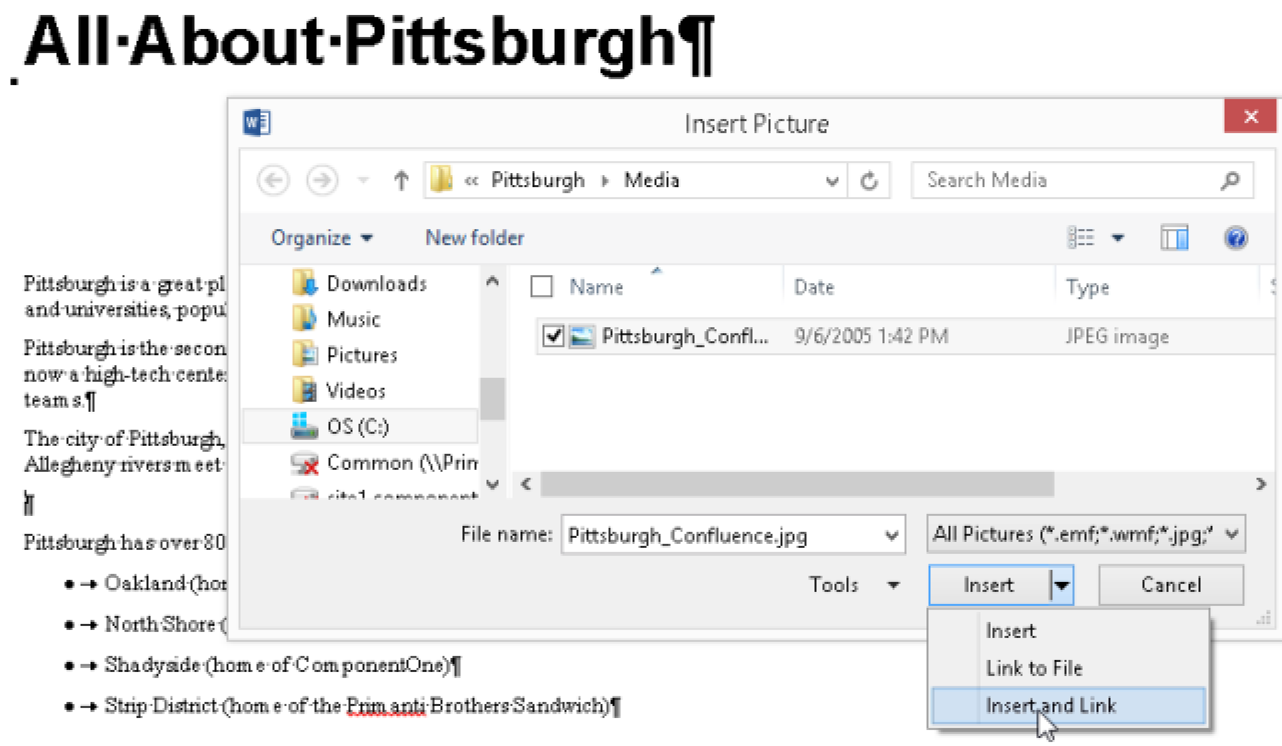
1. Using Windows Explorer, navigate to **..\Users\<username>\Documents\My Doc-To-Help Projects\Samples\All About Pittsburgh Sample Word\Media**.
2. Copy the **Pittsburgh_Confluence.jpg** file.
3. Drop it into the **..\Users\<username>\Documents\My Doc-To-Help Projects\Pittsburgh\Media** folder. Although you can use images from any folder, it is a 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® 7/8** and **Vista:** ..\Users\<username>\Documents\My Doc-To-Help Projects

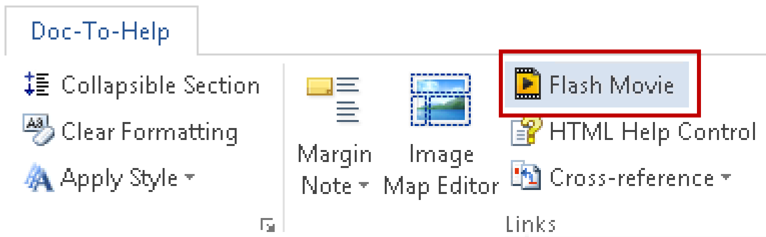
Now we are ready to insert our graphic. If it is not already open, double-click the **PittsburghQuickTour.doc** in the Doc-To-Help **Documents** pane to open it.
In Word:

1. In the **PittsburghQuickTour.doc**, add a new line after the third paragraph, which begins with “The city of Pittsburgh, Pennsylvania…”
2. On the **Insert** tab, in the **Illustrations** group, click the **Pictures** button to open the **Insert Picture** dialog box.
3. Choose **Pittsburgh_Confluence.jpg** and click the **Insert** drop-down list.
4. Choose **Insert and Link**. 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 a document using the **Flash Movie** button on the **Doc-To-Help** tab.



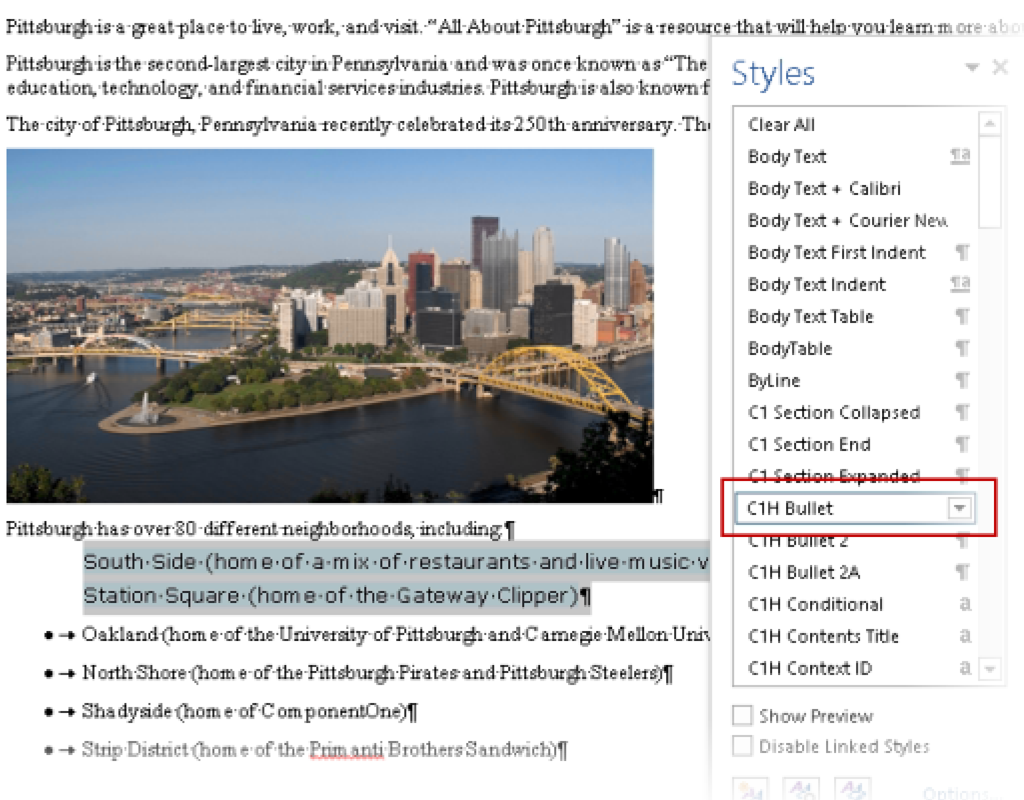
Applying a Style

Doc-To-Help uses styles to markup your source documents. In Word, these styles are stored in a template (.dot or .dotx) 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 76).

If you were using Doc-To-Help's Content 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** tabs.



Let's add some new text to the **PittsburghQuickTour.doc**. If it is not already open, double-click the **PittsburghQuickTour.doc** in the Doc-To-Help **Documents** pane to open it

In Word:

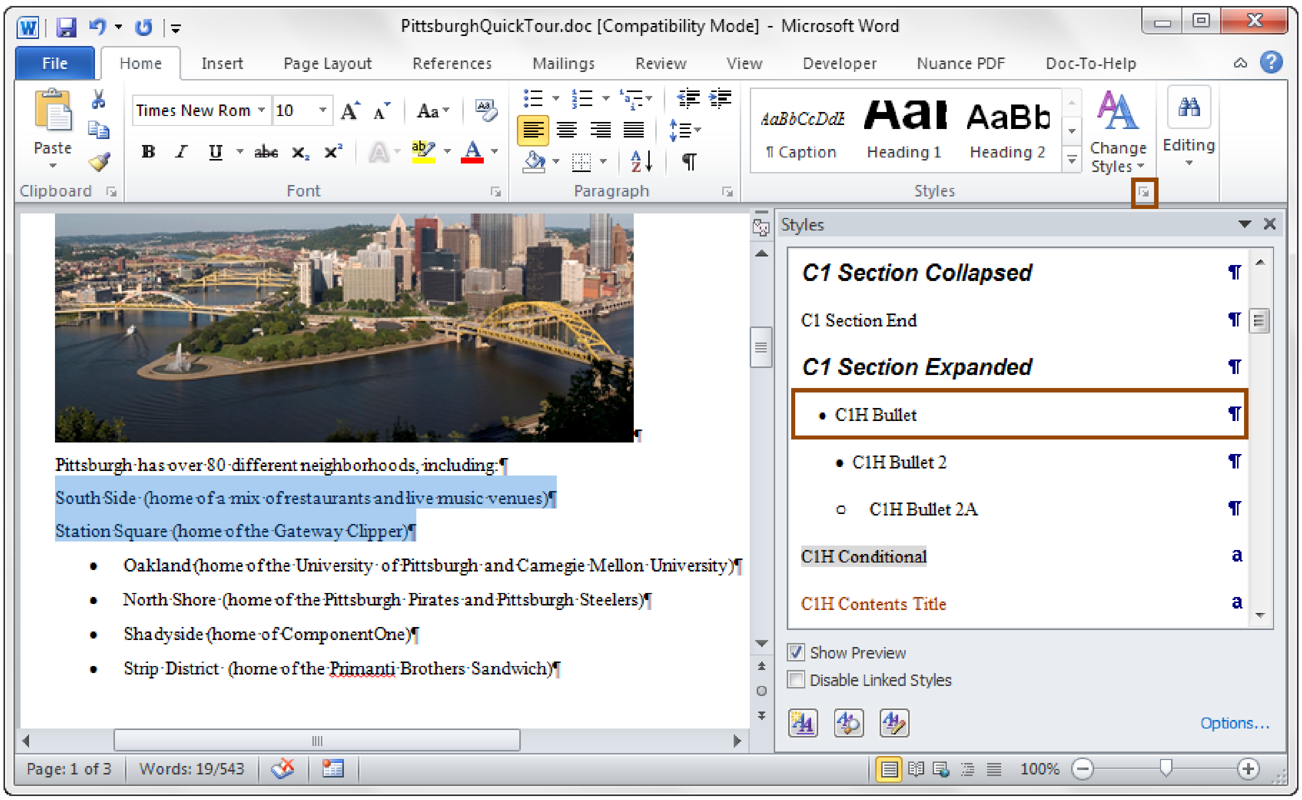
1. After "Pittsburgh has over 80 different neighborhoods," add:

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

   Station Square (home of the Gateway Clipper)

2. Select the text and click the **Styles** Dialog Box Launcher on the **Home** tab. The **Styles** dialog box opens.

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



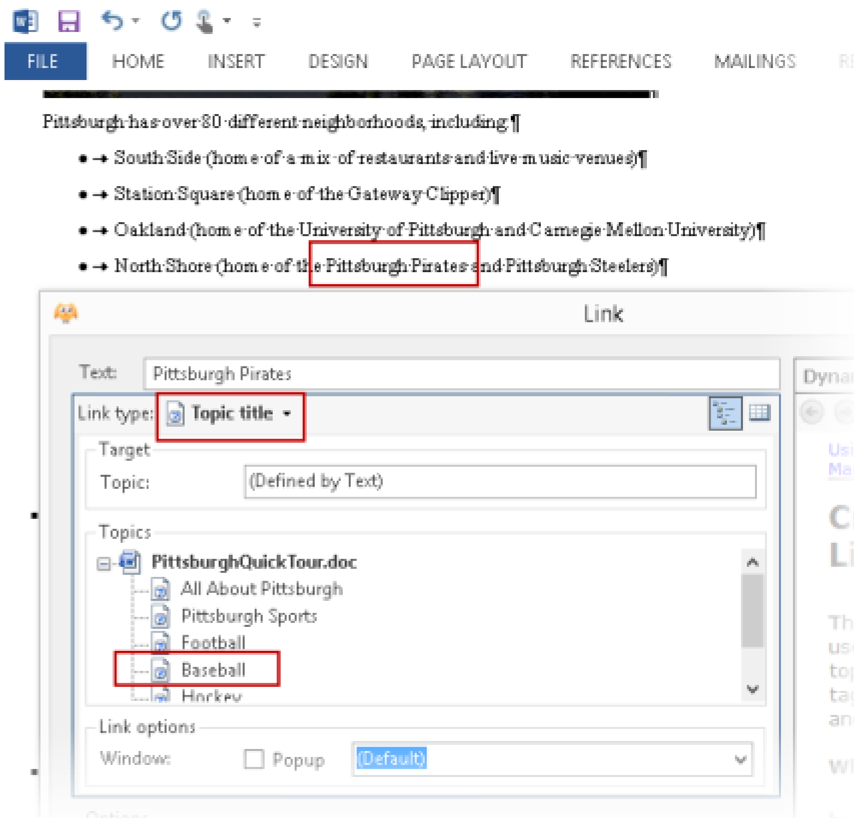
If you open the Doc-To-Help tab in Word, you will see that it includes the **Heading 1, 2, 3, 4** and **Body Text** buttons so you can quickly apply those styles.

Creating a Topic Link

You can easily create links to other topics as well as bookmarks, link tags, keywords, groups, and glossary entries.

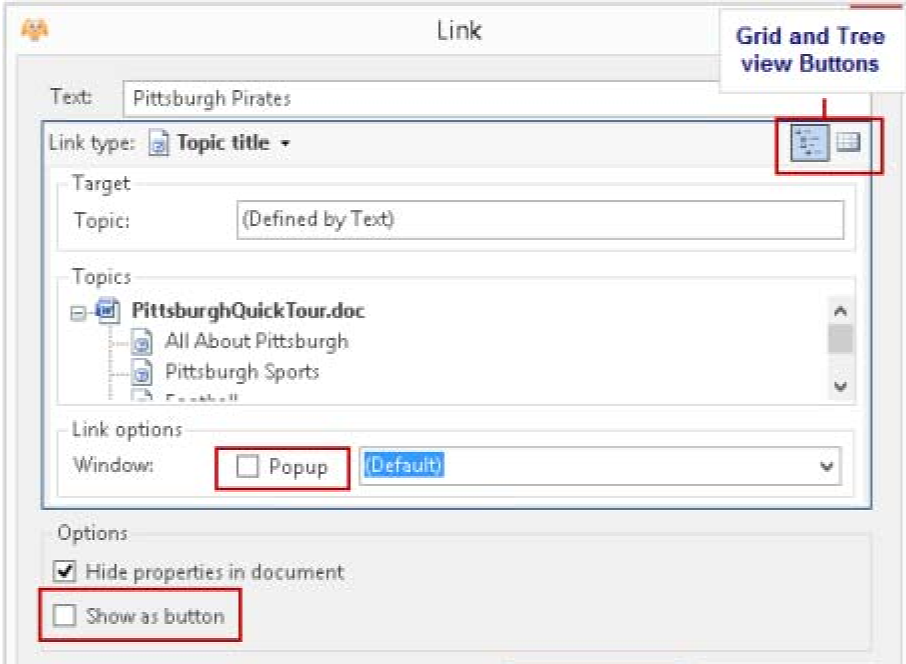
In Word:

1. In the **PittsburghQuickTour.doc**, go to the *North Shore* bullet, select the phrase *Pittsburgh Pirates* in the text, and click the **Link** button on the **Doc-To-Help** tab. The **Link** dialog box opens.



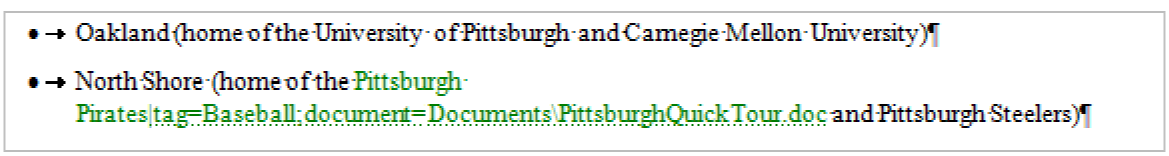
Take a quick look at the **Link** dialog box. You can also use this dialog box to create links to **Keywords** (AKA index entries), **Groups**, **Bookmarks**, **Link tags**, and **Glossary** entries. Simply click the **Link type** from the drop-down list. Notice in the **Options** area, you can select the **Show as Button** check box. This changes your link text to a button. Selecting the **Popup** check box in the **Link options** area makes the topic display in a pop-up window. By the way, linking will work in the PDF version of your manual target, as well as in your Help targets.

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



2. Make sure **Topic title** is selected for the **Link type**.
3. In the **Topics** area of the **Link** dialog box, expand **PittsburghQuickTour.doc** (if necessary) and choose **Baseball** from the list.
4. Click **OK**. This creates a topic link from the phrase *Pittsburgh Pirates* to the topic about baseball in the help target when it is built.

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*).



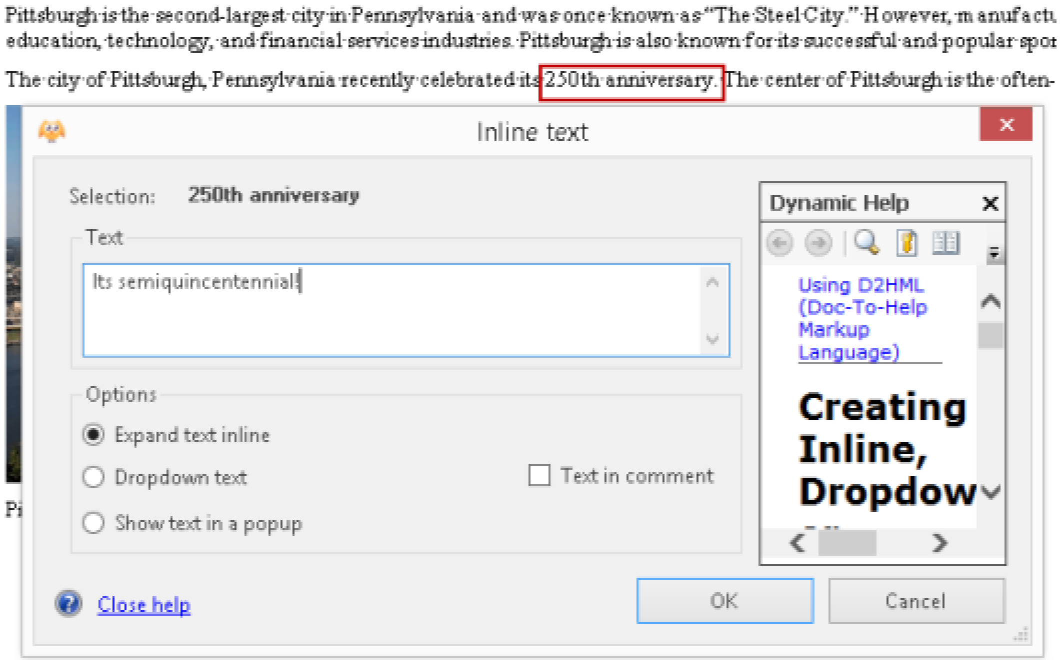
If you would like to hide hidden text, in versions of Word 2007 and later, choose the **Office button** (or **File** tab) > **Word Options**, and click **Display** on the left. Clear the **Hidden Text** check box on the right.

Creating Inline Text

Inline text is expanding, drop-down or pop-up text. It is a great way to add dynamic information to your Help project.

In Word:

1. In the **PittsburghQuickTour.doc**, select the phrase "250th Anniversary" from the first sentence and click the **Inline Text** button on the **Doc-To-Help** tab.
2. Choose the **Expand text inline** radio button, and add the text "Its semiquincentennial!" in the text box.



3. Click **OK**. Doc-To-Help informs 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 displays "Its semiquincentennial!" when clicked.
4. Save and close the document, build, and view the output if you would like.

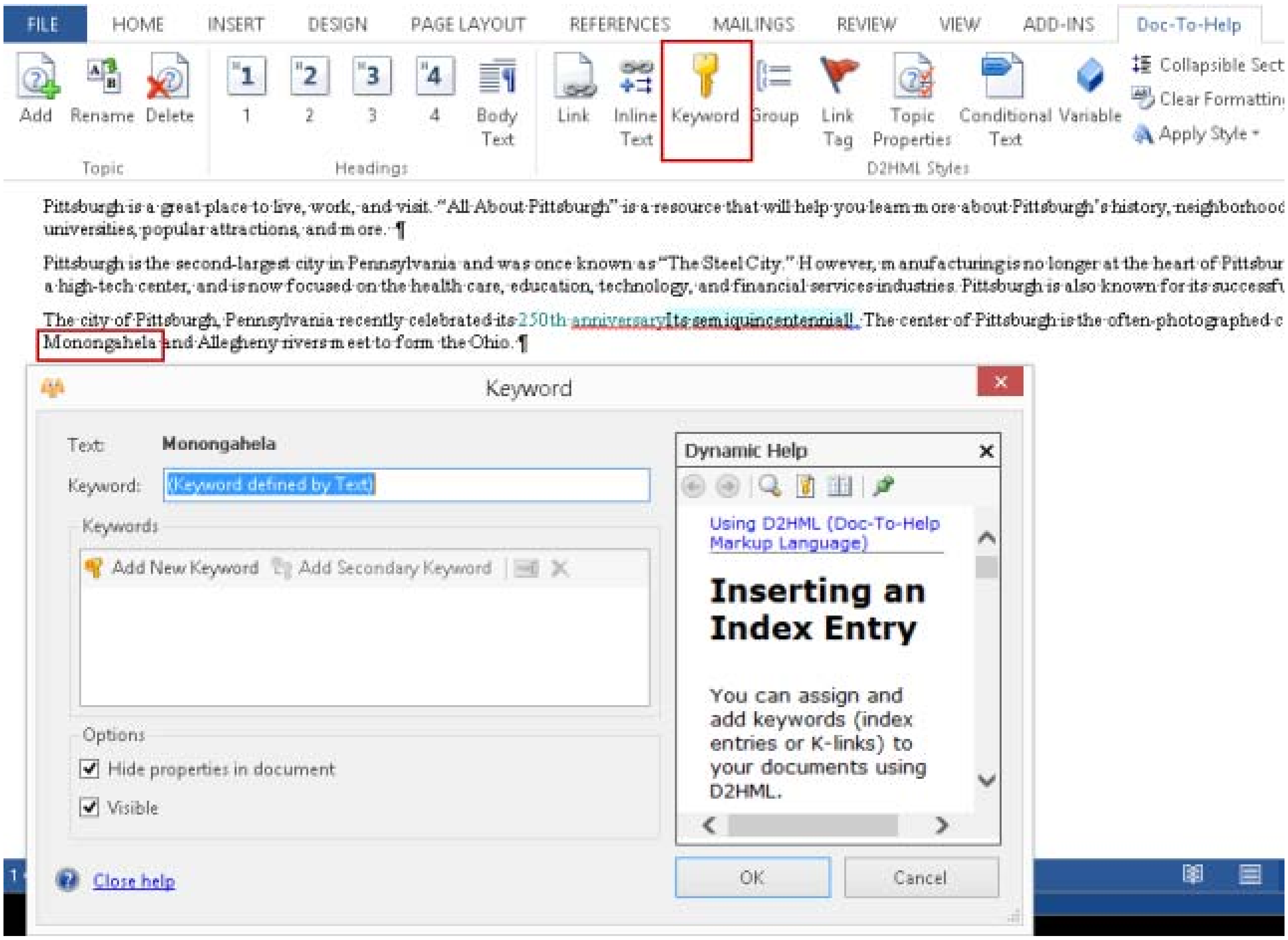
Creating an Index Entry

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.

In Word:

1. In the **PittsburghQuickTour.doc**, select the word "Monongahela" (it is in the third paragraph) and click the **Keyword** button on the **Doc-To-Help** tab.
2. Click **OK** to add "Monongahela" to your index.

   "Monongahela" appears 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.

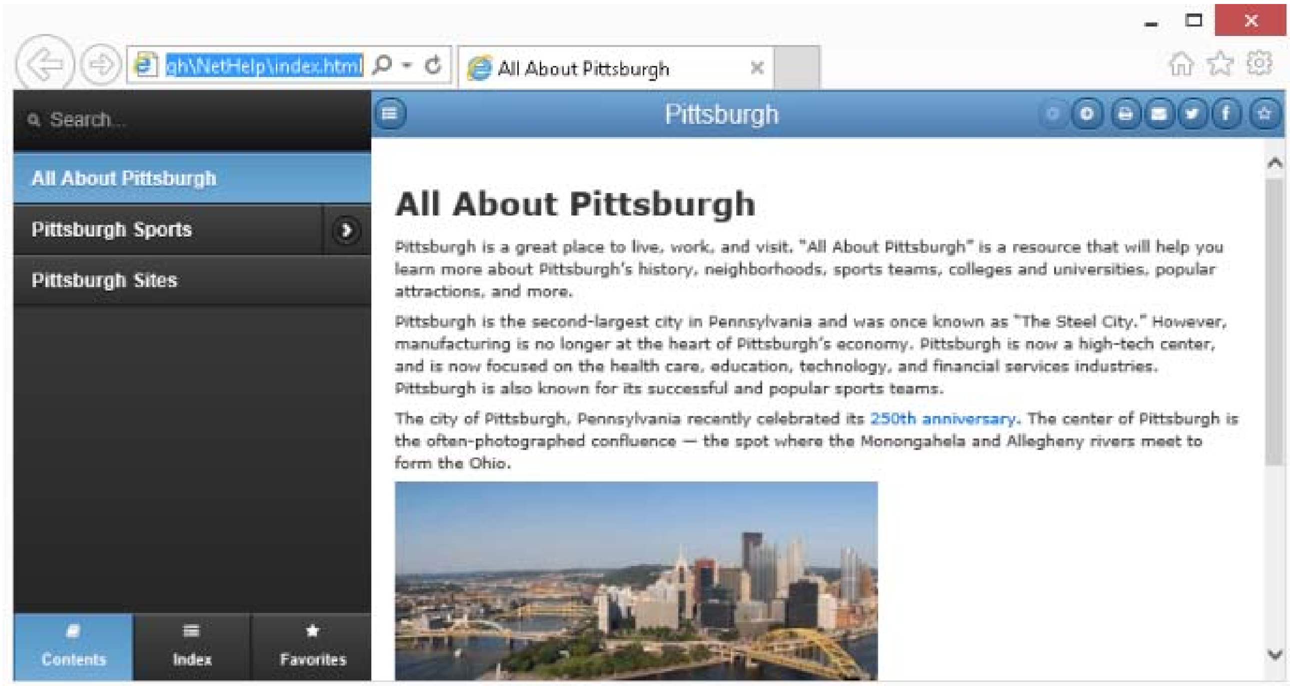


You can add additional keywords and secondary keywords from this dialog box. See Adding Items to the Index and Creating Groups on page 79.

3. Save and close the **PittsburghQuickTour.doc**.

In Doc-To-Help:

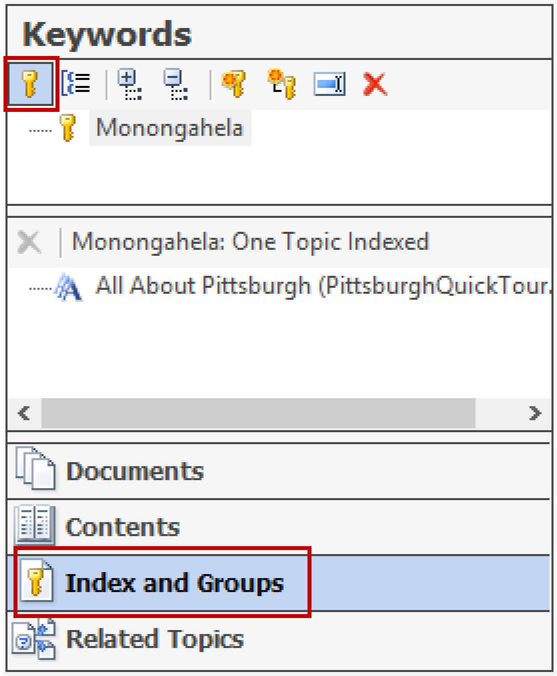
1. Click the **Build** button  on the **Home** tab.
2. After the build is complete, the target opens in your browser. See Building and Examining a Target on page 57 for more information.



3. Click "All About Pittsburgh" in the Table of Contents (TOC). You will 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 below the TOC).
   - 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 header 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.

In Doc-To-Help:
Back in the Doc-To-Help project, click **Index and Groups** in the lower-left pane. Notice that *Monongahela* now appears in the list of **Keywords**.

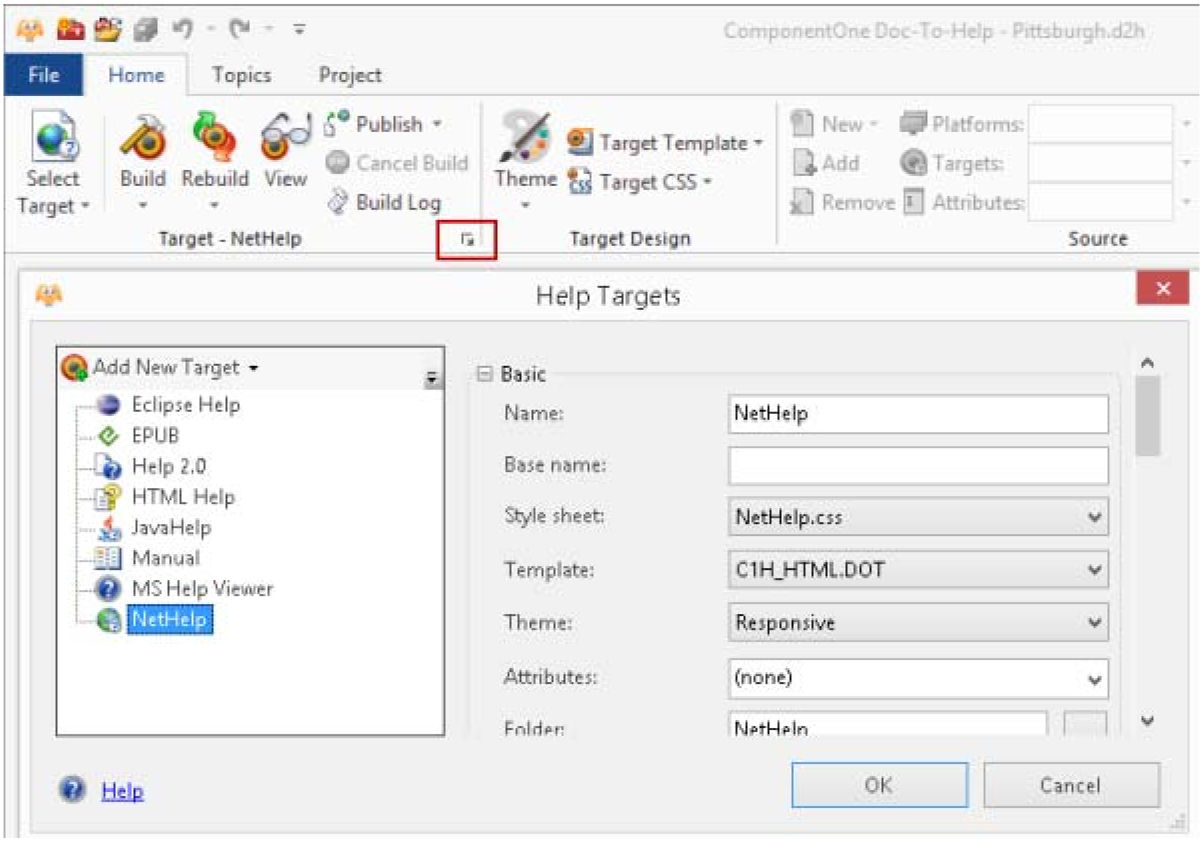


Defining Help Targets

By default, our sample has ten default Help Targets. Using the **Help Targets** dialog box in Doc-To-Help, we can rename our targets and delete the ones we do not need.

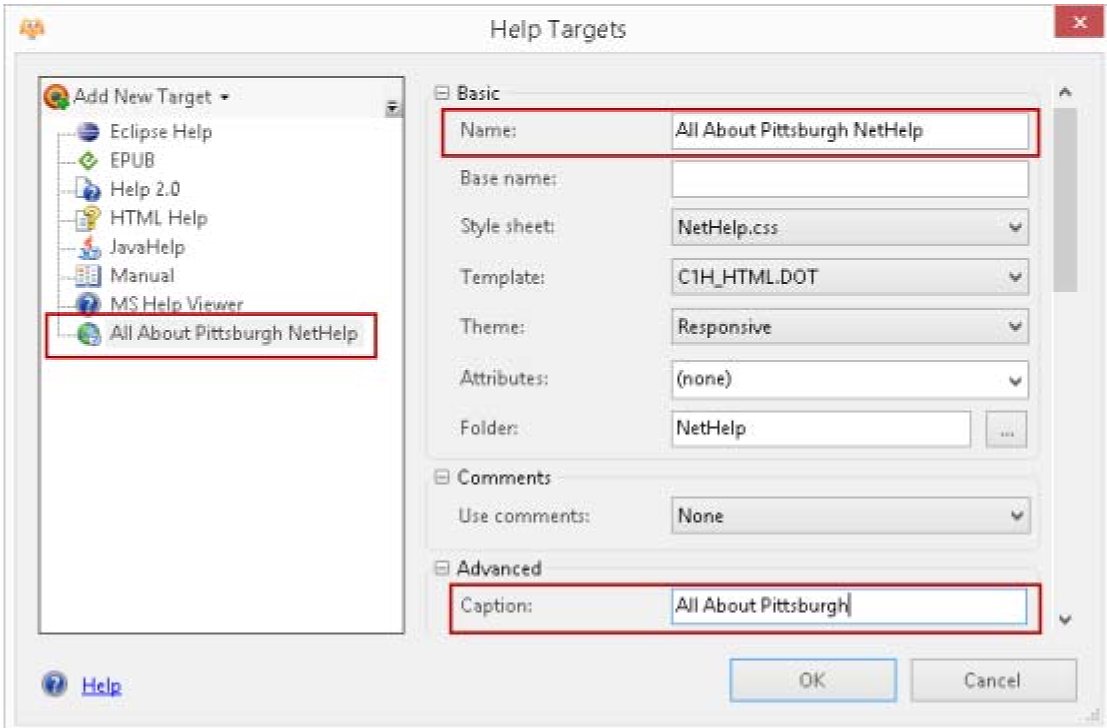
In Doc-To-Help:

1. On the **Home** tab, in the **Target** group, click the **Target** Dialog Box Launcher to open the **Help Targets** dialog box.



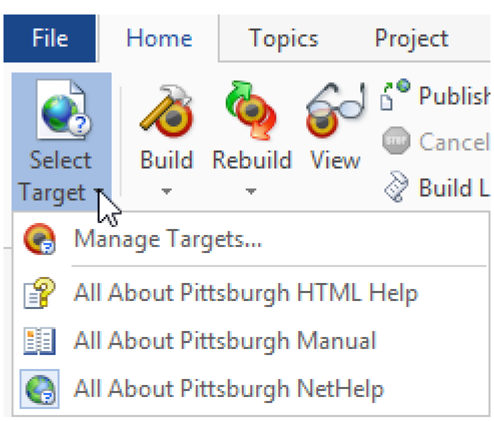
Since NetHelp is our default target, that target is selected.

2. On the right-side of the dialog box, change the **Name** field from "NetHelp" to "All About Pittsburgh NetHelp."
3. Click anywhere on the left side of the dialog box and the name of the target changes to match the name you entered.
4. Enter "All About Pittsburgh" in the **Caption** field. This changes the name displayed in the browser from "Pittsburgh" to the name we prefer.



5. Click the **HTML Help** (compiled HTML Help) **Target** in the list of targets on the left side of the dialog box.
6. Rename it to "All About Pittsburgh" by changing the **Name** field.
7. Select the **Manual Target** and change the following fields:
    a. Set the **Name** field to "All About Pittsburgh Manual."
    b. Set the **Title** field to "All About Pittsburgh."
    c. Set the **Supertitle** field to "Visitors Guide."
    d. Set **"By" line** field to "Pittsburgh Productions."

8. Since we do not need to create JavaHelp, MS Help Viewer, EPUB, Help 2.0, or Eclipse Help for this project, select those targets in the target list on the left side of the dialog box and click the **Remove Target** button for each. You can always add them back later.
9. Click **OK** to close the dialog box.
10. Now, click the **Select Target** button on the **Home** tab. The newly named targets are displayed, instead of the defaults.

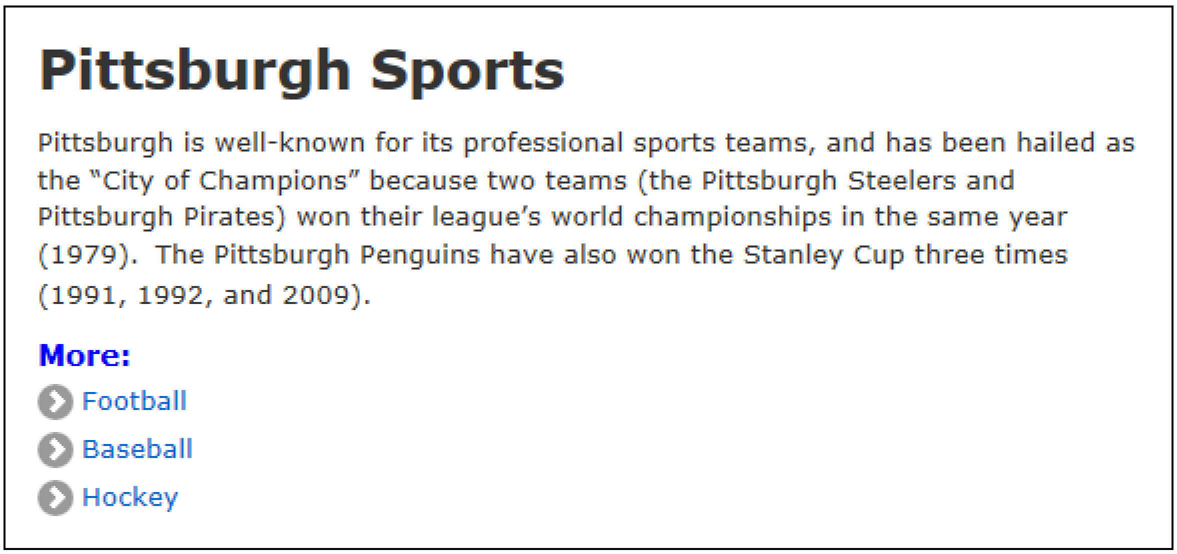


If you would like, save and close your documents. Select each target and then build and view the targets. You will see the changes we made displayed in each target.

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

Creating Custom Topic Relations

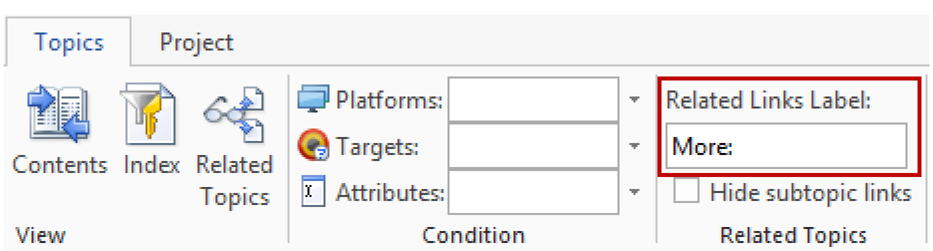
As we discussed earlier, when you open your NetHelp target and click the *Pittsburgh Sports* topic in the Table of Contents (TOC), you will notice that at the bottom of the topic, there is a section that begins with "More:", followed by links to *Football*, *Baseball*, and *Hockey*. These are called **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 the **PittsburghQuickTour.doc**. If there were Heading 3's under *Football*, they would appear as **Subtopic Links** in that topic.

In Doc-To-Help:
By default, the word *More* appears before the subtopic links, but you can change that. Simply open the **Help Targets** dialog box, select your Help target, and change the text in the **Label** field.

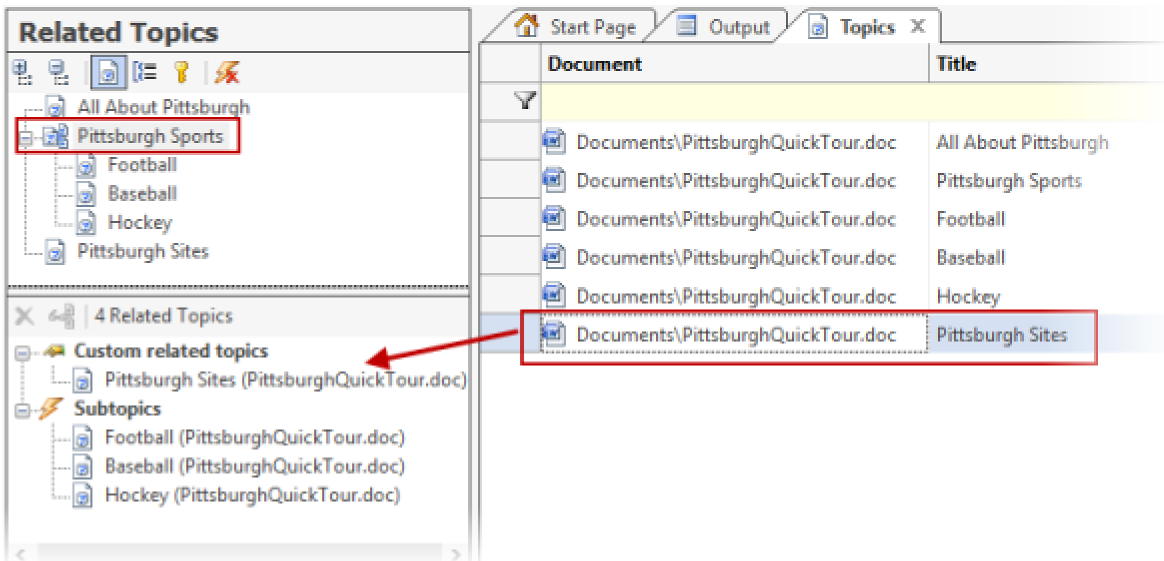
You can change the word appearing before subtopic links for individual topics by selecting a topic in the **Topics** window and changing the **Related Links Label** in the **Topics** tab of the Doc-To-Help ribbon.



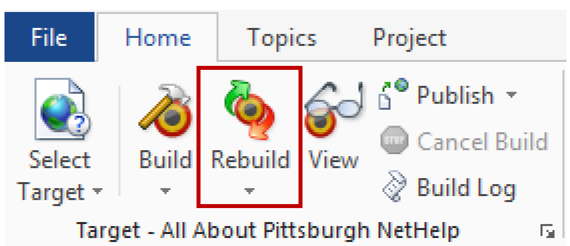
It is easy to add your own custom topic relations using the **Related Topics** pane.

1. Click **Related Topics** in the lower-left corner of the Doc-To-Help window.

2. In the **Related Topics** pane, select **Pittsburgh Sports** from the list. *Football*, *Baseball*, and *Hockey* are automatically displayed in the pane below the **Related Topics** pane.
3. 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**.



4. Save and close your documents if open, and then click the **Rebuild** button to build your target.



Look at the new list of subtopic links for **Pittsburgh Sports**. *Pittsburgh Sites* has been added to the list.

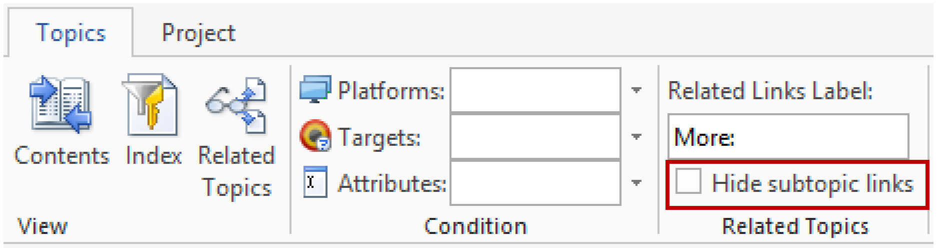
Pittsburgh Sports

Pittsburgh is well-known for its professional sports teams, and has been hailed as the "City of Champions" because two teams (the Pittsburgh Steelers and Pittsburgh Pirates) won their league's world championships in the same year (1979). The Pittsburgh Penguins have also won the Stanley Cup three times (1991, 1992, and 2009).

More:

- Pittsburgh Sites
- Football
- Baseball
- Hockey

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 **Related Topics** tab in the **Topics** tab of the Doc-To-Help ribbon.



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 "dead ends" — the user has links to other information and could be saved the trouble of using search, or checking the TOC or index.

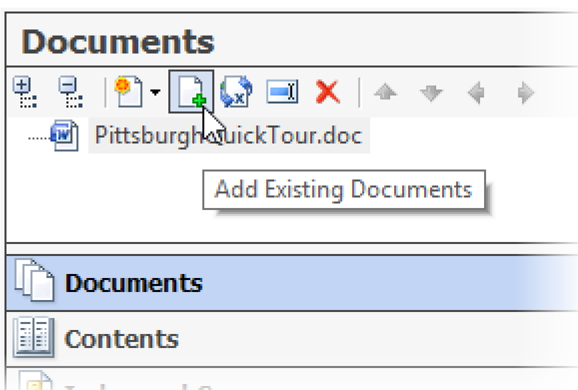
Adding a Document

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

We will take this opportunity to add a glossary to our project.

In Doc-To-Help:

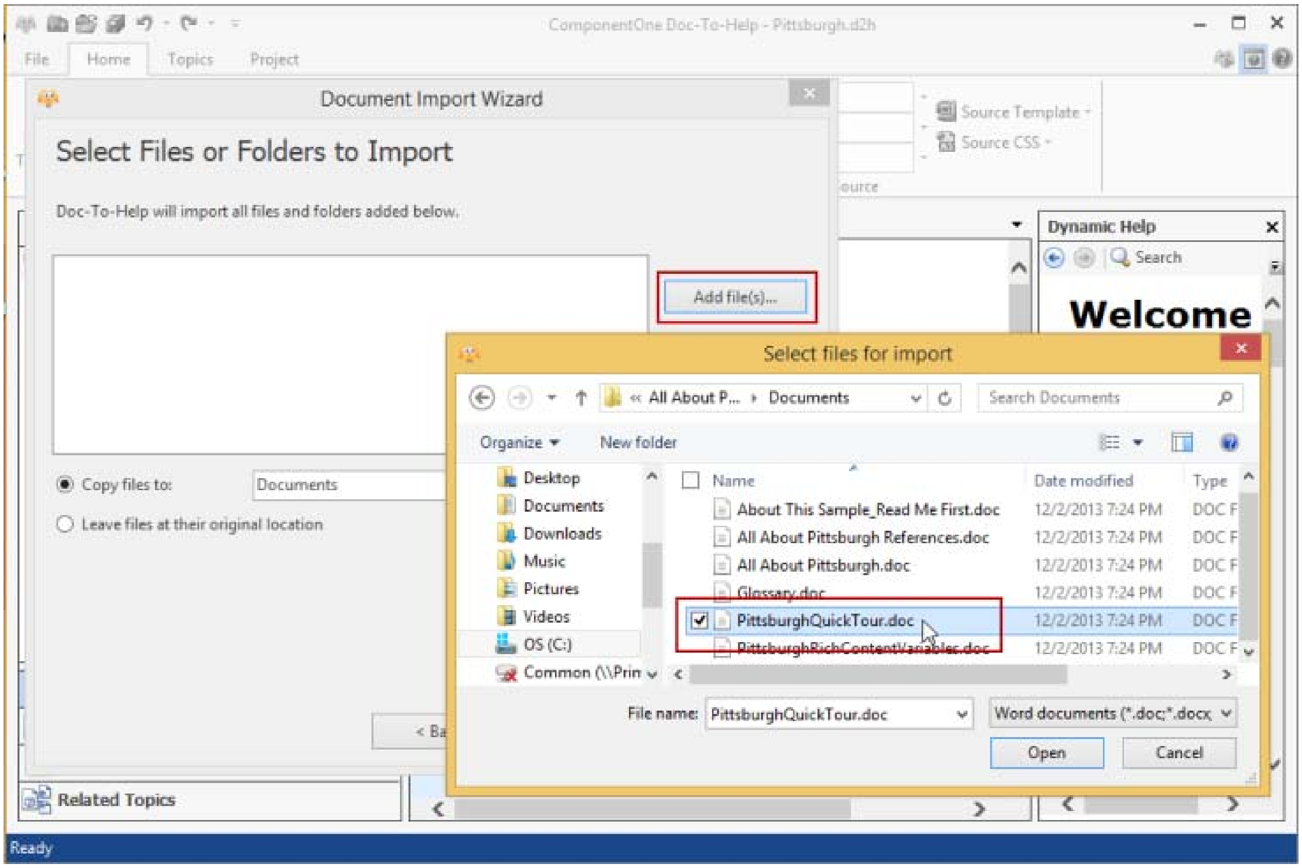
1. Click **Documents** in the lower-left corner of the Doc-To-Help window.
2. Click the **Add Existing Documents** button in the **Documents** pane.



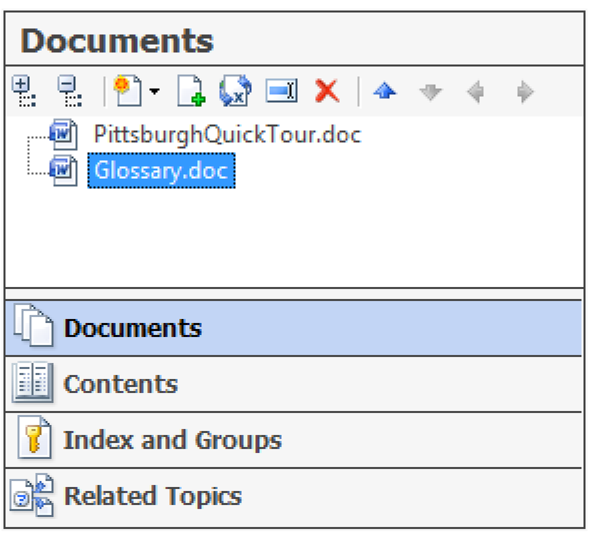
   The **Document Import Wizard** opens.

3. Follow the same steps we used when importing the **PittsburghQuickTour.doc**, but this time, choose **Glossary.doc** from the **..\Users\<username>\Documents\My Doc-To-Help Projects\Samples\All About Pittsburgh Sample Word\Documents** folder.

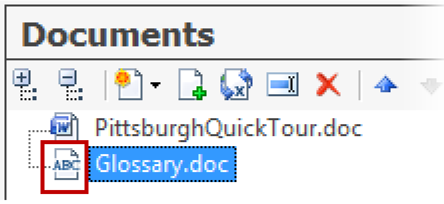
We could have imported this originally, but now you can see how easy it is to do after you have already created your project.



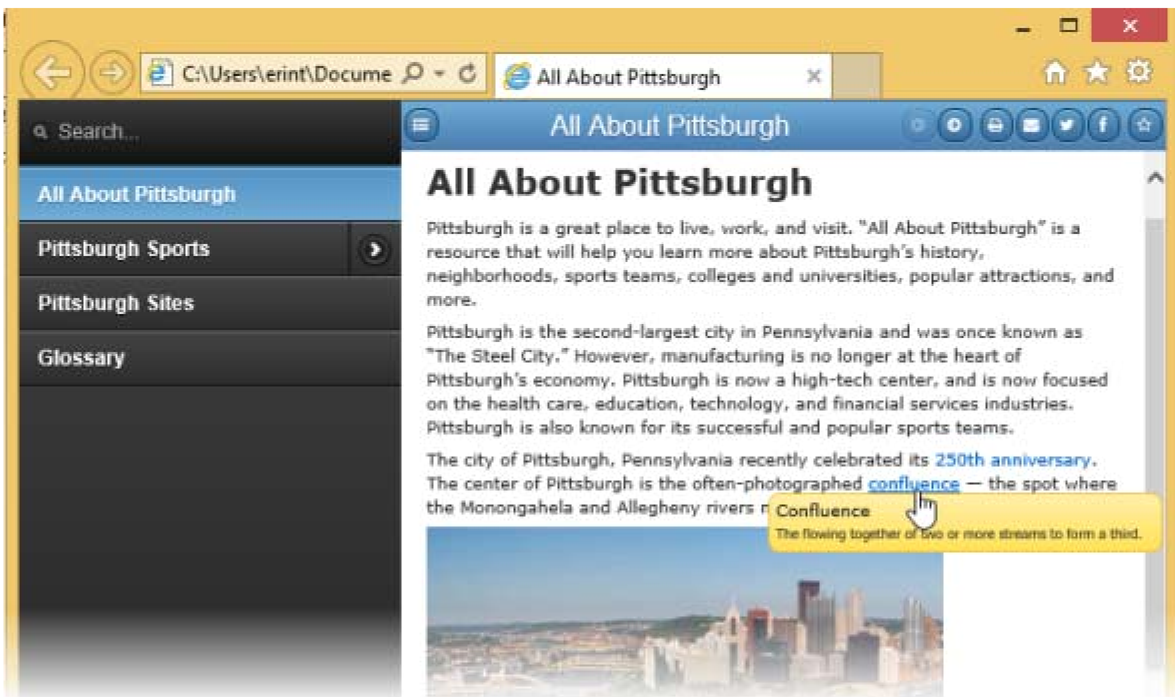
After the import is complete, the new document appears in the **Documents** pane.



4. Since a glossary is a special type of document in Doc-To-Help, right-click it and choose **Glossary** from the context menu. This document is now flagged as a glossary with a special icon.



5. Double-click the **Glossary.doc** to open it. You will see that two index entries are in this document: *Confluence* and *Primanti Brothers Sandwich*. If you put your cursor in either heading, you will notice that they are formatted with the **Glossary Heading** style. This style has a special behavior in online Help targets.
6. Close **Glossary.doc** and build your target. You will 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 pop-up window.



This pop-up window is a Doc-To-Help feature. Doc-To-Help automatically creates links from topic text to *identical* glossary entries, but only for the first instance of the word 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** Link Type 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 do not need to worry about alphabetizing them because Doc-To-Help will do it for you.

In Doc-To-Help:

1. Click **Documents** in the lower-left corner of the Doc-To-Help window.
2. Double-click **Glossary.doc** in the **Documents** pane to open it. This document 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 is easy to add more entries and alphabetize the list.

In Word:

1. After the first two entries in the **Glossary.doc**, 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.

2. Select “Ohio River” and set the style to **Glossary Heading**.
3. Select the definition, “A river that begins in Pittsburgh and joins the mighty Mississippi in Illinois.” Set the style to **C1H Popup Topic Text**.

   It should look like this:

Glossary

Confluence

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

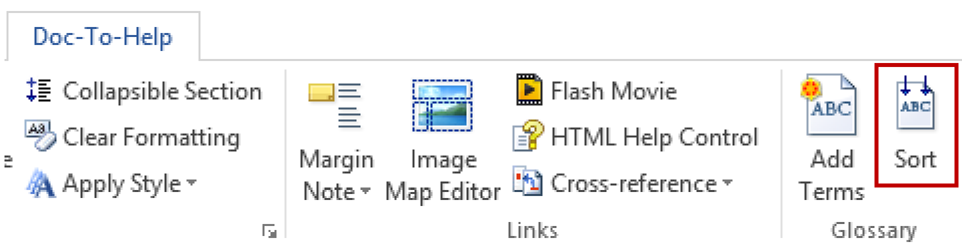
Primanti Brothers Sandwich

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

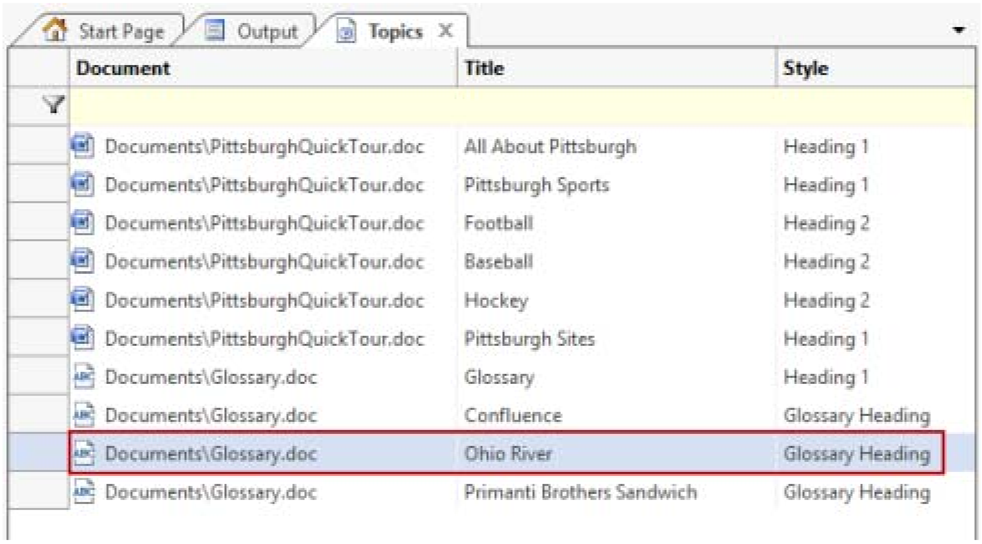
A river that begins in Pittsburgh and joins the mighty Mississippi in Illinois.

4. Ohio River is in the glossary, but now the glossary is not in alphabetical order. To fix that, click the **Sort** button in the **Glossary** group on the **Doc-To-Help** tab. The glossary is alphabetized.



5. Save and close **Glossary.doc**, and then build the target.

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



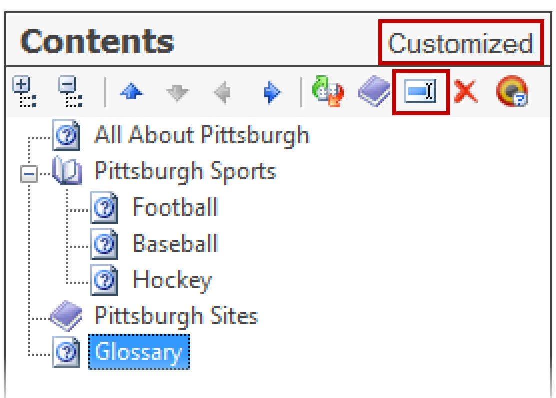
Creating a Customized Table of Contents

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

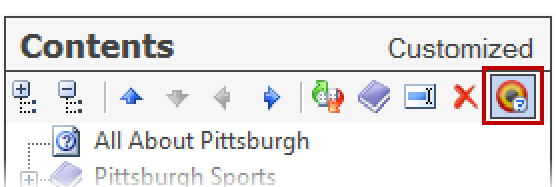
In Doc-To-Help:

1. Click **Contents** in the lower-left corner of the Doc-To-Help window to open the **Contents** pane.

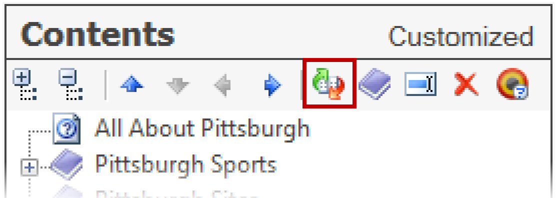
   If you select an item in the TOC and click **Remove Topic** to remove it or drag a topic from the **Topics** window into the **Contents pane** to add it, then the TOC is flagged as **Customized**. Changing the name of an item in the TOC (using the **Rename** button) also customizes it.

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

If you prefer to return to the original TOC, which is based on the document structure, click the **Rebuild Table of Contents** button.



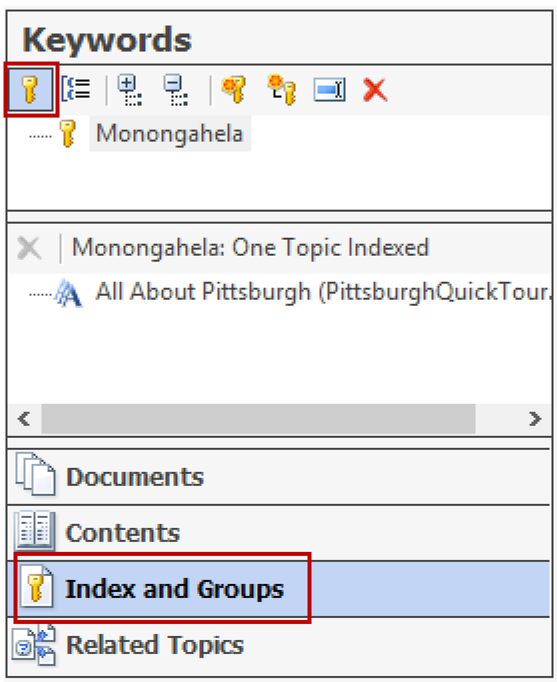
2. For this quick tour sample, click the **Rebuild Table of Contents** button.

Adding Items to the Index and Creating Groups

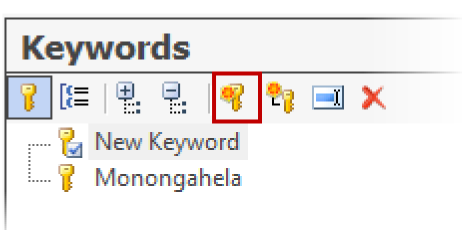
There is already one item in our index, which we added directly from one of our documents. We can also add items using the **Index and Groups** pane, along with the **Topics** window.

In Doc-To-Help:

1. Click **Index and Groups** in the lower-left corner of Doc-To-Help, and then click the **Keywords** button.

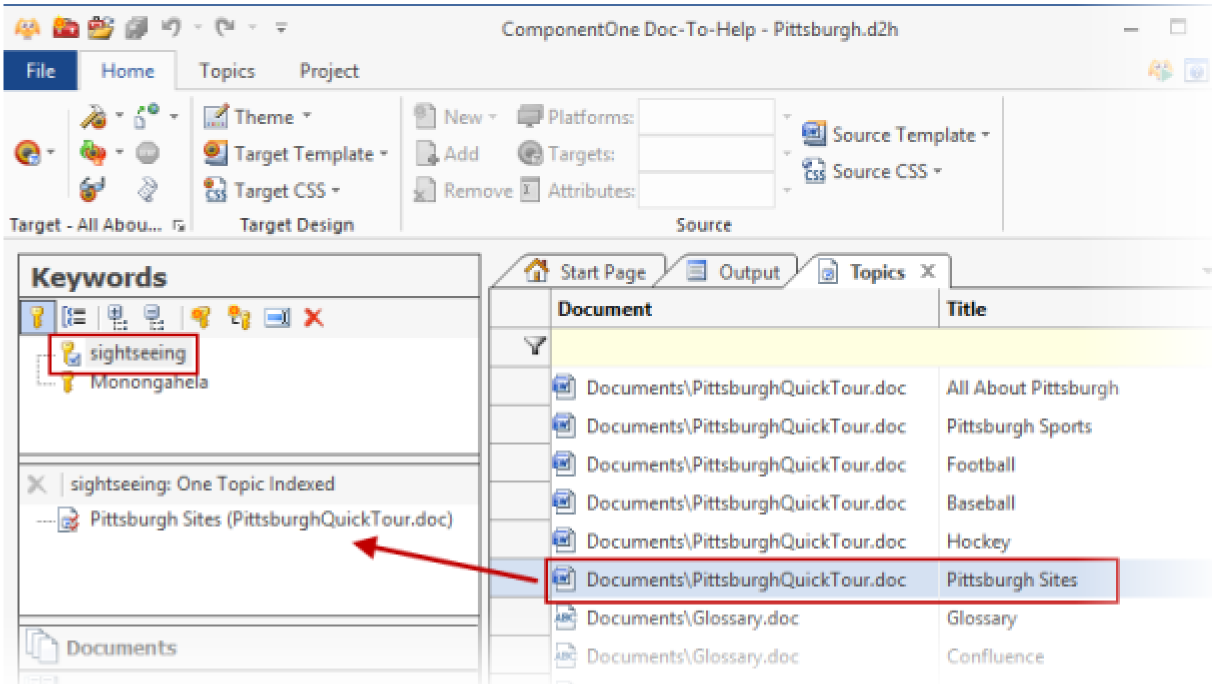


2. Click the **Add New Keyword** button.



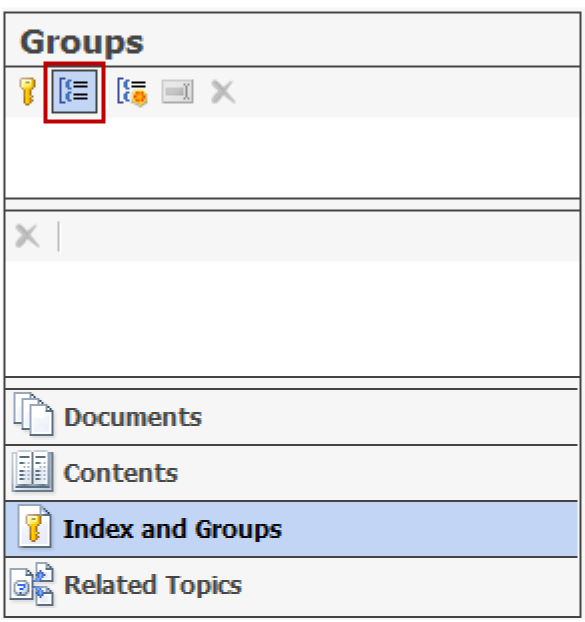
3. Name the new keyword "sightseeing."

4. 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.

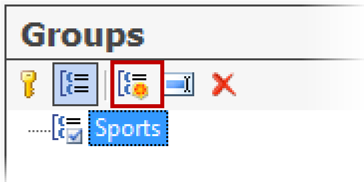


You can create topic **Groups** using this same pane. Topic groups are collections of topics you can create links to. When the user clicks the link, the topics in the group appear in a pop-up window or dialog box.

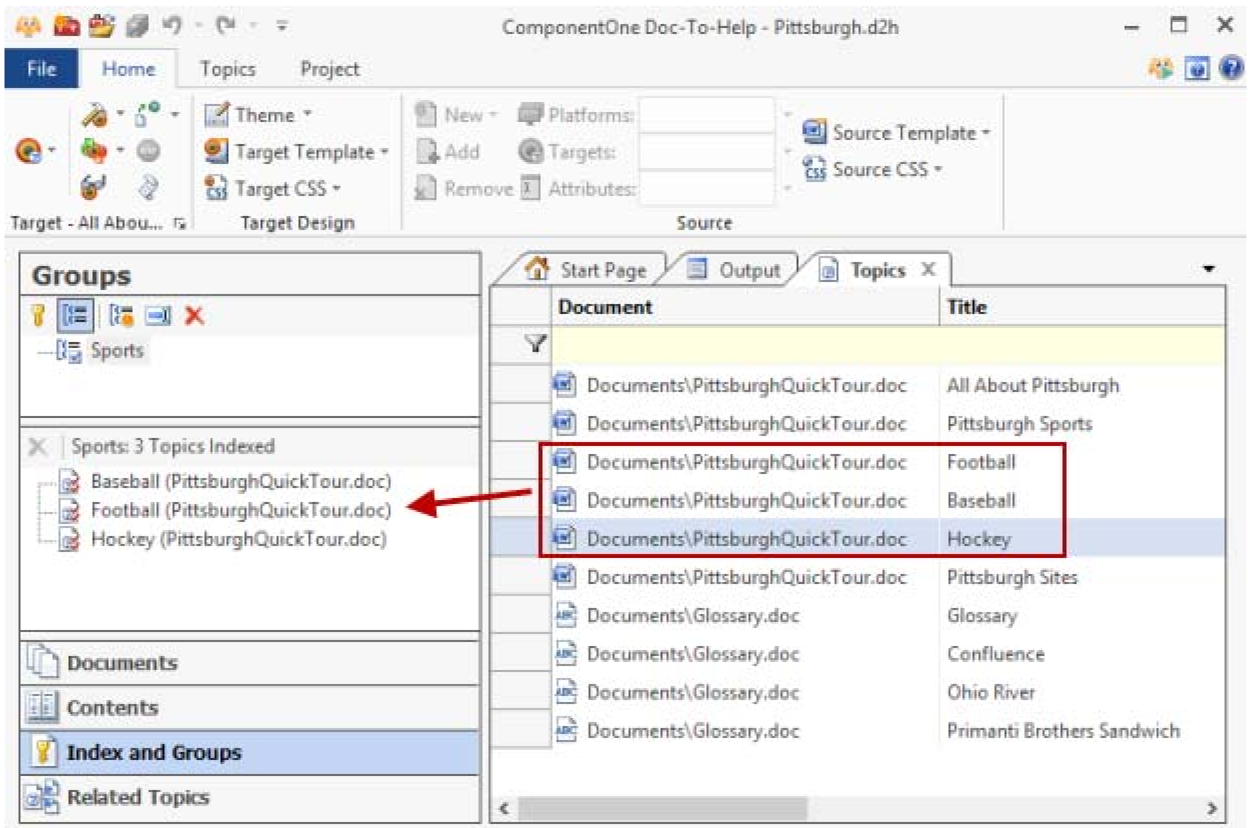
5. Click the **Groups** button in the **Index and Groups** pane.



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



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



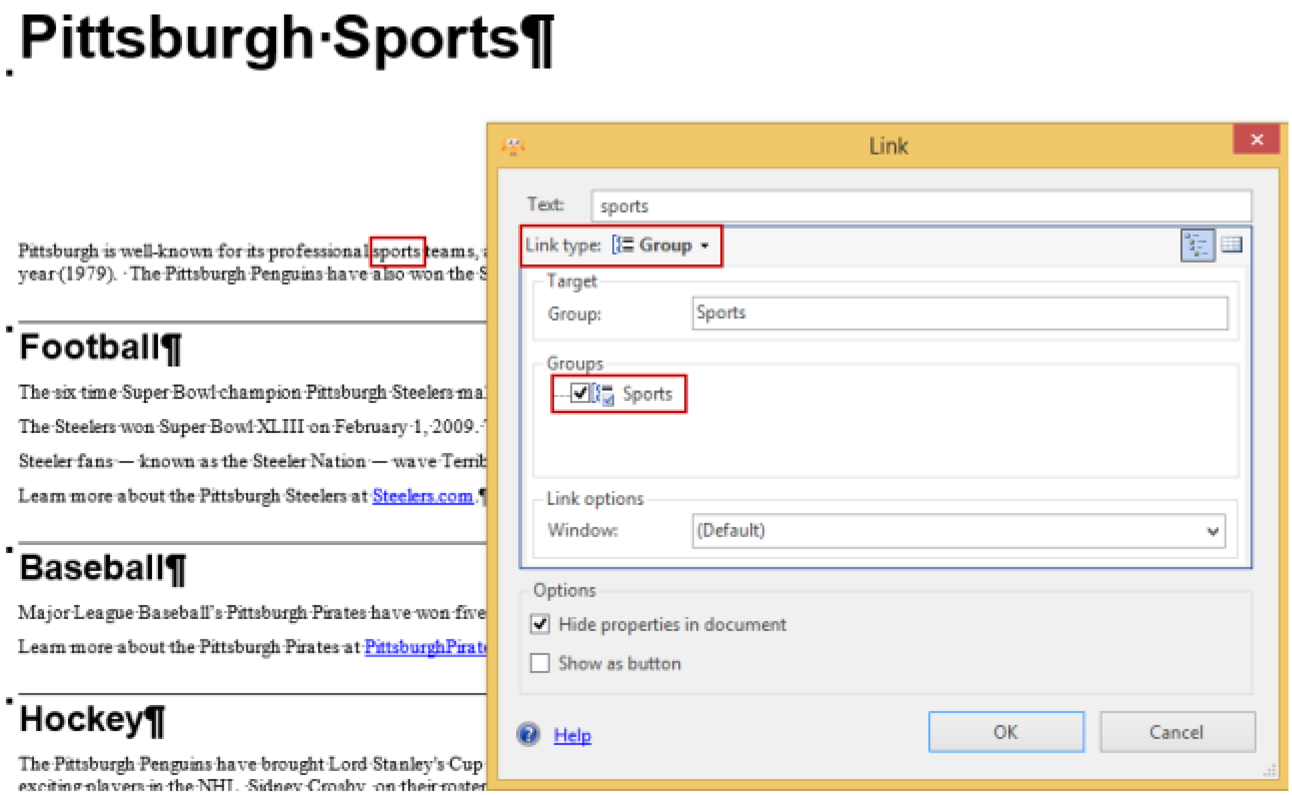
These topics are now in a group named *Sports* and you can create a link to them using the **Link** dialog box.

In Doc-To-Help:

1. Click **Documents** in the lower-left corner of Doc-To-Help to open the **Documents** pane.
2. Double-click **PittsburghQuickTour.doc** to open the document.

In Word:

1. In the *Pittsburgh Sports* topic, select the word *sports*.
2. Click the **Link** button on the **Doc-To-Help** tab and choose **Group** from the **Link type** drop-down list.
3. Select the **Sports** check box and click **OK**.



4. Save and close the document and then build your target.

View the target and notice that "sports" is a hyperlink. When you click 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 Quick Tour! Feel free to use this project to test-drive Doc-To-Help further.

Additional Resources

The schedule of free Doc-To-Help introductory webcasts can be found at: http://www.doctohelp.com/SuperPages/Webcasts/.

Information about training classes and self-paced training manuals is available here: http://www.doctohelp.com/SuperProducts/DocumentationSupport/Training+/.

Sample Doc-To-Help projects can be found in the following location. These projects demonstrate Doc-To-Help features and functionality. There are projects that feature each type of document editor, software documentation, Sandcastle reference documentation, an Employee manual, training materials, Responsive Help, and a sample in German.

- **Windows® 7/8 and Vista:** \\Users\<username>\Documents\My Doc-To-Help Projects\Samples

Also see the PDF quick reference guides available on the **Start Page** of Doc-To-Help, and the **Help** menu (available from the **File** tab.**)**

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 **File** tab > **Doc-To-Help Options** button. Click the **General** button, and 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 **File** tab > **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 133 to learn more about customizing Help Targets, Windows, and Project Properties. See Working with Source Documents on page 263, Documents pane on page 122, and Adding a Document to a Project on page 264 to learn more about documents. See Editing a CSS on page 176 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 122. Choose **Glossary** from the menu.

Watch the video: Starting a New Project in Doc-To-Help (3:48)

To open a project

1. Choose the **File** tab > **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 project created in an earlier version of Doc-To-Help, 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 **File** tab > **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 **File** tab > **Convert**. Choose the appropriate project type. A Wizard will guide you through the process.

You can convert your source documents directly to HTML5 during a legacy project conversion. (All types except Doc-To-Help 2000.) Choose the **Convert to HTML5** radio button in the **Choose Content Format** screen of the Wizard. Documents converted to HTML5 are edited in Doc-To-Help's built-in editor.

Please read the conversion notes for legacy files, which can be found at Converting RoboHelp HTML Notes on page 88, Converting RoboHelp Word Notes on page 92, Converting HTML Help Notes on page 97, Converting Doc-To-Help 2000 Notes on page 98, Converting WinHelp Notes on page 99.

To close a project

Choose the **File** tab > **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 185.

*** 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 87 for information on converting a legacy project.

You can convert your source documents directly to HTML5 during a legacy project conversion.

Files — All files in the RoboHelp project folder and its subfolders are copied to the Doc-To-Help project folder except those that are used by RoboHelp for its internal purposes. This is done to ensure that external files that may be referenced in the project source files are present in the converted folder and do not cause broken links. If you see files that you know are not needed, you can delete them manually.

Topics — Converted topic files form the Doc-To-Help Document 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 is more than one Print Documentation layout in the RoboHelp project that has 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; 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 87 for information on converting a legacy project.

You can convert your source documents directly to HTML5 during a legacy project conversion.

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 is more than one Print Documentation layout in the RoboHelp project that has 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 the HTML Help project files (.HHP, .HHC, .HHK, etc) to do so.

If you only have a .CHM file, you can still convert it, but first you must decompile it into an HTML Help project using the HTML Help Workshop. Please note that if you decompile a .chm file, you may receive the error message : “There is no TOC in the project”. If this happens, open the. HHP file and add the following line in the [OPTIONS] section: Contents file=<path-to-hhc-file>

If your HTML Help is produced by RoboHelp and you have the source RoboHelp project, do not use the generic converter, use the **RoboHelp to Doc-To-Help** converter instead. The Generic HTML Help converter does not convert RoboHelp-specific features.

See Creating and Converting Projects on page 87 for information on converting a legacy project.

You can convert your source documents directly to HTML5 during a legacy project conversion.

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 2014 by using the Doc-To-Help 2000 conversion wizard. See Creating and Converting Projects on page 87 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 2014 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 2014.

- 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://sourceforge.net/projects/helpdeco/.

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 87 for information on converting a legacy project.

You can convert your source documents directly to HTML5 during a legacy project conversion.

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).

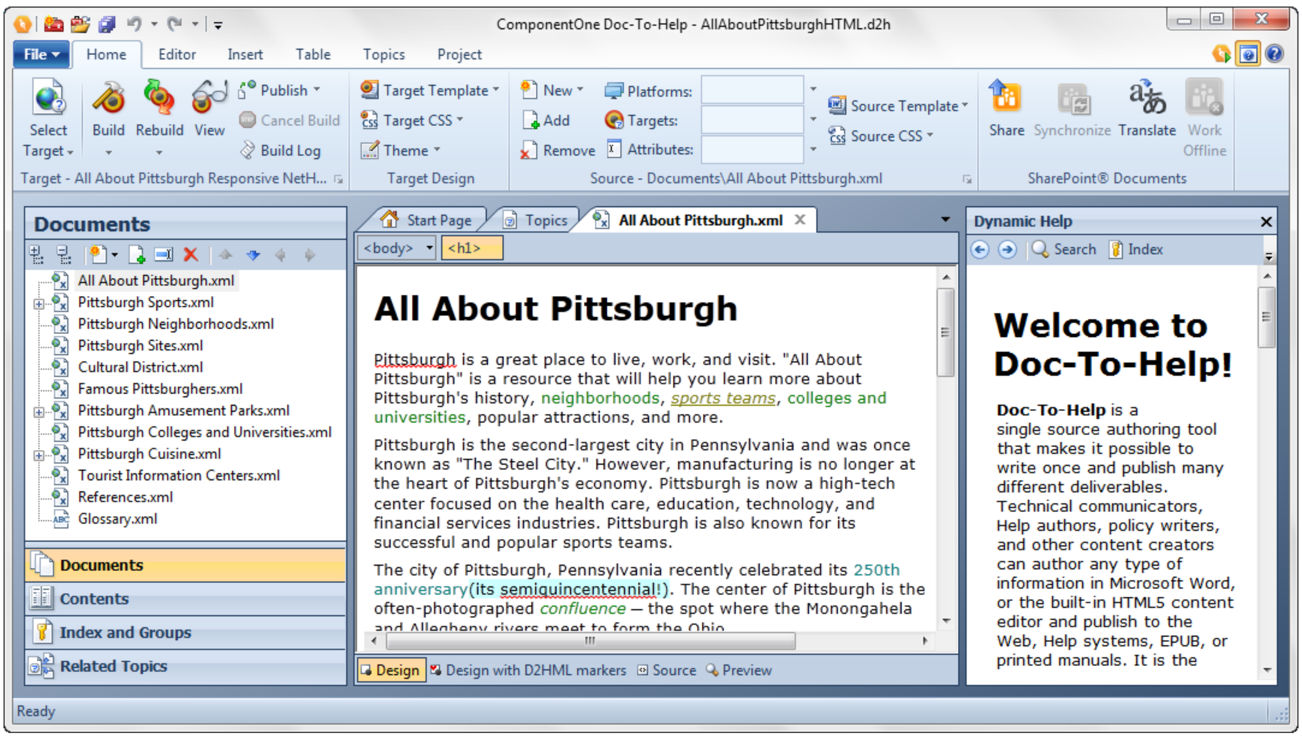
Hotspot images — Hotspot images (SHED images, segmented hypergraphics) are converted to Doc-To-Help image maps, images with hot spots. Image maps can then be explored and edited using the Doc-To-Help Image Map Editor.

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).

Navigating Doc-To-Help

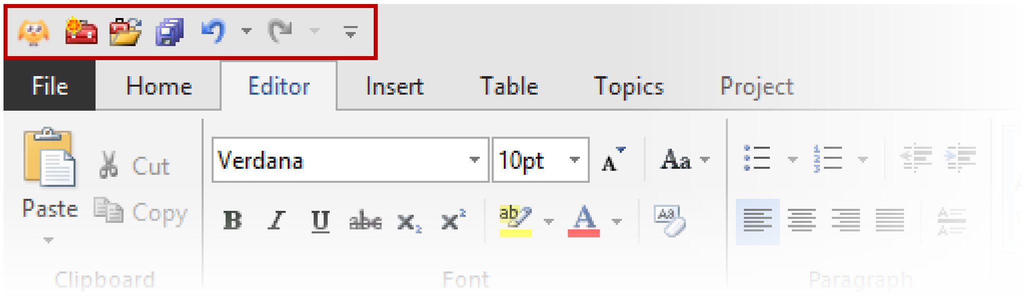


Doc-To-Help 2014’s interface incorporates Ribbons, Navigation Panes, Tabbed Windows, Super ToolTips, and Dynamic Help.

Watch the video: Tour of Doc-To-Help's Interface (5:26)

Quick Access toolbar on page 106
Ribbons on page 107
Super ToolTips on page 106
Navigation Panes on page 122
Windows on page 127
Dynamic Help on page 130

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.

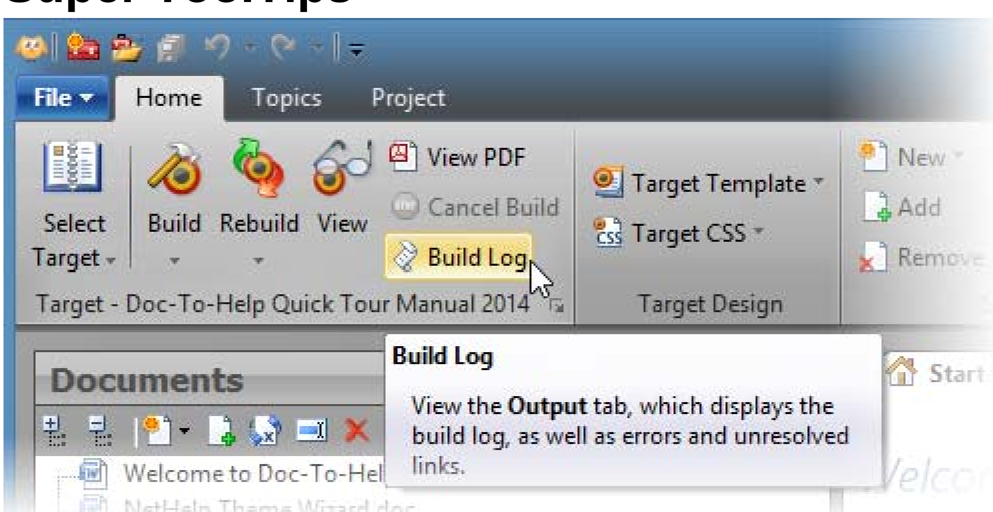
Quick Print — Print the displayed HTML5 document.

Print Preview — Preview the displayed HTML5 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 above the Ribbons on page 107, unless it is moved below them 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 2014 features Microsoft® Office 2010 style ribbons.

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

Doc-To-Help has eight ribbons: File on page 107, Home on page 108, Editor on page 110, Insert on page 115, Table on page 113, Topics on page 117, Project on page 118, and Team Authoring on page 119. The **File**, **Home**, **Topics**, and **Project** tabs are displayed by default; the **Editor**, **Insert**, and **Table tabs** appear only when an HTML5 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 106 for details.

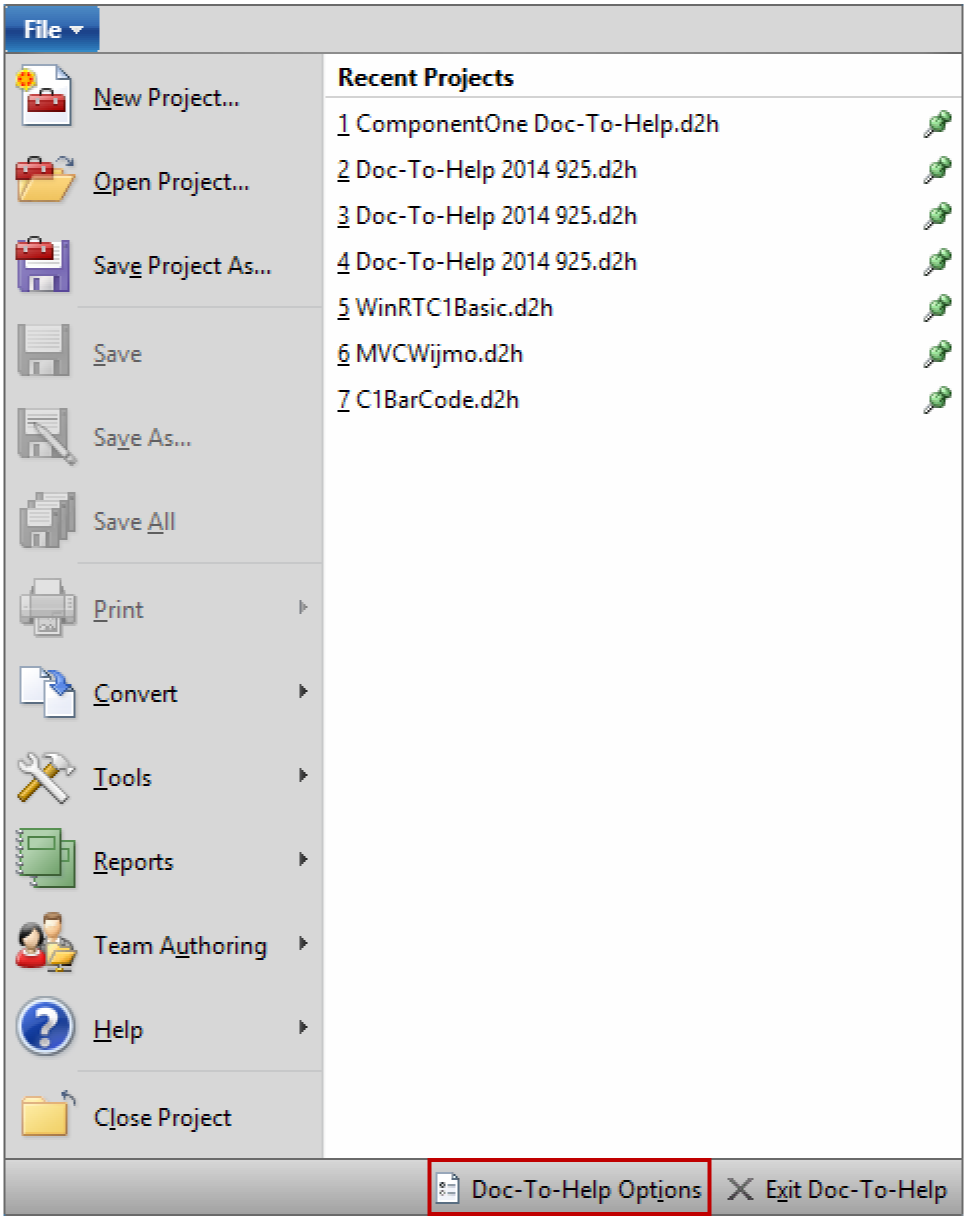
See Navigating Doc-To-Help on page 105 for information on other navigation methods.

File tab

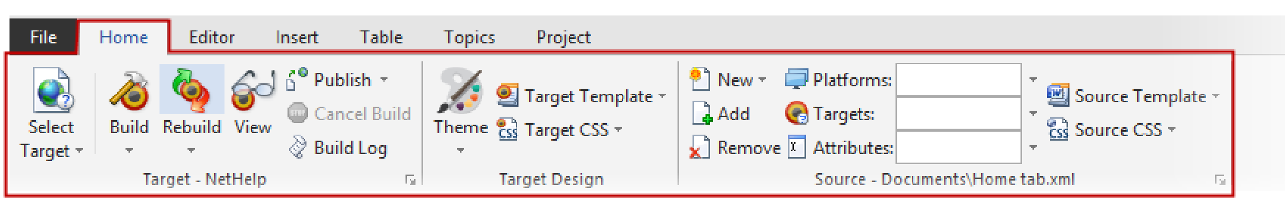
The **File tab** provides access to common functions such as opening and savings projects, printing, running reports, and setting options.

- Create or open a project on page 87
- Save a copy of your project on page 87
- Save HTML5 documents
- Print open HTML5 documents. You have the option to **Print**, **Quick Print**, or **Print Preview**.
- Convert an existing legacy project on page 87 (RoboHelp® HTML, HTML Help, RoboHelp® Word, WinHelp, and Doc-To-Help 2000)
- Schedule builds on page 342
- Run reports on page 349
- Access the **Tools** menu, where you can set options on page 34 (Startup options, Default Editors, and more), deactivate on page 34 Doc-To-Help, check for updates on page 33, compact on page 184 your project, or import/export project settings on page 185
- Create a Team Authoring project on page 381
- Access the Doc-To-Help documentation, Getting Started Wizard, 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.



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 341)
- Assigning templates and style sheets
- Setting document conditions
- Defining the look of your target(s)
- Collaborating with Microsoft® SharePoint®

See Ribbons on page 107 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 341 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 128, which displays the build log, as well as errors and unresolved links.

Publish to SharePoint — Publish the NetHelp Target to a SharePoint Document Library or SharePoint Wiki (Only visible if a NetHelp Target has been selected.) See Publishing to SharePoint on page 373.

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 133.

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 7.

Target CSS — This style-sheet controls the formatting of the Target for HTML source documents. See the Guide to Templates and Styles on page 7. Also see Editing a CSS on page 176.

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.

Source ribbon group (Home tab)

Tools in this ribbon group:

Create New Document — Add a new HTML5, HTML, or Word document to the project. See Adding a Document to a Project on page 264.

Add Existing Document — Add an existing HTML5, 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 162 (Project tab > Project ribbon group on page 119 > 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 7.

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 7. Also see Editing a CSS on page 176.

Click the dialog box launcher to open the Document Properties dialog box on page 297.

SharePoint Documents ribbon group (Home tab)

Tools in this ribbon group:

Share — Upload documents to a Microsoft® SharePoint® Library.

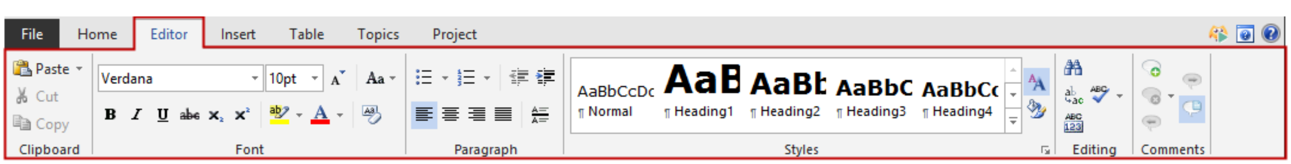
Synchronize — Synchronize shared documents that were changed either locally, or in the Microsoft SharePoint Library.

Translate — Upload documents to a Microsoft® SharePoint® Translation Library and create Doc-To-Help projects for translation to different languages.

Work Offline — In Offline Mode documents are disconnected from the SharePoint Library so Doc-To-Help will not check for updates and suggest synchronization automatically.

See Collaborating with SharePoint on page 353 for more information.

Editor tab



The **Editor tab** contains handy tools to edit and work with HTML5 documents, including:

- Cut/Copy/Paste text
- Format text

- Apply and create styles
- Find and replace text
- Provide word count statistics
- Check spelling
- Commenting

The Editor tab is only available when editing an HTML5 document within Doc-To-Help.

See Content Editor Window on page 128 for more on importing/converting documents to HTML5.

See Ribbons on page 107 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.

Font ribbon group (Editor tab)

Tools in this ribbon group:

Please note that these options will apply formatting without styles to your document. 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.

Small Font — Decrease the font size.

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

Bold — Make the selected text bold.

Italic — Italicize the selected text.

Underline — Underline the selected text.

Strikethrough — Draw a line through the middle of the selected text.

Subscript — Create small letters below the text baseline.

Superscript — Create small letters above the line of text.

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

Font color — Change the text color.

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

Paragraph ribbon group (Editor tab)

Tools in this ribbon group:

Please note that these options will apply formatting without styles to your document. 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.

Bullets — Start a bulleted list.

Numbers — Start a numbered list.

Decrease Indent — Decreases the indent level of the paragraph.

Increase Indent — Increases the indent level of the paragraph.

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.

Horizontal Line — Insert a horizontal line.

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 280 (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 176 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.

Word Count — Find out the number of words, characters, and paragraphs in the document.

Spelling — Check the spelling in the document. (You can set the language of the spell checker for each document using the drop-down; to set the default for all documents in the project click the **File** tab > **Tools** > **Options**, and click the **Spelling** button.)

Comments ribbon group (Editor tab)

Tools in this ribbon group:

Insert a Comment — Add a note about this part of the document.

Delete Comment — Deletes the comment; click for additional options, such as deleting all comments.

Previous Comment — Jump to the previous comment.

Next Comment — Jump to the next comment.

Show Comments — See all comments alongside the document.

See Adding Comments in the Content Editor on page 275 for more information.

Source Code ribbon group (Editor tab)

This ribbon group only displays if you are editing a document in Source View (See Content Editor window on page 128 for more information.)

Tools in this ribbon group:

Error List — Shows or hides the error list.

Fix Errors — Fixes all validation errors in the document.

Word Wrap — Switches the document view mode.

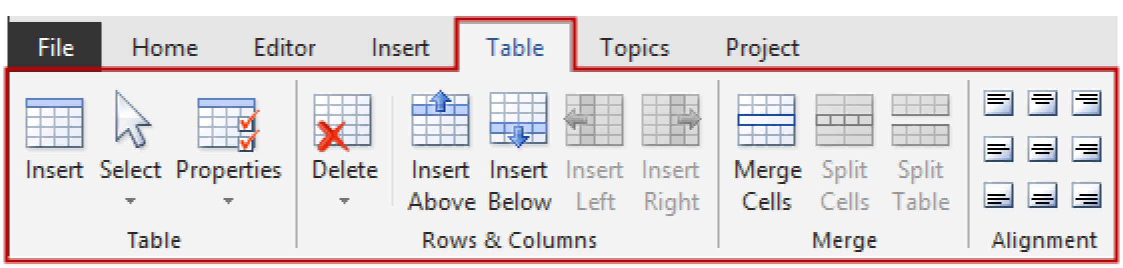
IntelliSense — Shows the IntelliSense (autocomplete) menu.

Format Selection — Formats the selected code.

Format Document — Formats the entire document.

See Working in the HTML5 Source Code View for more information.

Table tab



The **Table tab** allows you to work with tables in HTML5 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 HTML5 document within Doc-To-Help.

See Ribbons on page 107 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 Content Editor and Viewing and Modifying Table Cells, Rows, and Columns in the Content Editor 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.

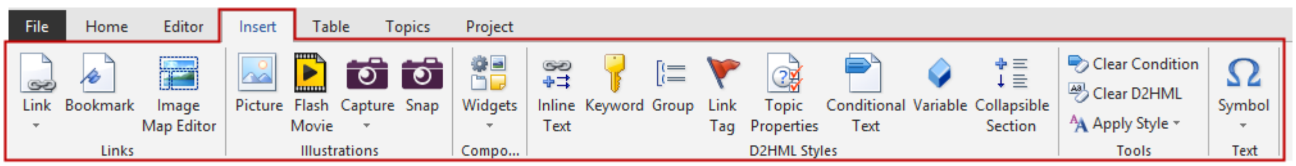
Align Center Right — Center text vertically and align it to the right side of the cell.

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 HTML5 documents, including:

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

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

The Insert tab is only available when editing an HTML5 document within Doc-To-Help. See Using D2HML (Doc-To-Help Markup Language on page 303.

For more on Styles, see Defining Character/Paragraph Styles and Topic Types on page 168.

See Ribbons on page 107 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 305 or Creating Hyperlinks in the Content Editor for more information. Use the drop-down to choose **In Project** or **External**.

Bookmark — Insert or edit a bookmark. See Creating Bookmarks in the Content Editor for more information.

Image Map Editor — Insert hot spot links within a graphics. See Creating Image Maps on page 296 for more information.

Illustrations ribbon group (Insert tab)

Tools in this ribbon group:

Picture — Insert or edit an image. See Inserting Images in the Content Editor for more information.

Flash Movie — Insert a movie in Flash format. See Inserting Flash Movies on page 295 for more information.

Components ribbon group (Insert tab)

Tools in this ribbon group:

Widgets — Use this option to insert one of seven widgets: Carousel, CodeHighlighter, Gallery, LightBox, Note, Tabs, or TopicContents. See Adding Widgets in the Content Editor on page 268 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 308 for more information.

Keyword — Associate one or more index keywords to this topic. See Inserting an Index Entry on page 309 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 310 for more information.

Link Tag — Assign or edit the link tag (unique identifier) for this topic. See Adding a Link Tag on page 311 for more information.

Topic Properties — Assign the properties of this topic, including the context ID. See Setting D2HML Topic Properties on page 312 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 313 for more information.

Variable — Insert variable text or content at the selected location. See Inserting a Variable on page 315 for more information.

Collapsible Section — Mark the selected text so that it will appear in HTML Targets as a collapsible section. See Creating an Expanding/Collapsing Section on page 316 for details.

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 317 for more information.

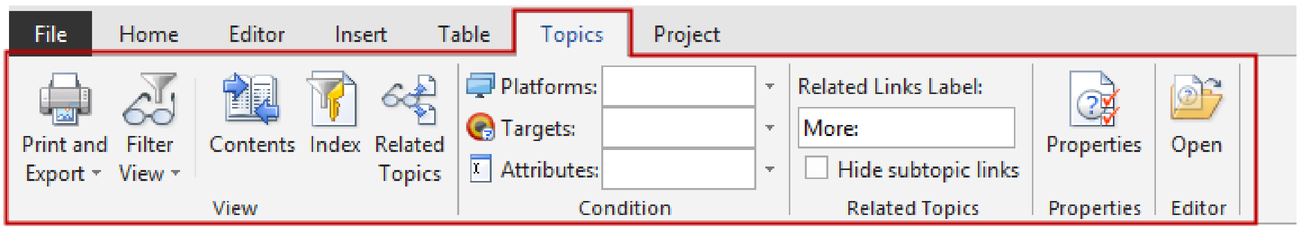
Apply Style — Format the selected text with any of the D2HML styles.

Text ribbon group (Insert tab)

Tools in this ribbon group:

Symbol — Insert symbols that are not on your keyboard, such as copyright symbols, trademark symbols, paragraph marks, and Unicode characters.

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 323 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 107 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 323 for more information.

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

Contents — Synchronize the Contents pane on page 124 with the selected topic.

Index — Display the keywords for the selected topic only.

Related Topics— Display the Related Topics pane on page 126 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 162 (Project tab > Project ribbon group on page 119 > Attributes button) to create custom attributes.

See Utilizing Conditions on page 160 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 325 for more information.

Properties ribbon group (Topics tab)

Tools in this ribbon group:

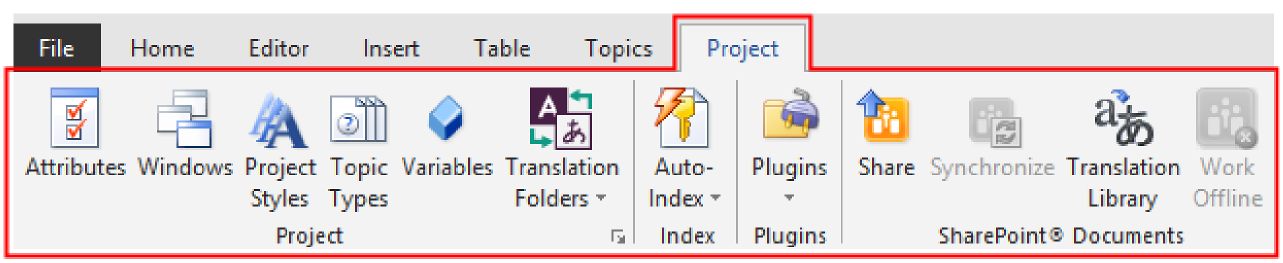
Properties — Click the **Properties** button to open the Topic Properties dialog box on page 321 for the selected topic.

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 162 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 163, where you can edit the default window types. Custom windows can also be created.

Project Styles — Opens the Project Styles dialog box on page 168, 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 129, 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 182.

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.

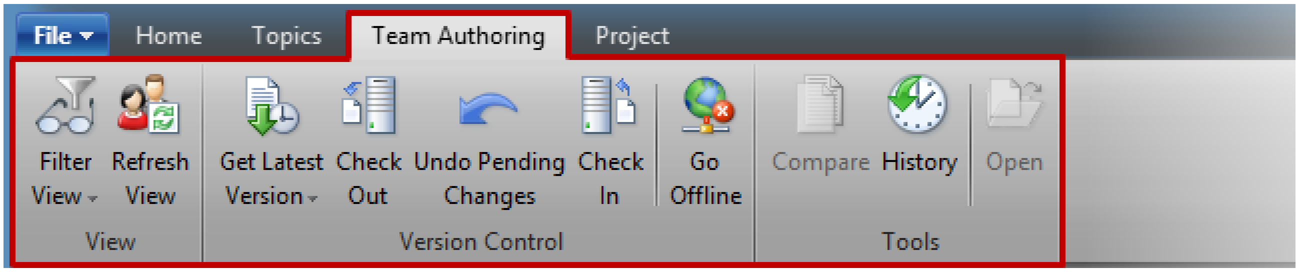
See Creating an Index or Groups on page 327 for more information.

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 409.

Team Authoring tab



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

- Enabling team support, including an interface to Microsoft® Team Foundation Server.
- Project and Document check in/check out

See Working on a Team on page 381 for more information.

See Ribbons on page 107 for more information on Doc-To-Help's ribbons.

View ribbon group (Team Authoring tab)

Filter View — Select the files that will be displayed in the Team Authoring window (options are: documents, auxiliary files, only pending changes).

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

Version Control ribbon group (Team Authoring tab)

Tools in this ribbon group:

Get Latest Version — Get the latest version of the document from the repository.

Check Out — Check document out of the repository.

Undo Pending Changes — Undo all changes to a document that has been checked out.

Check In — Check document into the repository.

Go Offline — Break connection with the server.

Tools ribbon group (Team Authoring tab)

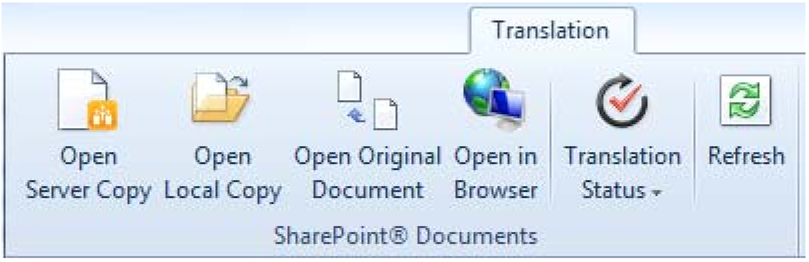
Tools in this ribbon group:

Compare (TFS projects only) **—** Compare the selected document with the version on the server.

History (TFS projects only) **—** View a history of the selected document's changes.

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

Translation tab



The **Translation tab** only appears when you open a translation project that has been uploaded to a Microsoft® SharePoint Translation Library. It allows you to:

- Open documents in the library, or the SharePoint Translation Library.
- Set the translation status of your documents.
- Refresh the status of your documents.

See Ribbons on page 107 for more information on Doc-To-Help's ribbons.

See Uploading and Working with Documents in a SharePoint Translation Library on page 360 for more information.

SharePoint Documents ribbon group (Translation tab)

Tools in this ribbon group:

Open Server Copy — Opens the server copy (the one stored on SharePoint) of the document chosen in the **Documents** pane or **Translations** window. It will open in Microsoft Word.

Open Local Copy — Opens the local copy (the one stored on your machine) of the document chosen in the **Documents** pane or **Translations** window. It will open in Microsoft Word.

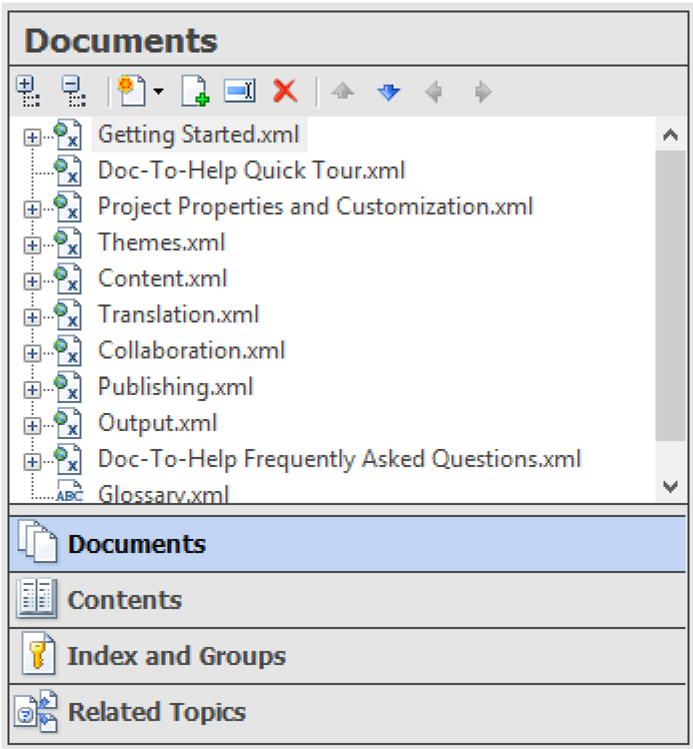
Open Original Document — Opens the original version of the document chosen in the **Documents** pane or **Translations** window.

Open in Browser — Opens the SharePoint Translation Library in a browser window.

Translation Status — Use to set the status of the project in Doc-To-Help. Options are **Not Started**, **In Progress**, and **Completed**.

Refresh — Refreshes the translation status of the documents stored in the SharePoint Translation Library. Updated statuses will display under the **Translations** tab.

Navigation Panes

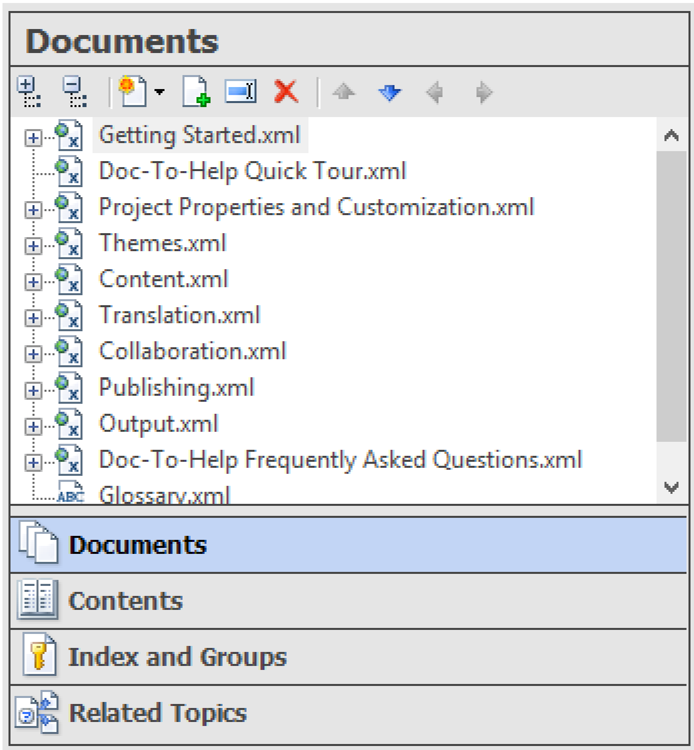


Doc-To-Help's navigation panes make it possible to quickly work with your documents, table of contents, index, and related topics. Click on the button to expand the pane and open the accompanying window on page 127.

See Navigating Doc-To-Help on page 105 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 264 for more information.)
- Convert documents to HTML5 (See Converting Existing Documents to HTML5 on page 265 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 HTML5
- 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 (HTML5) files within Doc-To-Help (in the **Content 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 34.

Using the Source ribbon group on page 109 (Home tab) you can:

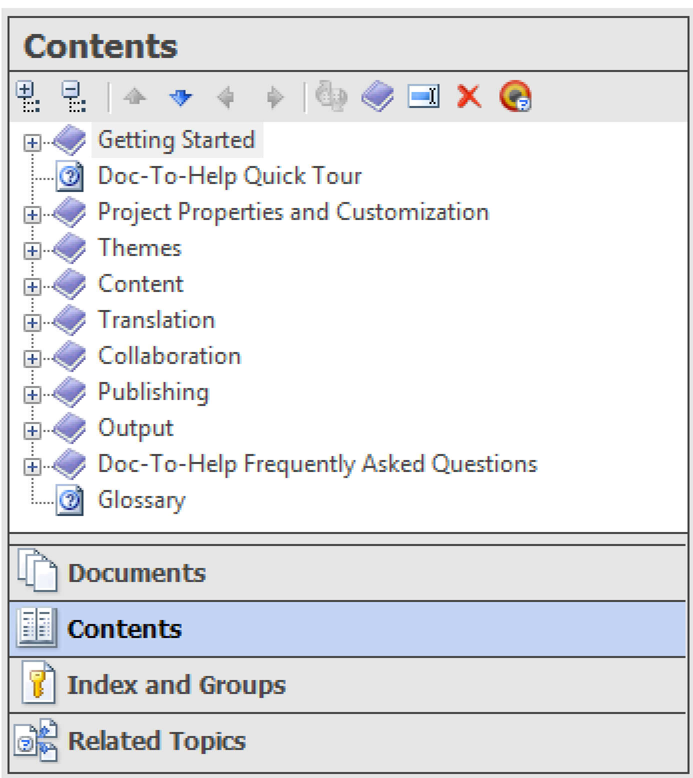
- Assign Source Templates and CSSs to your documents
- Designate Platform, Target, and Attribute-based conditions
- View Document Properties on page 297 (click the dialog box launcher)

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 109 on the **Home** tab.)

Right-click on a TOC item to:

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

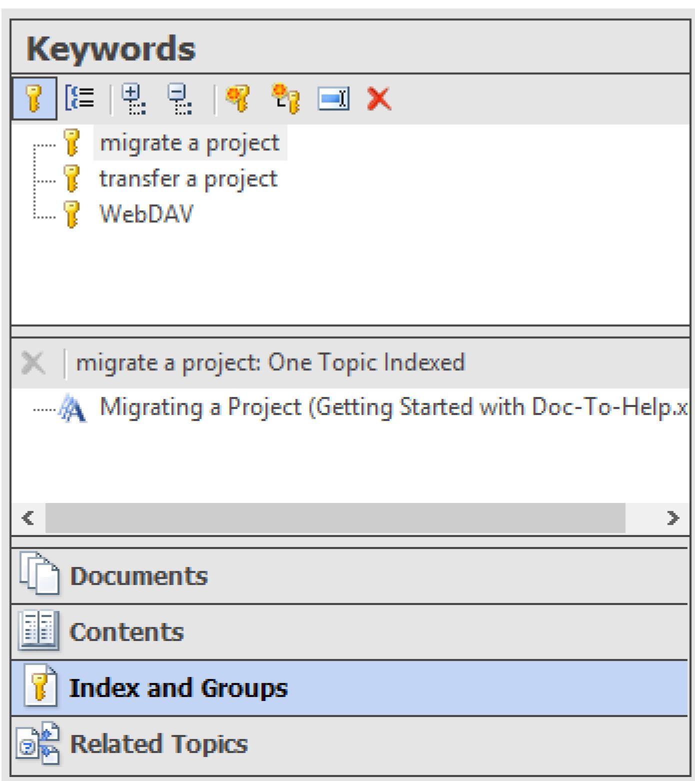
Note: You can create custom TOCs for each Help target. To do so, choose the appropriate Help Target from the **Home tab**, then click the **Target-Specific Table of Contents button** in the Contents pane. When you build the Target, a Target-specific TOC will be created. Target-specific TOCs will be flagged 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 329 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.

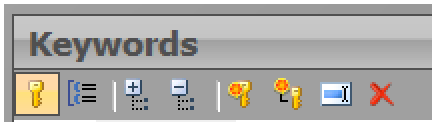


Keywords with a check box next to them have been added within the **Index and Groups pane**. Those without check boxes were added within the source documents.

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

- View/Edit your index or groups

Using the Keywords toolbar:



Expand/collapse the list

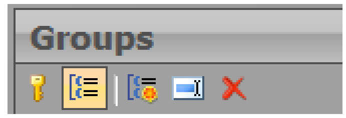
Add a new keyword

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 327 for more information about adding/linking to keywords and groups.

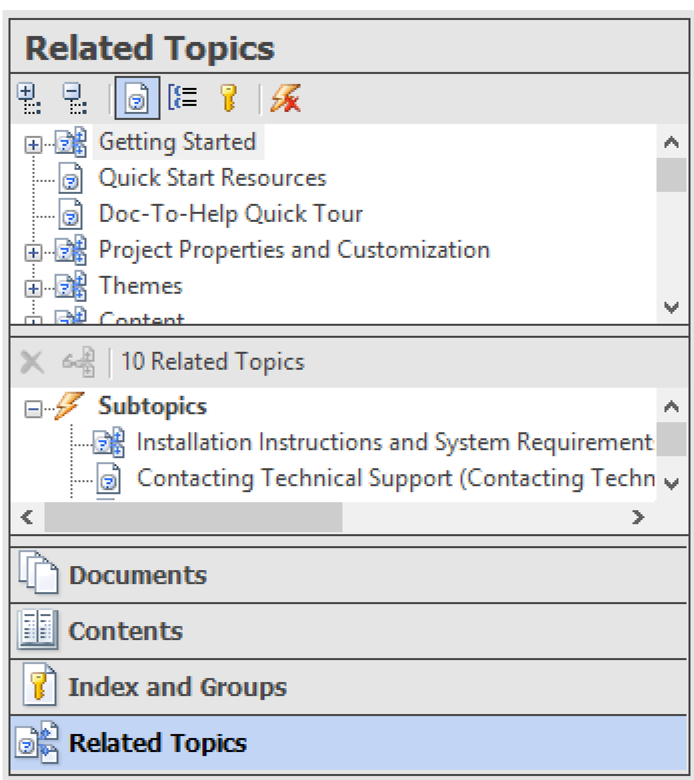
Related Topics pane

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

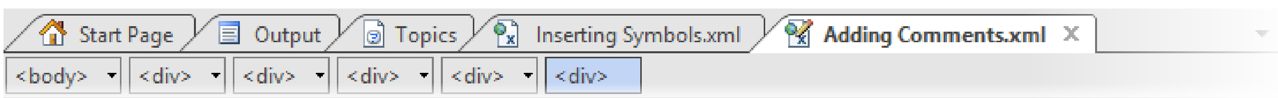
Right-click on a topic in the Hierarchy to:

- 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 321 for more information about relating topics.

Windows



Doc-To-Help has several windows that are displayed depending upon on the task. If using Doc-To-Help's built-in Content Editor, each topic will open in a different window. Navigate through them by clicking the tabs, or the drop-down on the right.

See Navigating Doc-To-Help on page 105 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 87 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 124, Index and Groups on page 125, or Related Topics on page 126 panes. You can also drag topics into source documents to create links, see Drag-and-Drop Linking on page 307 for more information.

You can also select topics and apply Platform, Target, or Attribute conditions to them. See Condition ribbon group on page 117 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)

Output window

The **Output** window will appear when a target is built. It displays the status of the build as it progresses. A progress bar and percentage indicator in the bottom right corner of Doc-To-Help will also display build progress.

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). Click the **Open Build Log** link to open a text version of the build log. 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.

Content Editor window

Doc-To-Help's Content Editor is an HTML5 editor that makes it easy to edit and validate HTML5 documents. You can work exclusively in WYSIWYG (Design mode), but if you need to, you can open Source View and edit the code as you wish. Source View features IntelliSense, line numbering, and more. The Content Editor also features visual undo and redo.

The **Content Editor** opens when you open or create an HTML5 document in your project. It cannot be opened independently. When it opens, three additional tabs will appear — Editor on page 110, Table on page 113, and Insert on page 115 — that work with the **Content Editor**. See Editing HTML5 Documents on page 266 for more information.

Comments can be added to your HTML5 documents, see Adding Comments in the Content Editor on page 275. You can leave comments for your own use, or have your entire team and/or reviewers add comments as well.

To open an HTML5 document

To open a project document in the Content Editor, open the **Documents** pane and double-click on the document name. The documents created in the Content Editor will have the .xml file extension.

To add a new HTML5 document to your project, see Adding a Document to a Project on page 264.

To convert an HTML or Microsoft Word document to HTML5, see Converting Existing Documents to HTML5 on page 265.

Content Editor modes

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

- **Design** — The default WYSIWYG editing mode. You can add comments to your document in this mode using the **Comments** ribbon group on the **Editor tab**.
- **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. (You can also add comments in this mode.)
- **Source** — Displays the HTML5 source. In this mode, the **Source Code** ribbon group will display in the **Editor tab**, which includes options to view the error list, fix validation errors, turn word wrap on/off, turn on IntelliSense, and format selected code or the entire document. See Working in the HTML5 Source Code View.
- **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 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 119, 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.

See Working in a Team Project on page 387 for more information.

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

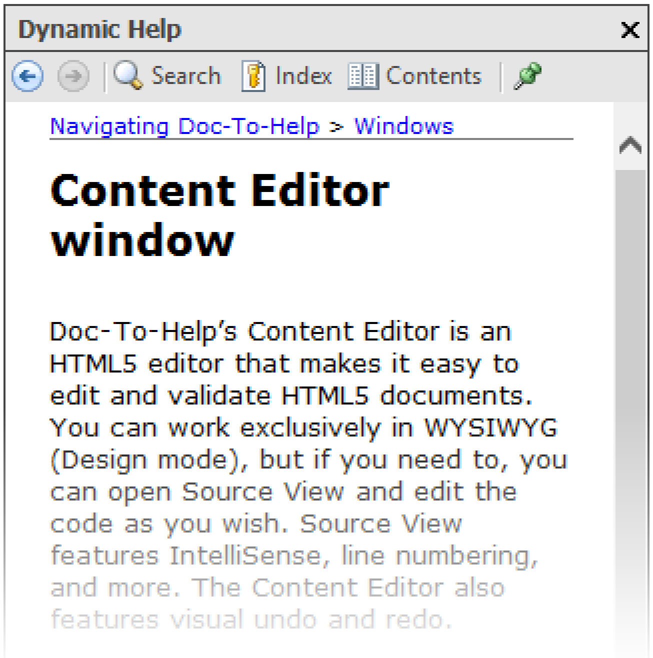
Translation window

The **Translation** window only appears when you open a translation project that has been uploaded to a SharePoint Translation Library.

See Uploading and Working with Documents in a SharePoint Translation Library on page 360 for more information.

Dynamic Help

The **Dynamic Help** window displays relevant Help automatically as you use your mouse to click within 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 106 — 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 186 to learn more.

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 .

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:

- NetHelp
- HTML Help
- EPUB on page 142
- Eclipse Help on page 144
- JavaHelp
- WinHelp
- Manual (Printed or Online)
- Microsoft Help Viewer on page 154
- Microsoft Help 2.0 on page 157

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 341.

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 **Add New Target** dialog box will open.

3. Enter the new Target **Name** and choose to either use the default Target properties, or copy the properties from an existing Target. Click **OK**. The new target will be added to the list. The Target can be further customized by editing the fields.

To delete a target, click the name of the target from the list and click **Remove Target**.

NetHelp Targets

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.

Doc-To-Help has two versions of NetHelp, **NetHelp Classic** and **NetHelp 2.0**. NetHelp 2.0 features a Responsive theme.

NetHelp 2.0 is jQuery-based. jQuery is a flexible new web technology that separates content from presentation and is easy to customize. For more information about the advantages of NetHelp 2.0, see the blog post Introducing NetHelp 2.0.

To switch from NetHelp Classic to NetHelp 2.0, open the **Help Targets** dialog box (Doc-To-Help **Home** tab, click the dialog box launcher on the **Target** ribbon group) and select the **NetHelp 2.0** check box. When you switch from NetHelp Classic to NetHelp 2.0, any theme customizations will need to be recreated in the Theme Designer (see Customizing NetHelp 2.0 Themes on page 200). If your NetHelp Classic project had context-sensitivity implemented, minor changes will need to be made when upgrading to NetHelp 2.0. See Context Sensitive Help in NetHelp for details.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 13. This section includes information on installing NetHelp locally and on a server.

See Creating Section 508 Compliant NetHelp on page 19 for more information on creating accessible NetHelp.

Please note that if no index entries are added to a project, the **Index** tab or pane will automatically not display in NetHelp 2.0 Targets.

Use the following options to configure your NetHelp output.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (The theme can also be changed using the **Theme** button in the **Target Design** ribbon group.)

NetHelp 2.0 Target: Use this check box to switch your target from NetHelp Classic to NetHelp 2.0. You can also switch a NetHelp 2.0 target back to NetHelp Classic by clearing this check box. By default, all new projects have this check box selected.

Attributes: The Attributes or Expression this target includes in the final build.

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

(This section is not used in NetHelp 2.0 targets.)

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

(This section is not used in NetHelp 2.0 targets.)

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** 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.

Comments

Use comments: Choose **DISQUS** (and specify your **DISQUS shortname**) to enable DISQUS commenting and feedback to your NetHelp 2.0 Targets. See Adding DISQUS Commenting to NetHelp 2.0 Targets on page 335 for more information.

DISQUS shortname: Unique identifier that references your DISQUS forum.

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 NetHelp Targets. Typically, the default will be *index.html*.

Default extension: The default extension for NetHelp target files is .htm. If desired, you can use this field to change the file extension to .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.

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

Search synonym list: If the **Search Type** is **JavaScript**, you can add a list of synonyms that can be included in the search by attaching a text file (.txt). The text file should include a group of synonyms on each line, with each separated by a space. For example:
car auto automobile
phone telephone cell mobile

Search Type: In NetHelp Classic targets, specifies the type of search used — **Java** or **JavaScript**. (NetHelp 2.0 targets use JavaScript search only.)

If NetHelp will be installed locally, JavaScript search is recommended to avoid dependence on Java being installed on end-user machines. JavaScript search supports exact phrase, Boolean, and fuzzy searches. "Exact phrase" search means that if you enclose a phrase in double quotes, NetHelp will search for that exact phrase in the Help file. "Boolean" search means that you can use "AND" or "OR" (no quotes) between words or exact phrases when searching. You can also use "NOT" before a word or exact phrase to exclude topics containing that phrase from the results. By default, if there is no "AND" or "OR" between words, "AND" is assumed. "Fuzzy" search will display alternative search options (and results) if the user enters a search term that is close to the term entered. The introductory text displayed for alternative terms is "Did you mean" and can be changed in the **Change suggestion field**. The terms used for AND, OR, or NOT can be changed in the appropriate **Search keyword AND**, **OR**, and **NOT** fields if localizing. A text file with search synonyms can be added using the **Search synonym list** field.

If a NetHelp will be installed on a server, Java search is recommended for larger projects. 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.

- **Normal** — No special features for Section 508 compliance are enabled.
- **Section 508 Compliance** — Special features for 508 compliance are enabled.

Email address: The text to be used for the **Email** option in NetHelp

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

Mark of the Web: In NetHelp targets, NetHelp installed locally can be given the Mark of the Web (MOTW). This means that anyone viewing your NetHelp locally will not receive a browser security warning first (this ActiveX security warning is dependent on the security settings of the machine). The Mark of the Web adds commentary text (by default: <!-- saved from url=(0014)about:internet -->) to every HTML file in the NetHelp Target. There are some issues with the MOTW; for example, links to PDFs will not open from NetHelp files using the MOTW. To learn more see:

- http://jtfassociates.com/using-the-mark-of-the-web-motw/
- http://msdn.microsoft.com/en-us/library/ms537628(VS.85).aspx

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. (This option is not used in NetHelp 2.0 targets.)

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. (This option is not used in NetHelp 2.0 targets.)

Generate XHTML: If selected, the online help output is generated as XHTML. (For NetHelp 2.0 and Eclipse Help targets, the output is always HTML5 in XML form.)

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.

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

(This section is not applicable for NetHelp 2.0 targets. In NetHelp 2.0, strings are edited using the **Theme Designer**, see Customizing NetHelp 2.0 Themes on page 200.)

Breadcrumbs label: For HMTL Help targets, specifies the text that precedes topic breadcrumbs (for example, “You are here:”).

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: Specifies the text that will be displayed on the Search tab.

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 address: 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.

Change suggestion: If the **Search Type** is **JavaScript**, this is the introductory text displayed before suggested fuzzy search term results. The default is "Did you mean."

Load search engine: This is the text displayed when the Help search engine is loading. The default is "Loading search engine …"

Search keyword AND: If the **Search Type** is **JavaScript**, this is the text used for AND searches in Boolean searches. (This option is for localization.)

Search keyword OR: If the **Search Type** is **JavaScript**, this is the text used for OR searches in Boolean searches. (This option is for localization.)

Search keyword NOT: If the **Search Type** is **JavaScript**, this is the text used for NOT exclusions in "Boolean" searches. (This option is for localization.)

HTML Help Targets

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 13.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (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.

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** 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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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, 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, 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, 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. (For NetHelp 2.0 and Eclipse Help targets, the output is always HTML5 in XML form.)

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.

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.

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:").

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.

EPUB Targets

Use the following options to configure your EPUB output.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 13.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (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.

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.

EPUB

Title: The title specified will display on the cover page of the EPUB and in the EPUB reader.

Author: The author name specified will display in the EPUB reader. (If none is specified, will use "Unknown".)

Identifier: The unique identifying number for this EPUB, required by the EPUB standard. Doc-To-Help generates this number automatically; click the **Refresh** button to the right of this field to regenerate the identifier.

Publisher: The publisher name specified will display in the EPUB reader.

Publication date: The publication date will display in the EPUB reader and is the current date by default.

Cover image: The image added with this field will be used for the EPUB's cover page and also on the Title page.

Language: The language of the book, required by the EPUB standard.

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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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, all expanding text (created with the **Inline Text** button) in the project will be displayed in the EPUB.

Show dropdown text: If selected, all dropdown text (created with the **Inline Text** button) will be displayed in the EPUB.

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.

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.

Eclipse Help Targets

Use the following options to configure your Eclipse Help output.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 13.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (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.

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.

Eclipse plugin

When you click the **View Target** button, all Eclipse target files will be copied to the "Eclipse directory\plugins\plugin name" folder, where "Eclipse directory" is the eclipse.exe installation directory specified in the Doc-To-Help Options on page 34 dialog box, and "plugin name" is a name constructed from the **ID** and **Version** properties.

ID: The plugin ID, for example, "com.mycompany.projectname". The **ID** property is the first part of the plugin directory name where the Eclipse Help target files will be stored. By default, the first part is "com", the second part is the **Provider** property, and the third part is the **Name** property. If you change the **Name** and/or **Provider** properties, the default ID value will also change.

Version: The plugin version, for example "1.0.1". This property is the second part of the plugin directory name where the Eclipse Help target files will be stored. The default is "1.0.0".

Name: The plugin name, by default the **Caption** property value.

Provider: The plugin provider name, “MyCompany” by default.

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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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.

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.

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.

JavaHelp Targets

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 13.

The Java runtime environment (JRE) must be installed on the end user's machine to view JavaHelp.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (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.

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** 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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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. (For NetHelp 2.0 and Eclipse Help targets, the output is always HTML5 in XML form.)

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.

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:").

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.

WinHelp Targets

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 13.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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.

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, 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, 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.

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.

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.

Manual Targets

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 13.

If you would like to set the section break type (odd, even, none) that begins each book chapter, you can do so in the **Project Styles** dialog box using the **Section break** property. See Paragraph Styles on page 171 for more information.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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.

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 Word format for the manual output. The default is .doc if Word 2003 or below is installed. If Word 2007 or above is installed, the default will be .docx.

Generate PDF target: If selected, a PDF version of your manual target will be generated along with the .doc or .docx version. Projects authored in Microsoft Word 2007/2010/2013 will use Word's built-in PDF converter. If using Word 2007, the built-in converter will only be used if the "2007 Microsoft Office Add-in: Microsoft Save as PDF" is installed from http://www.microsoft.com/downloads/en/details.aspx?FamilyID=f1fc413c-6d89-4f15-991b-63b07ba5f2e5&displaylang=en.
“Doc-To-Help PDF Converter” will appear in Word’s printer list and is used by Doc-To-Help in the conversion.

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 numbered 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.

Section break: Doc-To-Help automatically inserts a break between each of your Source documents in a Manual Target. If you have Word source documents that begin with a secondary heading style (for example, a Heading 2), use this option to control the type of section break that will be inserted in your Manual Target. The available values are: **Next page**, **Continuous**, **Even page**, **Odd page**. If you would like no page break, set it to **Continuous**. This option is only available if the **Create master document** option is selected.

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.

Microsoft Help Viewer Targets

Use the following options to configure your Microsoft Help Viewer output. Please note that this output is only used for developing help for Microsoft Visual Studio 2010 and above.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 13.

MHV resources:

- **MSHelpWiki.com**: http://www.mshelpwiki.com/wiki/tiki-index.php?page=Help3
- **MSDN Forum**: http://social.msdn.microsoft.com/Forums/en-US/devdocs/threads
- **Yahoo! Group**: http://tech.groups.yahoo.com/group/MSHelpViewer/

Basic

Name: The name of the target, which will also appear in the list on the left of the dialog box. This name is also used when you select a target using the **Select Target** button in the Target ribbon group on page 109.

Base name: If you would like your Target output to have a different name than the project name, enter it here.

Self-branded: If selected, allows you to use your own CSS and scripts for the Microsoft Help Viewer 1.x target, otherwise the target would use the default Microsoft Help Viewer 1.x CSS and scripts. Doc-To-Help supports only self-branded targets.

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 7. (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 7. (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. (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 162.

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

Deployment

Product vendor: The vendor name, used for installing and uninstalling the target in the Microsoft Help Viewer 1.x. The target's base name will be used if this field is left empty.

Product name: The target name, used for installing, uninstalling, and identifying the target in the Microsoft Help Viewer 1.x. The target's base name will be used if this field is left empty.

Product book: The target book, used for installing, uninstalling, and organizing the target in the Microsoft Help Viewer 1.x. The target's base name will be used if this field is left empty.

Topic version: The target topic's version. It is used to determine differences between version one and subsequent versions of the installed topic. It is 100 by default.

Doc-To-Help allows you to install a target and later reinstall newer versions into the local Microsoft Help Viewer 1.x. However, if the Target properties "Product vendor," "Product name," or "Product book" are changed between builds the old target version won't be uninstalled from the Microsoft Help Viewer 1.x.

Users will be able to uninstall or install targets manually using the Microsoft Help Library Manager. Doc-To-Help produces two install files: HelpContentSetup.msha and <target base name>.mshc.

Parent in Contents

Topic ID: The ID of the parent topic in Microsoft Help Viewer 1.x, which will be the parent for all root entries in the target table of contents.

Topic version: Version of the parent topic in Microsoft Help Viewer 1.x, which will be the parent for all root entries in the target table of contents (100 by default).

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.

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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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.

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 321.

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.

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.

Warnings about Limitations

Warn about Popup link: If selected, will write a warning to the build log about popup links usage in topics. Only self-branded targets support popup links, so warnings will be ignored for those.

Warn about Group link: If selected, will write a warning to the build log about group links usage in topics. Only self-branded targets support group links, so warnings will be ignored for those.

Warn about Keyword link: If selected, will write a warning to the build log about keyword links usage in topics. Only self-branded targets support keyword links, so warnings will be ignored for those.

Warn about Bookmark link: If selected, will write a warning to the build log about links to bookmarks in topics. This target type doesn't support links to bookmarks that are located in another topic, so those links will be ignored.

Warn about Mid Topic link: If selected, will write a warning to the build log about links to mid topics in topics. This target type doesn't support links to mid topics that are located in another topic, so those links will be ignored.

Warn about Contents title: If selected, will write a warning to the build log about the use of a custom contents title for a topic. This target type doesn't support topic contents title customization. Values of this property will be ignored.

Warn about Display title: If selected, will write a warning to the build log about the use of a custom display title for a topic. This target type doesn't support topic display title customization. Values of this property will be ignored.

Warn about Keyword level depth: If selected, will write a warning to the build log about the use of keywords higher than the second level. This target type supports only first and second level keywords. Keywords higher than the second level will be ignored.

Warn about 'Contents only' styles: If selected, will write a warning to the build log about the use of topics whose styles have been set to the "Contents only" property. This target type doesn't support topics with this style; all table of contents entries must be a topic, not just an empty book. The style "Contents only" property will be ignored.

Warn about 'Suppress empty topic' property: If selected, will write a warning to the build log about the use of topics whose styles have been set to the "Suppress empty topic" property. This target type doesn't support with this style, all table of contents entries must be topic, not just an empty book. The style "'Suppress empty topic" property will be ignored.

Microsoft Help 2.0 Targets

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 2002-2008.

For more information on Targets and how to distribute them, see Doc-To-Help Outputs and Deliverables on page 13.

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 **Select Target** button in the **Target** ribbon group.

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. (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. (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**. (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.

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** 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.

XML transformation: This property allows you to insert boilerplate content or code into every Help topic, for example: headers, footers, or code that automates analytics. Custom HTML5 code can be added inside the <head> element, above the topic title, or at the end of topic text. Click the **Wizard** button to open the **Transformation Wizard** dialog box and enter your custom code. Existing configuration files can be selected by clicking the ellipsis button.

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. (For NetHelp 2.0 and Eclipse Help targets, the output is always HTML5 in XML form.)

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.

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:”).

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 313. Text within **Variables** can also be conditionalized, see Creating Variables.
- **Topics** To apply, see Setting Topic Conditions on page 325.
- **Documents** To apply, see Setting Document Properties on page 297.

Conditions available are:

- **Platforms**
  Eclipse Help — Text/Topic/Document will be included in all Eclipse Targets.
  EPUB — Text/Topic/Document will be included in all EPUB Targets.
  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 MS Help 2.0 Targets.
  Microsoft Help Viewer — Text/Topic/Document will be included in Microsoft Help Viewer Targets.
  NetHelp — Text/Topic/Document will be included in NetHelp (browser-based 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) **Eclipse Help** — Text/Topic/Document will be included in this Eclipse Help Target.
  (Your project name) **EPUB** — Text/Topic/Document will be included in this EPUB Target.
  (Your project name) **HTML Help 1.x** — 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) **Microsoft Help 2.0** — Text/Topic/Document will be included in this MS Help 2.0 Target.
  (Your project name) **Microsoft Help Viewer** — Text/Topic/Document will be included in this Microsoft Help Viewer Target.
  (Your project name) **NetHelp** — Text/Topic/Document will be included in this NetHelp (browser-based uncompiled HTML Help) Target.
  (Your project name) **Manual** — Text/Topic/Document will be included in this printed manual Target (.doc, .docx, and .pdf).
  (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 162 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 133. Click the drop-down on the **Attributes** field and choose the appropriate Attribute(s) or create a custom Expression. See Defining Attributes on page 162.

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 313. Build attributes can also be assigned to specific Character Styles on page 168, Topics on page 325, and even entire Documents on page 297. Then specify the attributes to include in the build of each Target on page 133 — 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 313.
   - To applicable **Character Styles**. See Defining Character/Paragraph Styles and Topic Types on page 168.
   - To **Text Variables**. See Creating Variables.
   - To **Topics**. See Setting Topic Conditions on page 325.
   - To entire Documents. See Setting Document Properties on page 297.
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 133 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 109.

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.

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 119.

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.

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 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.
For NetHelp 2.0 Targets, the Navigation Pane will be disabled for the main window because the visibility of the Contents, Search and Index tabs is managed by editing the Theme. See Customizing Themes.

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.

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.

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 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 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.

Microsoft Help Viewer Window

The size and position of windows in the Microsoft Help Viewer target (used for developing help for Microsoft Visual Studio 2010 and above) cannot be customized.

EPUB Window

The size and position of windows in the EPUB target cannot be customized.

Microsoft Help 2.0 Window

The size and position of windows in the Help 2.0 target (used for developing help for Microsoft Visual Studio 2005 and 2007) 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.

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 7 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 303.

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 321.

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, click on the **Add New Style** drop-down in the left panel. Choose **Paragraph Style**, **Character Style**, or **Topic Type** from the list. The **Add New Style** dialog box will open.
2. Enter the new **Style Name** and choose to either use the default style properties, or copy the properties from an existing style. Click **OK**. The new style will be added to the list. The style can be further customized by editing the fields.

If you added a new Paragraph style and chose **Use default properties**, 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 171 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 119.

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 (**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.

Section break: This option applies to Manual Targets only. It is used to set the type of section break at the beginning of each chapter of the master document. It can be used to set book chapters to begin on an odd page (a typical scenario). To do so, choose the Paragraph Style for the first style in each chapter (usually Heading 1) and set this property to Odd Page. In new projects this property is set to "Odd Page" for the Heading 1 style by default; in existing projects it is set to "None" by default.

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 that may be edited, or you can create new ones.

Navigation

Auto Subtopic Links: If selected, subtopic buttons and links will be displayed automatically for this paragraph style, if this Heading style has topics below it in the hierarchy. For example, if a topic with a Heading 1 style has topics with the Heading 2 style following it (these are subtopics), then in the online Help the Heading 1 topic will automatically have links to the Heading 2 topics at the bottom.

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 and links will be displayed automatically for this paragraph style, if this Heading style has topics below it in the hierarchy. For example, if a topic with a Heading 1 style has topics with the Heading 2 style following it (these are subtopics), then in the online Help the Heading 1 topic will automatically have links to the Heading 2 topics at the bottom.

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.

Use comments: If selected, all topics with this Topic Type will have DISQUS commenting enabled in NetHelp 2.0 Targets. See Adding DISQUS Commenting to NetHelp 2.0 Targets on page 335 for more information.

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.

Exclude from Search: If selected, all topics with this Topic Type will not be displayed in search results.

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.

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 168 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 File Style Sheets 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 178.

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 HTML5 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 your machine.

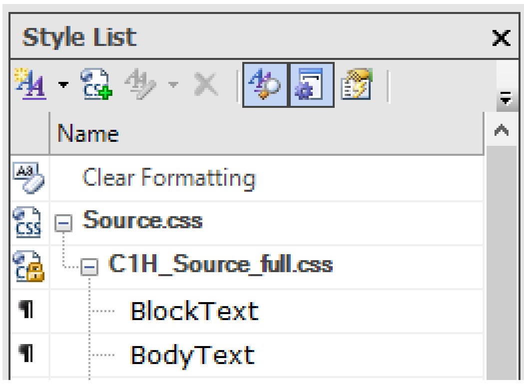
To modify a Style

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 File Style Sheets 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.

Note: 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 176 to learn how to open the **Style List** window.

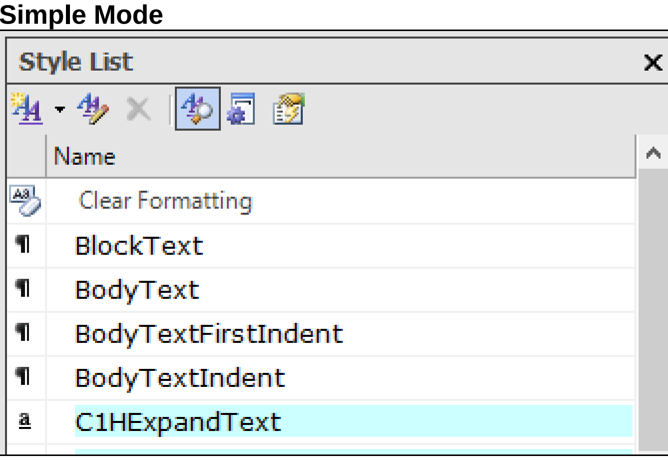
For more information on editing styles, see Editing a CSS on page 176. If your project does not contain HTML5 (.xml) source files, you can edit the styles with the **Style List**, but cannot apply them with it. The styles will be applied using the appropriate HTML editor.

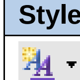
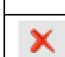
For information on applying styles using the **Styles List**, see Applying Styles in the Content Editor.

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 **Content 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

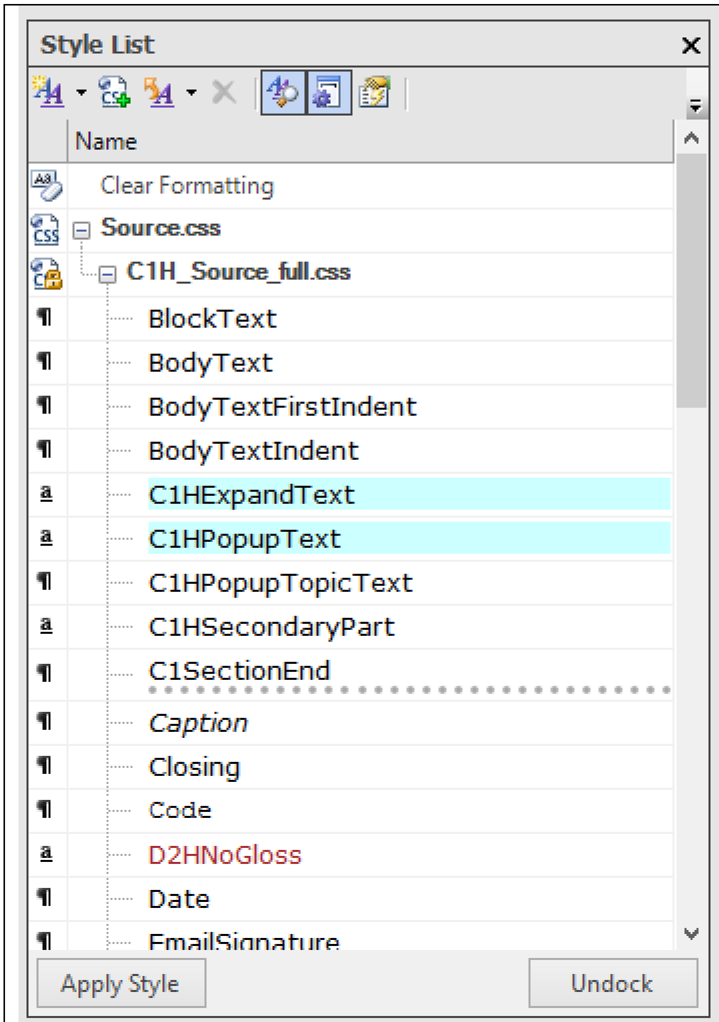


Style List Toolbar	
	**New Style** — Create an original style or tag
	**Modify Style** — Create a style based on an existing one (a derived style). If the style has a behavior associated with it, that behavior will be retained.
	**Delete Style** — Delete the chosen style. Read-only styles can not be 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.	

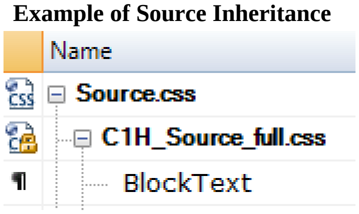
When you click the **Advanced Mode** button in the **Style List** toolbar, the display will changed to illustrate the inheritance of Styles and provide a few additional options.

Advanced Mode

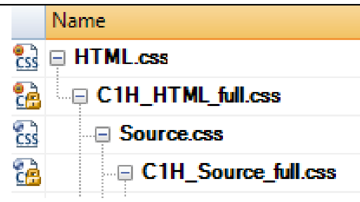


Style List Toolbar	
	New Style — Create an original style or tag
	Add Style Sheet — Add a Style Sheet to the list.
	Modify Style — Create a style based on an existing one (a derived style). If the style has a behavior associated with it, that behavior will be retained. (If you select a .css in the list, 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.
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**



Icons	
	This icon indicates the CSS is unlocked and styles are editable.

		
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.		This icon indicates the CSS is read-only and not editable (but you can create derived styles from it).
		Indicates a paragraph style.
Creating Styles		Indicates a character style.
Styles can be created by using the **Style List** toolbar, or by using the right-click menu.		An arrow next to a paragraph style or character style icon indicates that a derived style has been created from this style.

How Style Sheets are inherited

When you create a project that uses HTML5 and/or HTML 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 File Style Sheets to learn more about the available style sheets and how they work together.

When you create a new project in Doc-To-Help 2014, 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 168 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 C:\Program Files [or Program Files (x86)]\ MadCap Software\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 129.

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 (HTML5, HTML, 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
- Entire topics

To open the Variables window

From the **Project** tab, Project ribbon group on page 119, choose the **Variables** toolbar button. The Variables window on page 129 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 162 (Project tab > Project ribbon group on page 119 > **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 **HTML5**, **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 2007/2010/2013, 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 HTML5). 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 HTML5 or HTML.

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 263.

Variables are inserted in the Content Editor window on page 128 using the **Variable** button in the Insert tab on page 115. See Inserting a Variable on page 315.

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 119.

General

Location: The location where this project is stored. (Read-only)

Title: The name of the Doc-To-Help project without the .d2h extension. This name is used as the default value for several Help Target and Help window properties. For the HTML Help window, Help 2.0, JavaHelp, and NetHelp, it is the default **Caption**. For the WinHelp window, WinHelp, and Manual, it is the default **Title**. For the MSHelp Viewer, it is the default **Product Vendor**, **Product Name**, and **Product Book**.

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, HTML5, and HTML 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: The rule used to name the .htm/html files generated for each topic in browser-based outputs. The rule is used when the topic URL is generated automatically. If you change this rule, you can apply the new rule to all existing topics in your project. You can also modify the URL of a topic manually, which will override the default rule. You can view the URL of each topic in the **Topic Properties** dialog box (**URL** field).

- **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.

Truncate file name length: This property limits the length of the .htm/html file name generated for each topic in browser-based outputs to this number of characters. The default is 64. You can view the URL of each topic in the **Topic Properties** dialog box (**URL** field).

'On Page' text: Specifies the default text used to complete cross-references in a printed manual target. For example "See Creating a Topic **on page** 5."

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) 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.

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.
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.
Language for Translation: This property is set automatically when you choose the default language for your project in the **New Project Wizard**.
Default Spelling: This drop-down is used to set the default spell-checker language for all HTML5 source documents in the project simultaneously. The default value, **System locale**, is taken from the **Control Panel** > **Region and Language** > **Administrative** > **Current language** setting for non-Unicode programs. Please note that this setting will not override HTML5 documents where the language for the spell-checker has already been defined using the **Editor** tab > **Spelling** button.

Update Customized Table of Contents in Build: If selected, any customized tables of contents in your project will be updated with all new topics added to your project since the TOC was created. Target-specific TOCs will not be updated.

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 182 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, HTML5, and HTML 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 119. See Documenting Your Class Library with Microsoft® Sandcastle on page 409 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 **File** tab > **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 **File** tab > **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 **File** tab > **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 [or Program Files (x86)]\ MadCap Software\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 87.

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 323 for details.

To automatically generate Context IDs

1. Open the **Project Styles** dialog box. (See Defining Character/Paragraph Styles and Topic Types on page 168 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 182 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 401 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 127.

To view/edit/assign a topic's Context ID for HTMLHelp, NetHelp, and WinHelp projects

1. Open the Topics window on page 127.
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 Classic Help target
Project folder name: **NetHelp**
File name: _contextIds.js
Additional file needed: D2H_ctxt.*, located in C:\Program Files [or Program Files (x86)]\ MadCap Software\DocToHelp\Context-Sensitive Help\NetHelpClassicSamples.zip

NetHelp 2.0 Help target
Project folder name: **NetHelp**
File name: context.xml
Additional file needed: D2H_ctxt.*, located in C:\Program Files [or Program Files (x86)]\ MadCap Software\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 Classic on page 188 or Context Sensitive Help in NetHelp 2.0 on page 192 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 187 or Context Sensitive Help in Microsoft Help 2.0 on page 188 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 321 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 2002-2008. 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 321 for more information.)

Context Sensitive Help in Microsoft Help Viewer

The Microsoft Help Viewer Help format is used only in help systems integrated with Visual Studio 2010 and above. 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 321 for more information.)

Context Sensitive Help in NetHelp Classic

NetHelp Classic, 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 Contents, Index and Search tabs.

NetHelp Classic 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 186 for more information.

To call NetHelp Classic 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 [or Program Files (x86)]\ MadCap Software\DocToHelp\Context-Sensitive Help\ NetHelpClassicSamples.zip file

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

Note: If your project is a **NetHelp Classic** project with context sensitive help implemented and you switch to **NetHelp 2.0**, you must replace the existing D2H_ctxt.* files with the newer versions in the C:\Program Files [or Program Files (x86)]\ MadCap Software\DocToHelp\Context-Sensitive Help folder and recompile if necessary.

Samples demonstrating how to use context-sensitive help with NetHelp Classic in each of these environments can be found in the Samples directory in the C:\Program Files [or Program Files (x86)]\ MadCap Software\DocToHelp\Context-Sensitive Help\ NetHelpClassicSamples.zip file. 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** — default.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. Please note that **NetHelp 2.0** and **NetHelp Classic** have different functions and examples. For NetHelp 2.0 see Context Sensitive Help in NetHelp 2.0 on page 192.

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 186 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 C:\Program Files [or Program Files (x86)]\ MadCap Software\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 186 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 [or Program Files (x86)]\ MadCap Software\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 186 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 [or Program Files (x86)]\ MadCap Software\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 [or Program Files (x86)]\ MadCap Software\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);
```

Context Sensitive Help in NetHelp 2.0

NetHelp 2.0, 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 2.0 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 2.0 navigation with the Contents, Index and Search tabs.

NetHelp 2.0 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 186 for more information.

To call NetHelp 2.0 from your application or web pages, use one of the D2H_ctxt.* source files provided with Doc-To-Help; they define the ShowHelp function needed to enable context-sensitive help.

Note: The D2H_ctxt.* files are installed by default in the C:\Program Files [or Program Files (x86)]\ MadCap Software\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 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

Note: If your project is a **NetHelp Classic** project with context sensitive help implemented and you switch to **NetHelp 2.0**, you must replace the existing D2H_ctxt.* files with the newer versions in the C:\Program Files [or Program Files (x86)]\ MadCap Software\DocToHelp\Context-Sensitive Help folder and recompile if necessary.

Samples demonstrating how to use context-sensitive help with NetHelp in each of these environments can be found in the C:\Program Files [or Program Files (x86)]\ MadCap Software\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
- **VB.NET** — VB.NET.vbproj
- **WebPages** — index.html

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 (See NetHelp 2.0: Supported Parameters and How to Use Them on page 196).

The function is described below in more detail. Please note that **NetHelp 2.0** and **NetHelp Classic** have different functions and examples. For NetHelp Classic see Context Sensitive Help in NetHelp Classic. on page 188.

Function ShowHelp in JavaScript (web applications and web pages)

Syntax

```
function nethelp.showHelp(options)
```

The function accepts only one parameter “options” which is an object with the following properties:

Parameters

- **query** (string or object) – pass a string with a relative URL to open a topic by URL, or pass an object with two properties to perform a query by context id, keyword or group or perform a search:
  - **key** (string) - can be one of the following values: "*keyword*", "*group*", "*id*" and "*search*”.
  - **value** (string) – depends on the key parameter, if key is “*id*” then value should contain a context id, if key is “*keyword*” then value should contain a keyword and so on.
- **tab** (string) - This parameter specifies the tab (by index or name) that will be shown when NetHelp is loaded. If any other value is specified the parameter will be ignored. Possible values: for Contents *0* or *toc*, for Index *1* or *index*, for search *2* or *search*. The default value is *0*.
- **url** (string) — URL of the main web page of the help or a direct topic URL. 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 [or Program Files (x86)]\ MadCap Software\DocToHelp/Samples/Pittsburgh250XMLSource/NetHelp/index.html***; or, for a server-deployed NetHelp target it can be ***http://www.mycompany.com/myhelp/index.html***. 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.

- **window** (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.
- **topicOnly** (bool) – If *true* then the topic will display without the Contents, Index and Search and without header panel and toolbars. 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. If *false* is passed then the complete help system with Contents, Index and Search and header panel and toolbars will display. The user will be able to navigate through the links in the topic and through the Contents, Index and Search. The default value is *false*.

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: nethelp.showHelp({ url:
"../../../Samples/Pittsburgh250XMLSource/NetHelp/index.html", query: { id: "91"
} })'>Show topic with Context ID = 91</a>
```

Function 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 in your application.
- Show help in a separate browser window. In this case you don't need to use the WebBrowser control.

Syntax

C#

```
ContextSensitiveHelp.ShowHelp(Query query, Parameters param, string url, object
window)
```

Visual Basic .NET

```
ContextSensitiveHelp.ShowHelp(ByVal query As Query, ByVal param As Parameters,
ByVal url As String, ByVal window As Object) As Boolean
```

Parameters

- **query** (Query class) — represents Query parameters: QueryType, Value
  - **QueryType** (QueryTypes enumeration) – possible values: *ContextID*, *Keyword*, *Group*, *Search*, *TopicUrl*, *None*. Specifies what type of query must be performed. It is used together with the Value parameter.

    - **Value** (string) – corresponds to specified QueryType and contains either a context id value, keyword, group, search query or a topic URL.
- **param** (Parameter class) – represent Hash parameters: ActiveTab, TopicOnly
    - **Tab** (Tabs enumeration) – Specifies the active tab when NetHelp is loaded. Possible values: *Default*, *Contents*, *Index*, *Search*.
    - **TopicOnly** (bool) - If *true* then the topic is displayed without the Contents, Index and Search and without header panel and toolbars. 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. If *false* is passed then the complete help system is displayed with Contents, Index and Search and header panel and toolbars. The user will be able to navigate through the links in the topic and through the Contents, Index and Search.
- **url** (string) — URL of the main web page of the help or a direct topic URL. 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 [or Program Files (x86)]\ MadCap Software\DocToHelp /Samples/Pittsburgh250XMLSource/NetHelp/index.html***; or, for a server-deployed NetHelp target it can be ***http://www.mycompany.com/myhelp/index.html***. 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.
- **window** (WebBrowser object) — If this parameter is *null* (Nothing in Visual Basic .NET), 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.

To enable the function

- Add one of the D2H_ctxt.* files to your project; use D2H_ctxt.cs for C# or D2H_ctxt.vb for Visual Basic .NET.
- 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#

1. **Open default topic and Index tab**

```
string url = “file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html”;

Query query = new Query();

Parameters parameters = new Parameters(Tabs.Index);

ContextSensitiveHelp. ShowHelp(query, parameters, url, webBrowser);
```

2. **Open default topic with default tab (Contents) in topic only mode**

```
string url = “file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html”;

Query query = new Query(QueryTypes.ContextID, "91");

Parameters parameters = new Parameters(Tabs.Default, true);

ContextSensitiveHelp.ShowHelp(query, parameters, url, webBrowser);
```

3. **Open specified topic and search tab**

```
string url = “file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html”;
Query query = new Query(QueryTypes.TopicUrl, "documents/pittsburgh_sports.htm");
Parameters parameters = new Parameters(Tabs.Search);
ContextSensitiveHelp.ShowHelp(query, parameters, url, webBrowser);
```

To open NetHelp in an external browser window pass null instead of the "webBrowser" value.

Visual Basic .NET

1. **Open default topic and Index tab**

```
Dim url As String, query As Query, parameters As Parameters
url = "file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html"
query = New Query()
parameters = New Parameters(Tabs.Index)
ContextSensitiveHelp.ShowHelp(query, parameters, url, webBrowser);
```

2. **Open default topic with default tab (Contents) in topic only mode**

```
Dim url As String, query As Query, parameters As Parameters
url = "file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html"
query = new Query(QueryTypes.ContextID, "91")
parameters = new Parameters(Tabs.Default, true)
ContextSensitiveHelp.ShowHelp(query, parameters, url, webBrowser);
```

3. **Open specified topic and search tab**

```
Dim url As String, query As Query, parameters As Parameters
url = "file:///C:/Program Files/MadCap Software/DocToHelp
/Samples/Pittsburgh250XMLSource/NetHelp/index.html"
query = new Query(QueryTypes.TopicUrl,
"documents/pittsburgh_sports.htm")
parameters = new Parameters(Tabs.Search)
ContextSensitiveHelp.ShowHelp(query, parameters, url, webBrowser);
```

NetHelp 2.0: Supported Parameters and How to Use Them

Example of a URL for NetHelp 2.0 context-sensitive help:

```
http://mysite.com/target/index.html?tab=1&topiconly=true#!documents/topic.html
```

Here the section after **?** is called ***'query'*** and the part after **#** is called ***'hash'*** according to the common URL terminology, see http://en.wikipedia.org/wiki/Url#Syntax (any URL has the following structure):

```
scheme://domain:port/path?query_string#hash_fragment).
```

QUERY parameters

topiconly=true
All NetHelp elements will be hidden, only the topic text will be shown and a css-class "*topic-only*" will be applied to the **<body>** element.

tab=0|toc|1|index|2|search
This parameter specifies the tab (by index or name) that will be shown when NetHelp is loaded. If any other value is specified the parameter will be ignored.

HASH parameters

#!url
"url" is a path to a topic, either relative to the root NetHelp folder or absolute.

#?query
"query" is a key-value pair that defines a context sensitive query in the following format: "*key=value*". Key can be one of the following values: "*keyword*", "*group*", "*id*" and "*search*".

Examples:

Open NetHelp with Search tab active

```
http://mysite.com/target/index.html?tab=2
```

or

```
http://mysite.com/target/index.html?tab=search
```

Open NetHelp and specified topic:

```
http://mysite.com/target/index.html#!documents/topic.html
```

Open NetHelp and specified topic and scroll topic to the "bookmark":

```
http://mysite.com/target/index.html#!documents/topic.html#bookmark
```

If the keyword "*sports*" is assigned only to one topic then this topic will be displayed, if the keyword is assigned to several topics then the list of topics will be displayed, otherwise if no topics are associated with the keyword the "No topics found" will be displayed.

```
http://mysite.com/target/index.html#?keyword=sports
```

If the group "*sports*" is assigned only to one topic then this topic will be displayed, if the group is assigned to several topics then the list of topics will be displayed, otherwise if no topics are associated with the group the "No topics found" will be displayed.

```
http://mysite.com/target/index.html#?group=sports
```

Displays topic for search query "*sports*":

```
http://mysite.com/target/index.html#?search=sports
```

Open topic with context id "*10*", if there is no topics for the specified context id "No topics found" will be displayed

```
http://mysite.com/target/index.html#?id=10
```

You can use the Query and Hash parameters at the same time. Examples:

Open topic with hidden NetHelp elements (no tabs, header, toolbars):

```
http://mysite.com/target/index.html?topiconly=true#!documents/topic.html
```

Open specified topic and tab Index:

`http://mysite.com/target/index.html?tab=index#!documents/topic.html`

Search for "*sports*" and open search tab

`http://mysite.com/target/index.html?tab=search#?search=sports`

Search for "*sports*" and display results in topic only mode

`http://mysite.com/target/index.html?topiconly=true#?search=sports`

5.

Customizing Themes with the Theme Designer

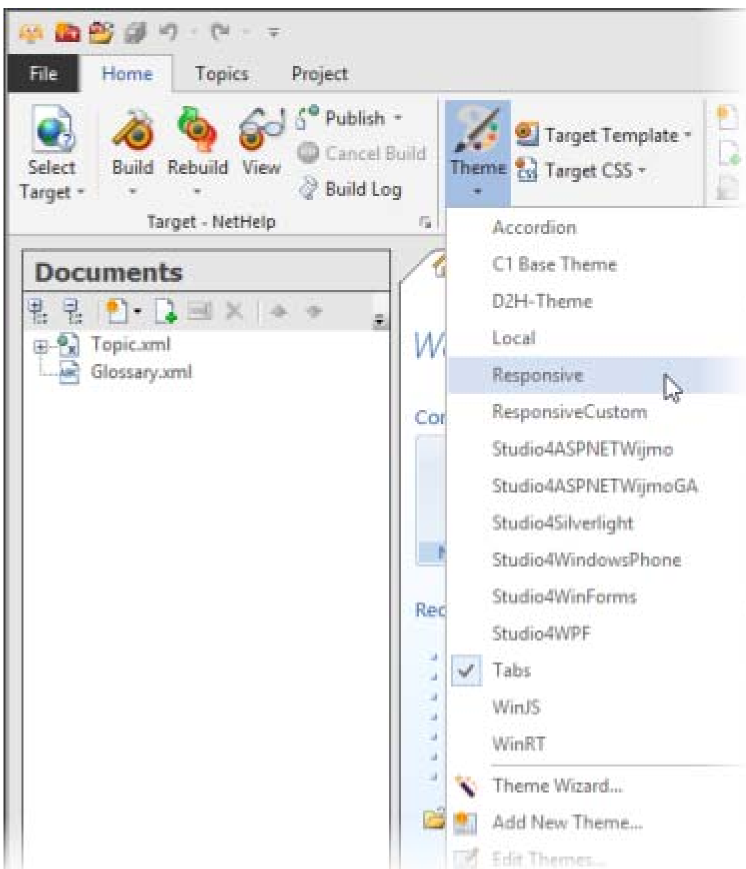
Doc-To-Help uses **Themes** to style the "skin" that surrounds your online Help content, as well as the button display and labels, icons, colors, and so much more. Several pre-defined themes are included for each Target, and it is easy to create a custom theme to meet your needs and preferences.

For information on customizing:

- **NetHelp 2.0** Targets – see Customizing NetHelp 2.0 Themes on page 200.
- **Eclipse Help** and **EPUB** Targets – see Customizing Eclipse Help and EPUB Themes on page 230.
- **NetHelp Classic**, **HTML Help**, **Microsoft Help Viewer**, **Microsoft Help 2.0**, or **JavaHelp** Targets – see Customizing NetHelp Classic, HTML Help, Microsoft Help Viewer, Microsoft Help 2.0, and JavaHelp Themes on page 235.

To select and preview a Theme

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). If the Theme does not open automatically, click the **Theme** button.



Adding a Theme or Theme Configuration

In order to create a custom Theme or Theme Configuration, you must first add a new Theme.

To add a theme

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.**
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.

Note: You can also create a new Theme by clicking the **Add New Theme** button in the **Theme Preview** dialog box.

To add a theme configuration (Eclipse Help and EPUB only)

1. Create a new Theme.
2. In the Theme Preview dialog box, click the **Add New Theme Configuration** button.
3. **Name** the new configuration, and choose the **Source Configuration.**
4. Click **OK**.

Note: Customized Themes and Configurations are stored by default in the \\My Doc-To-Help Projects\Doc-To-Help\Themes\ folder.

Customizing NetHelp 2.0 Themes

The following Themes are included with **NetHelp 2.0** Targets. These themes are the starting point for creating your customized Themes.

Target	Themes Available	Description
NetHelp 2.0	**Responsive**	• Designed to work well on both desktop and mobile platforms • Adapts to the window size automatically • Works on Android 4.0 or later and iOS. On the desktop, IE8 or later • The language will be English by default • Style Sheets available include jquerymobile\standard, jquerymobile\light, and jquerymobile\dark
	Tabs (this is the default Theme)	• Tripane window with Contents, Index, Search tabs at left • Breadcrumbs above Help content • Previous, Next, Home, Print Email buttons • The language will be English by default; the Style Sheet wijmo\aristo
	Accordion	• Tripane window with Contents, Index, Search accordion buttons at left • Breadcrumbs above Help content • Previous, Next, Home, Print Email buttons • The language will be English by default; the Style Sheet wijmo\aristo

The Responsive Theme is based on jQueryMobile 1.3.2. The Tabs and Accordion Themes are based on jQuery UI.

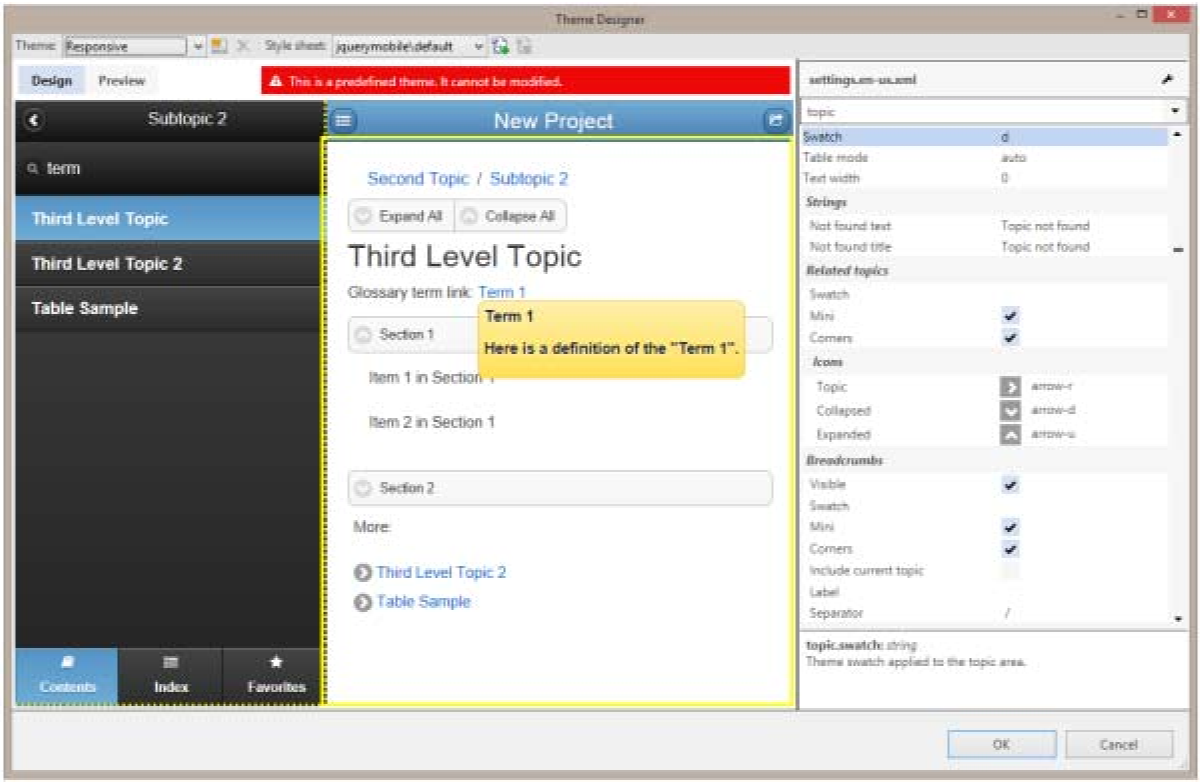
The Responsive Theme adapts to the current window size (screen resolution) — if the window size is too small, the Side panel is closed automatically so that only the topic is displayed. In addition, visual elements such as breadcrumbs, buttons, and related topics change their appearance and size. This Theme also includes a Favorites tab which allows end users to "favorite" topics (the favorite is stored on the user side).

To create a new theme

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.** (Responsive, Tabs, or Accordion)
4. Click **OK**. The **Theme Designer** will open.

   You can now customize your Theme. See Using the NetHelp 2.0 Theme Designer on page 201.



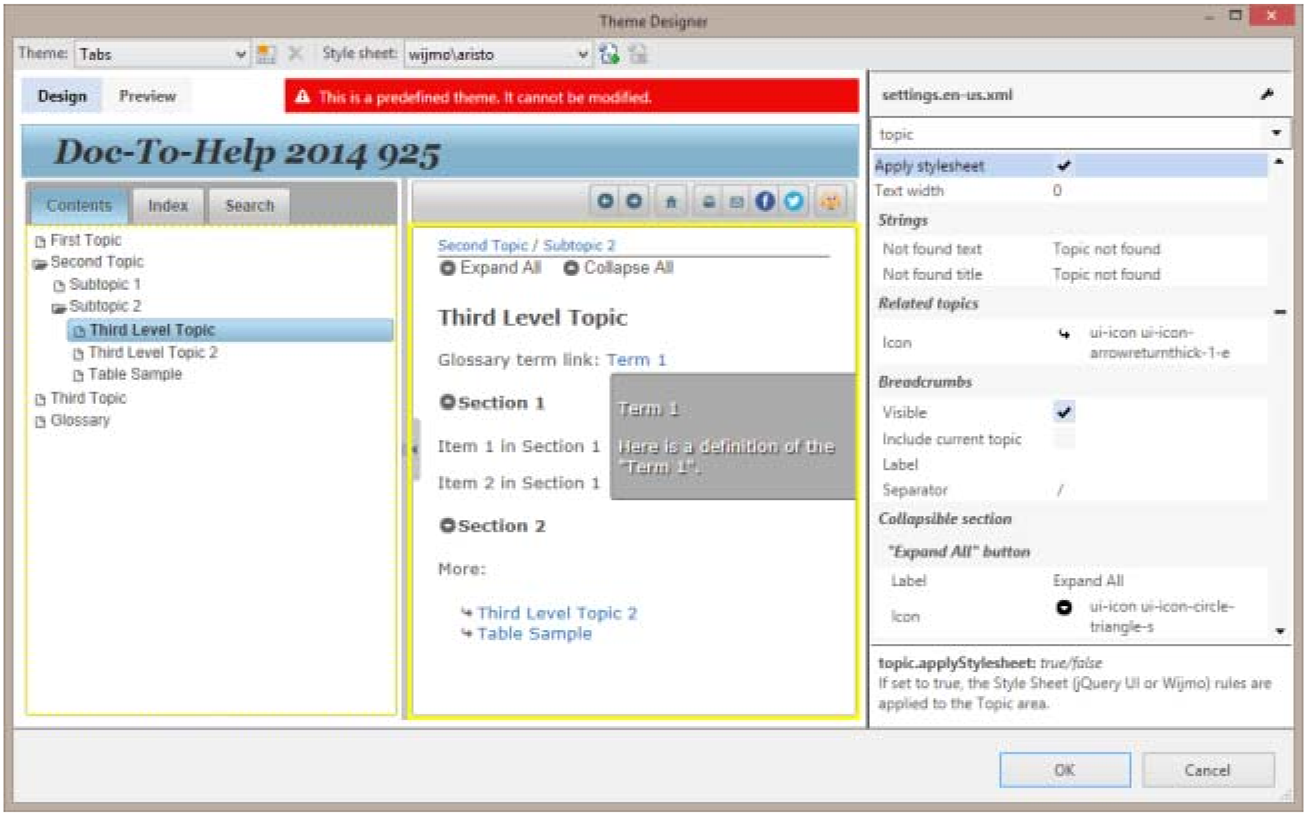
Note: You can also create a new Theme by clicking the **Add New Theme** button in the **Theme Designer**.

In order to use the NetHelp 2.0 Theme Designer, Microsoft Internet Explorer 8 must be installed on your machine; IE 9 and above are recommended.

For information on customizing:

- **NetHelp Classic**, **HTML Help**, **Microsoft Help Viewer**, **Microsoft Help 2.0**, or **JavaHelp** Targets – see Customizing NetHelp Classic, HTML Help, Microsoft Help Viewer, Microsoft Help 2.0, and JavaHelp Themes on page 235.
- **Eclipse Help**, and **EPUB** Targets – see Customizing Eclipse Help and EPUB Themes on page 230.

Using the NetHelp 2.0 Theme Designer



The NetHelp 2.0 Theme Designer makes it easy to create Themes visually. You can change the following (and more) quickly and easily. Please note that the options in the Theme Designer vary based on the theme chosen. The Responsive theme options are slightly different than the options for Accordion or Tabs.

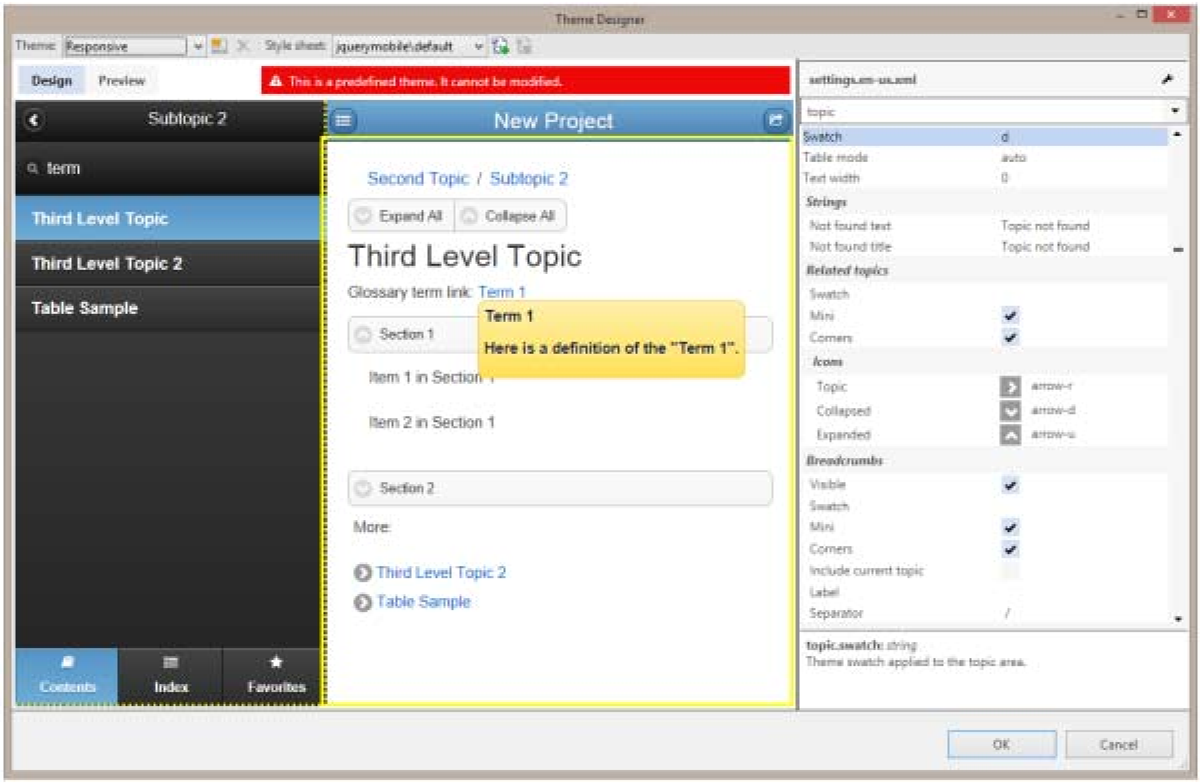
- **Header** (add a logo, expand/reduce the height of the header)
- **Toolbar buttons** (add/delete toolbar buttons, change the button order and alignment)
- **Icons** (select a new icon for any button, upload your own, or display no icon)
- The **table of contents**, **index**, and **search** areas (change the title that appears in each, the messages, the visibility of the area)
- Topic **breadcrumbs** (turn breadcrumbs on/off, add a label, include the current topic)
- **Language settings** (choose a different language for the theme — 14 are included, add your own language configuration file, or edit any of the strings in the selected configuration file)
- The **colors and fonts** used. All you need to do is choose a different style sheet, download a new one from the jQuery ThemeRoller (Tabs or Accordion Themes) or jQuery Mobile ThemeRoller (Responsive theme), or create a new one with the appropriate ThemeRoller.

Note: Customized Themes are stored by default in \\My Doc-To-Help Projects\Doc-To-Help\Themes. You can change this location using the Doc-To-Help **Options** dialog box, **Files** button. (To open the **Options** dialog box, choose the **File** tab > **Tools** > **Options**.)

You can preview how your changes will look and behave by clicking the **Preview** button in the Theme Designer. You can't edit the Theme in Preview mode.

NetHelp 2.0 Theme Designer Settings (Responsive)

The Responsive Theme adapts to the current window size/screen resolution — if the window size is too small, the Side panel is closed automatically so that only the topic is displayed and visual elements such as breadcrumbs, buttons, and related topics change their appearance and size. This Theme also includes a Favorites tab which allows end users to "favorite" topics (the favorite is stored on the user side). This Theme works on Android 4.0 or later and iOS; on the desktop, IE8 or later.



The Theme Designer settings can all be edited in the Settings area on the right side of the Theme Designer. To find the field you'd like to edit quickly, simply select it in the area on the left. The Settings area will display the field you need to edit. For global settings (**General**, **Context Sensitive**, **Toolbars**, and **Accessibility**), use the drop-down to display those fields.

There are several settings that are common throughout the Theme Designer for the Responsive Theme:

- **Swatches** —"Swatches" are used to set the color scheme of most elements. You can change appearance of a swatch or create new ones with the jQueryMobile Theme Roller. By default most elements have an empty value for the Swatch setting, which means that the swatch will use the settings of its "parent". For example, if the TOC has an empty swatch, it means that TOC will use the swatch of the Side panel. By default only the Side panel and the Topic have swatch values, which allow you to change the appearance for many swatches with two settings. There are five swatches in the default theme: a, b, c, d, e.
- **Mini** — If chosen (the default), the compact view of the element will be used. The compact view uses less vertical height.
- **Corners** — If chosen, the element will have rounded corners.

General (general)

The General settings are global settings for this Theme.

Right-To-Left — If selected, the Theme text direction will be set for right-to-left languages. If you choose the Hebrew configuration file (settings.he-il.xml) for this Theme, this check box will be selected automatically.

Update title — Switches the header title displayed on the browser tab from the Topic title to the title of the project. If you would like the browser tab to remain static, rather than changing as the user navigates the help, select this check box.

Default swatch — The swatch chosen here will controls the color scheme for all the elements in the theme, unless you change each individually. You can also change the Topic and/or Side Panel swatches to change those areas separately.

Fav icon — Specifies which icon will appear in the browser when your users mark your NetHelp as a "Favorite."

Context Sensitive (contextSensitive)

The Context Sensitive settings are global settings for this Theme.

Strings

Title — The title used in the browser window title when a list of keywords, groups, or search results is shown on the page. This string can contain macros #{key} and #{value} that are replaced with the context-key ("keyword", "group", "id", or "search") and context-value. Default: "Topics for #{key}: #{value}"

Not found message — This message is shown when the result is empty (when the context-key is "keyword", "group", or "id"). Default: "No topics found"

Not supported message — This message is shown when the specified context-key is not supported. Default: "The key "#{key}" is not supported"

Accessibility (accessibility)

The Accessibility settings are global settings for this Theme. See Creating Section 508 Compliant Help on page 19 for more information about Doc-To-Help's accessibility features.

Link menu header — This text is read by the accessibility device if a keyword or group link has more than one topic to display. Default: "#{count} topics found"

Contents

Closed book — 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. Default: "Closed book without topic"

Open book — 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. Default: "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. Default: "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. Default: "Open book with topic"

Topic — This text is read by the accessibility device when you mouse over a topic. Default: "Topic"

Collapsible sections

Collapsed section — This text is read by the accessibility device for the collapsible section icons. Default: "Click to expand"

Expanded section — This text is read by the accessibility device for the expanded section icons. Default: “Click to collapse

Buttons (buttons)

The Button settings control the label, icon, behavior, and text displayed on the buttons of the NetHelp 2.0 toolbar.

If you would like to remove, rearrange, change the alignment, or add a custom button to the NetHelp 2.0 toolbar, use the **Toolbar** (on page **Error! Bookmark not defined.**toolbars.right and toolbars.left) on page 213 fields.

Label — The name that will be used for the label and tooltip text of the button.

- **prev** — Default: “Previous”.
- **next** — Default: “Next”
- **home** — Default: “Home”
- **print** — Default: “Print”
- **email** — Default: “Email” The email address used for this button is support@mycompany.com by default. This should be changed in the **Help Targets** dialog box.
- **twitter** — Default: “Twitter”
- **facebook** — Default: “Facebook”
- **favorites** — Default: “Favorites”
- **sidePanel** — Default: “Side panel” (This button opens and closes the Side Panel.)
- **actions** — Default: “Actions” This button is displayed only when a toolbar is collapsed. Clicking this button will display a menu containing all the buttons from the collapsed toolbar. When a toolbar is not collapsed this button is hidden. If there is no Actions button on a toolbar it can't be collapsed.
- **poweredBy** — Default: “Created with Doc-To-Help” The button also includes a default click action that opens the Doc-To-Help website. This click action can be edited.

Icon — The icon displayed on the button. You can choose an alternate icon, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display. There are more than 300 vector-based FontAwesome icons available for your use, as well as others.

Icon position — Choose the position of the icon on the button.

Show Label — If selected, the label text will be displayed on the button.

Collapsible — Each button has a setting named Collapsible. When it is checked, it means that the button can be hidden when there is not enough width to display all buttons on a toolbar and the toolbar must be collapsed. When it is not selected the button will always be visible.

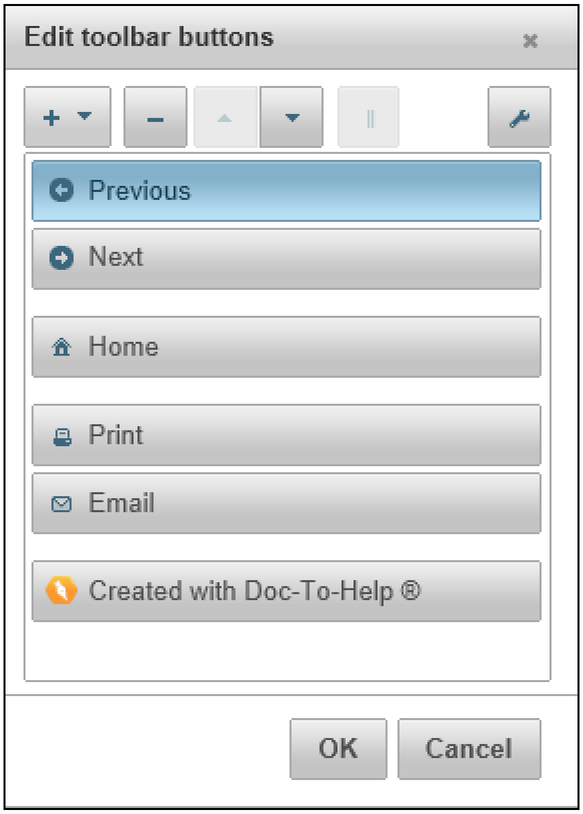
Click (for poweredBy and Custom buttons only) — The JavaScript code of the onclick event handler. This can be edited.

Toolbars (toolbars.left and toolbars.right)

The Toolbar settings control which buttons are included in your NetHelp 2.0 toolbar and how they are displayed. You can remove and rearrange the buttons, as well as add custom buttons. The alignment of the buttons can also be changed.

If you would like to change the NetHelp 2.0 toolbar button labels, icons, and other options, edit the **Buttons** fields.

Content — This field is used to add, remove, and rearrange the buttons in the NetHelp 2.0 toolbar. Click the button to open the **Edit toolbar buttons** dialog box. Buttons include Previous, Next, Print, Email, Twitter, Facebook, Favorites, Collapsed, and Side Panel.



To edit a button: select it, then click on the Wrench button to open that button's Properties for editing.

To delete a button: select it, then click the **–** button. If you remove a default button (**Previous**, **Next**, **Home**, **Print**, **Email**, **Powered By**, etc.) you can easily add it back by clicking the + button You cannot remove buttons that were defined in other Settings files. See Using the Settings List Editor for Localization and more on page 217.

To add a custom button: click the + button.

To rearrange the buttons: use the **up** and **down arrows**.

To start a new button group: click the **Start Group of Buttons** button.

Page Header (pageHeader)

Visible — If selected, the header area will be displayed.

Height — Adjusts the height of the header, in pixels.

Show Text — If selected, the project Title will displayed in the header.

Topic

Swatch — The swatch chosen here will controls the color scheme for all the elements of the Topic area, unless you change each individually.

Text width — Used to manually set the width of the text (in pixels) in the topic window. By default, it is set to "0", which means the text will adjust to fill the window.

Table mode — The display of tables in a responsive Target is important. This setting determines how ALL tables will be formatted in the target. This setting has four possible values: None, ColumnToggle, Reflow, and Auto.

- **None** means the table display will not be modified.
- **ColumnToggle** means there will be a Columns button above each table that will show a list with columns and allow users to select the ones they want to display.
- **Reflow** means that tables will display in a special 2 column view when the window width isn't wide enough. In this two column view columns become rows so a single row from the initial table will be displayed as a record with multiple rows.
- **Auto** means that on large screens tables will be displayed as-is (without any modifications), but on smaller screens all widths will be reset to autosize determined by the content.

Strings

Not found text — The message displayed when a topic cannot be loaded. Default: "Topic not found"

Not found title — This window title displayed when a topic cannot be loaded. Default: "Topic not found"

Related topics

You can set **Swatch**, **Mini**, and **Corner** settings for Related Topics.

Not found title — This window title displayed when a topic cannot be loaded. Default: "Topic not found"

Icons— Choose the icon that will appear before each Subtopic Link. There are options for topic, collapsed, and expanded.

Breadcrumbs

Breadcrumbs are the trail of topics at the top of the Topic area.

Visible — If selected, breadcrumbs will be displayed.

You can set **Swatch**, **Mini**, and **Corner** settings for Breadcrumbs.

Include current topic — If selected, the current topic will be displayed in the breadcrumbs.

Label — The text label that will be shown before the breadcrumbs. Default: no label

Separator — The character or text displayed between breadcrumb items. Default: "/"

Icons— Choose the icon that will appear before each Breadcrumb. There are options for topic, collapsed, and expanded.

Collapsible section

You can set **Swatch**, **Mini**, and **Corner** settings for Collapsible Sections.

Icon position — Choose the position of the icon on the button.

Icons— Choose the icon that will appear before each section. There are options for collapsed and expanded.

The button labels and icons for the **Expand All** and **Collapse All** buttons in the topic. See Creating an Expanding/Collapsing Section on page 316 for information about creating expanding/collapsing sections.

- Default label for the **Expand All** button: "Expand All"
- Default label for the **Collapse All** button: "Collapse All"

For all four buttons, you can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Spinner

Message text — The text that displays in the spinner animation while the topic is loading. Default: "Loading …"

Popup section

You can set **Swatch** and **Corner** settings for the Popup Sections.

TOC

Label — The text displayed on the Table of Contents. Default: "Contents"

Swatch — The swatch chosen here controls the color scheme for all the elements of the Contents (TOC) area, unless you change each individually.

Icons — The icons displayed for a **Topic**, **Closed book**, or the **Tab icon**. You can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

backButton

Label — The tooltip for the Back Button.

You can set **Swatch** and **Corner** settings for the Popup Sections.

Icon — The icon displayed on the button. You can choose an alternate icon, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display. There are more than 300 vector-based FontAwesome icons available for your use, as well as others.

Index

Label — The text displayed on the Index. Default: "Index"

Swatch — You can set the **Swatch** for the Index.

Visible — If selected, the Index is displayed.

Hide empty — If selected, the Index will automatically be hidden if there are no index keywords in your project.

Icons

Item icon — The icon displayed next to each item in the Index (by default, no icon is displayed).

Tab icon — The icon displayed on the Index tab.

Strings

Filter tooltip — The tooltip displayed for the Index text box. Default: "Filter keywords"

Not found message — The message shown when no keywords are found. Default: "No keywords found"

Found message — The message shown when one or more keywords are found. It can contain a {count} macro that is replaced with the number of keywords found. Default: "#{count} keywords found"

More text — The text for the link that displays if the keywords don't fit on the screen and aren't scrolled. Default: "More…"

Search

Label — The text displayed on the Search tab. Default: "Search"

Swatch — You can set the **Swatch** for the Search pane.

Visible — If selected, the Search tab is displayed.

Icons

Item icon — The icon displayed next to each item in the Search (by default, no icon is displayed).

Tab icon — The icon displayed on the Search tab.

You can choose an alternate icon, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display. There are more than 300 vector-based FontAwesome icons available for your use, as well as others.

Strings

Filter tooltip — The Tooltip for the Search text box. Default: "Search topics"

Filter text — Default: "Search…"

Help message — This text will be shown in a popup when the user clicks the "Help" button on the Search tab. The logical operators will be replaced with the ones defined in the Operators fields of the Search properties. Default: "You can use logical operators #{and}, #{or}, #{not} Example: football #{or} hockey, sports #{and} #{not} baseball"

Loading message — This message is shown when the data is loading. Default: "Loading Search Engine…"

Error message — This message is shown if the data fails to load. Default: "Error: Search engine failed to load"

Disabled message — This message is shown when search is disabled. Default: "Search is disabled"

Not found message — This message is shown when no keywords are found. Default: "No topics found"

Found message — This message is shown when one or more topics are found. It can contain a {count} macro that is replaced with the number of topics found. Default: "# {count} topic(s) found."

Correcting message — This message is shown when the search engine suggests a search string correction to the user. It can contain a #{query} macro that is replaced with a link to the corrected search string. Default: "Did you mean #{query}?"

More text — The text for the link that displays if the found topics don't fit on the screen and aren't scrolled. Default: "More…"

Operators

Operators — The text name of the search operators "AND" , "NOT", and "OR".

SidePanel

Swatch — The swatch chosen here controls the color scheme for all the elements of the Side Panel area, unless you change each individually.

Width — The default width of the Side Panel (in pixels).

Collapsed — If selected, the entire Side Panel will be collapsed by default.

Side panel header

Visible — If selected the Side Panel Header is visible. (This is the default.)

Swatch — You can set the **Swatch** for the Side Panel Header pane.

Height —Adjusts the height of the Side Panel Header, in pixels.

Logo Image — You can choose a logo or other image that will display in the Side Panel Header.

Favorites

Label — The text displayed on the Favorites tab. Default: "Favorites"

Swatch — You can set the **Swatch** for the Favorites pane.

Visible — If selected, the Favorites tab is displayed.

Icons

Delete icon — The icon displayed on the Delete button.

Tab icon — The icon displayed on the Favorites tab.

You can choose an alternate icon, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display. There are more than 300 vector-based FontAwesome icons available for your use, as well as others.

Strings

Filter text — This message is shown when Favorites filter text box is empty. Default: "Filter favorites…"

Delete — The tooltip text for the Delete button.

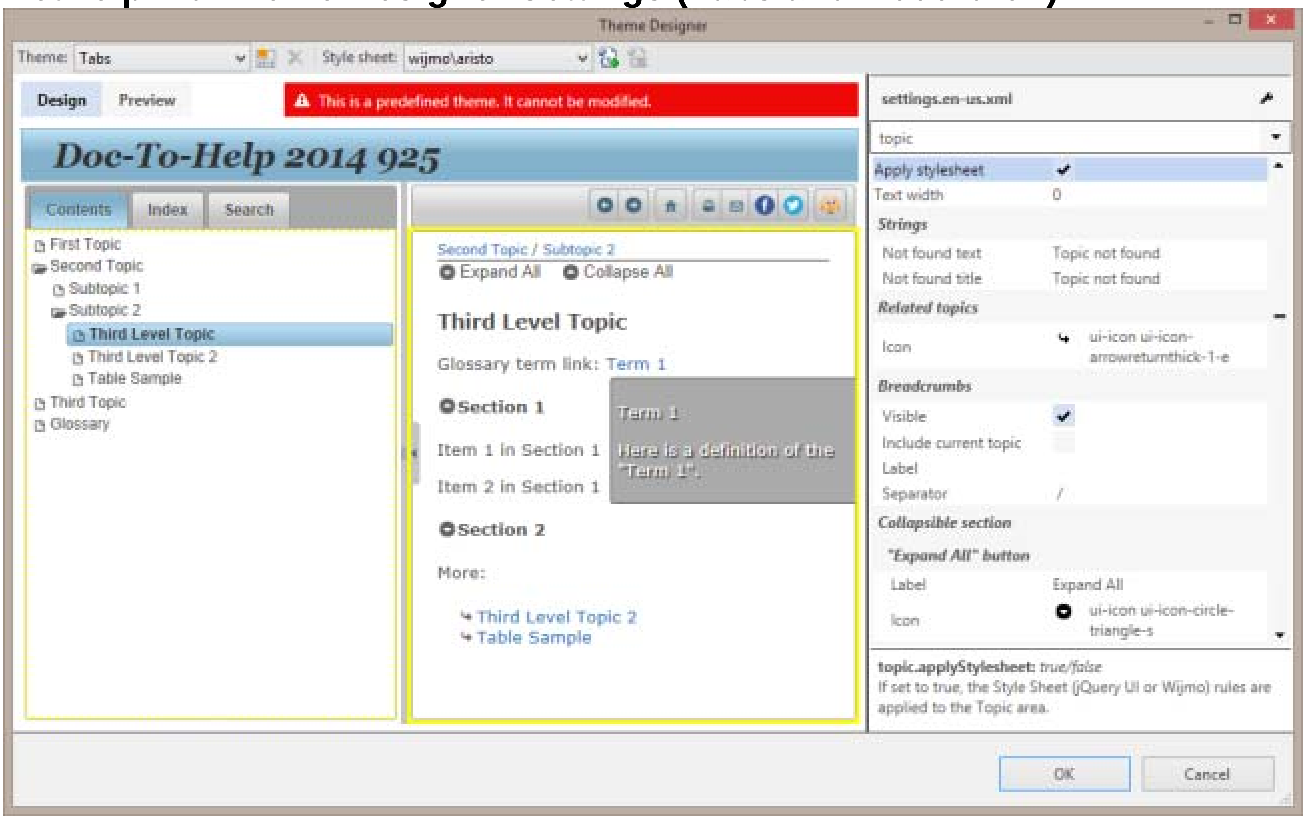
NavBar

Visible — If selected, the Navigation Bar is displayed. (This is the set of tabs at the bottom of the Side Panel — Contents, Index, and Favorites.)

Swatch — You can set the **Swatch** for the Navigation Bar.

Icon position — Choose the position of the icon on the button **—** left, right, top, or bottom.

NetHelp 2.0 Theme Designer Settings (Tabs and Accordion)



The Theme Designer settings can all be edited in the Settings area on the right side of the Theme Designer. To find the field you'd like to edit quickly, simply select it in the area on the left. The Settings area will display the field you need to edit. For global settings (General, Context Sensitive, Toolbar, and Accessibility), use the drop-down to display those fields.

General (general)

The General settings are global settings for this Theme.

Right-To-Left — If selected, the Theme text direction will be set for right-to-left languages. If you choose the Hebrew configuration file (settings.he-il.xml) for this Theme, this check box will be selected automatically.

Update title — Switches the header title displayed on the browser tab from the Topic title to the title of the project. If you would like the browser tab to remain static, rather than changing as the user navigates the help, select this check box.

Fav icon — Specifies which icon will appear in the browser when your users mark your NetHelp as a "Favorite."

Context Sensitive (contextSensitive)

The Context Sensitive settings are global settings for this Theme.

Strings

Title — The title used in the browser window title when a list of keywords, groups, or search results is shown on the page. This string can contain macros #{key} and #{value} that are replaced with the context-key ("keyword", "group", "id", or "search") and context-value. Default: "Topics for #{key}: #{value}"

Not found message — This message is shown when the result is empty (when the context-key is "keyword", "group", or "id"). Default: "No topics found"

Not supported message — This message is shown when the specified context-key is not supported. Default: "The key "#{key}" is not supported"

Accessibility (accessibility)

The Accessibility settings are global settings for this Theme. See Creating Section 508 Compliant Help on page 19 for more information about Doc-To-Help's accessibility features.

Link menu header — This text is read by the accessibility device if a keyword or group link has more than one topic to display. Default: "#{count} topics found"

Contents

Closed book — 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. Default: "Closed book without topic"

Open book — 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. Default: "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. Default: "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. Default: "Open book with topic"

Topic — This text is read by the accessibility device when you mouse over a topic. Default: "Topic"

Collapsible sections

Collapsed section — This text is read by the accessibility device for the collapsible section icons. Default: "Click to expand"

Expanded section — This text is read by the accessibility device for the expanded section icons. Default: "Click to collapse

Buttons (buttons)

The Button settings control the label, icon, behavior, and text displayed on the buttons of the NetHelp 2.0 toolbar. The default buttons included in the NetHelp 2.0 toolbar are: **Previous**, **Next**, **Home**, **Print**, **Email**, and **Powered By**.

If you would like to remove, rearrange, change the alignment, or add a custom button to the NetHelp 2.0 toolbar, use the **Toolbar** (toolbar.topicTop) on page 213 fields.

Label — The name that will be used for the label and tooltip text of the button.

- **prev** — Default: “Previous”.
- **next** — Default: “Next”
- **home** — Default: “Home”
- **print** — Default: “Print”
- **email** — Default: “Email” The email address used for this button is support@mycompany.com by default. This should be changed in the **Help Targets** dialog box.
- **twitter** — Default: “Twitter”
- **facebook** — Default: “Facebook”
- **poweredBy** — Default: “Created with Doc-To-Help” The button also includes a default click action that opens the Doc-To-Help website. This click action can be edited.

Icon — The icon displayed on the button. You can choose an alternate icon, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Show Label — If selected, the label text will be displayed on the button.

Click (for poweredBy and Custom buttons only) — JavaScript code of the onclick event handler.

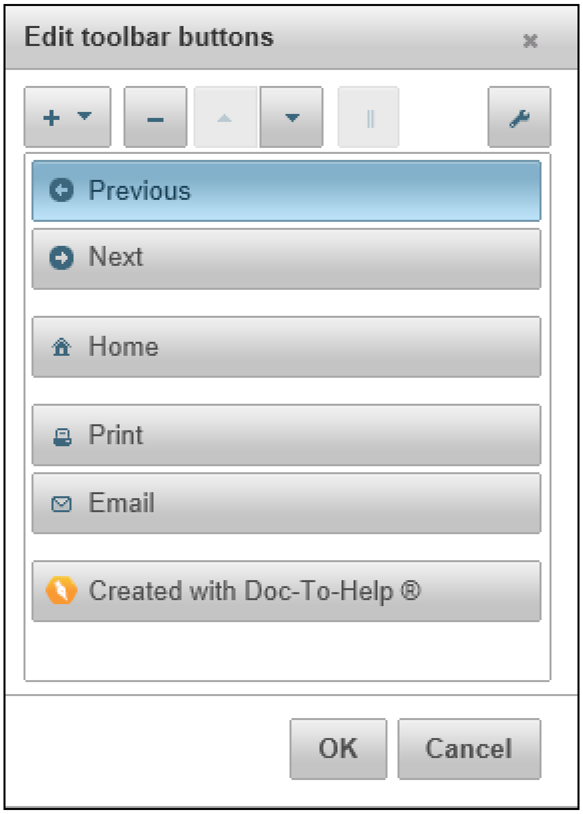
Toolbar (toolbar.topicTop)

The Toolbar settings control which buttons are included in your NetHelp 2.0 toolbar and how they are displayed. You can remove and rearrange the buttons, as well as add custom buttons. The alignment of the buttons can also be changed.

If you would like to change the NetHelp 2.0 toolbar button labels, icons, and other options, edit the **Buttons** fields.

Align — Controls the alignment of the NetHelp 2.0 toolbar buttons. By default, they are aligned right, but can be changed to left or center.

Content — This field is used to add, remove, and rearrange the buttons in the NetHelp 2.0 toolbar. Click the Wrench button in this field to open the **Edit Toolbar Buttons** dialog box.



To edit a button: select it, then click on the Wrench button to open that button's Properties for editing.

To delete a button: select it, then click the **–** button. If you remove a default button (**Previous**, **Next**, **Home**, **Print**, **Email**, or **Powered By**) you can easily add it back by clicking the + button You cannot remove buttons that were defined in other Settings files. See Using the Settings List Editor for Localization and more on page 217.

To add a custom button: click the **+** button.

To rearrange the buttons: use the **up** and **down arrows**.

To start a new button group: click the **Start Group of Buttons** button.

Page Header (pageHeader)

Visible — If selected, the header area will be visible.

Height — Adjusts the height of the header, in pixels.

Logo Image — Inserts an image in the upper left of the header. The image will be copied to your Theme folder \\My Doc-To-Help Projects\Doc-To-Help\Themes\NetHelp 2.0\(Name of Theme)\images

Show Text — If selected, the project Title will displayed in the header.

Topic

Apply stylesheet — If selected, the jQuery or Wijmo style sheet rules are applied in the Topic area.

Text width — Used to manually set the width of the text (in pixels) in the topic window. By default, it is set to “0”, which means the text will adjust to fill the window.

Strings

Not found text — The message displayed when a topic cannot be loaded. Default: “Topic not found”

Not found title — This window title displayed when a topic cannot be loaded. Default: “Topic not found”

Related topics

Icon — Choose the icon that will appear before each Subtopic Link.

Breadcrumbs

Breadcrumbs are the trail of topics at the top of the Topic area.
Visible — If selected, breadcrumbs will be displayed.
Include current topic — If selected, the current topic will be displayed in the breadcrumbs.
Label — The text label that will be shown before the breadcrumbs. Default: no label
Separator — The character or text displayed between breadcrumb items. Default: “/”

Collapsible section

The button labels and icons for the **Expand All**, **Collapse All**, **Expanded Section**, and **Collapsed Section** buttons in the topic. See Creating an Expanding/Collapsing Section on page 316 for information about creating expanding/collapsing sections.

- Default label for the **Expand All** button: “Expand All”
- Default label for the **Collapse All** button: “Collapse All”

For all four buttons, you can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Spinner

Message text — The text that displays in the spinner animation while the topic is loading. Default: “Loading …”

TOC

Label — The text displayed on the TOC tab or accordion button. Default: “Contents”

Icons — The icons displayed in the Table of Contents for a **Topic**, **Closed book**, or **Open book**. You can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Index

Label — The text displayed on the Index tab or accordion button. Default: “Index”

Visible — If selected, the Index is displayed.

Hide empty — If selected, the Index will automatically be hidden if there are no index keywords in your project.

Strings

Filter tooltip — The tooltip displayed for the Index text box. Default: "Filter keywords"

Not found message — The message shown when no keywords are found. Default: "No keywords found"

Found message — The message shown when one or more keywords are found. It can contain a {count} macro that is replaced with the number of keywords found. Default: "#{count} keywords found"

More text — The text for the link that displays if the keywords don't fit on the screen and aren't scrolled. Default: "More…"

Search

Label — The text displayed on the Search tab or accordion button. Default: "Search"

Visible — If selected, the Search tab or accordion button is displayed.

Strings

Filter tooltip — The Tooltip for the Search text box. Default: "Search topics"

Help message — This text will be shown in a popup when the user clicks the "Help" button on the Search tab. The logical operators will be replaced with the ones defined in the Operators fields of the Search properties. Default: "You can use logical operators #{and}, #{or}, #{not} Example: football #{or} hockey, sports #{and} #{not} baseball"

Loading message — This message is shown when the data is loading. Default: "Loading Search Engine…"

Error message — This message is shown if the data fails to load. Default: "Error: Search engine failed to load"

Disabled message — This message is shown when search is disabled. Default: "Search is disabled"

Not found message — This message is shown when no keywords are found. Default: "No topics found"

Found message — This message is shown when one or more topics are found. It can contain a {count} macro that is replaced with the number of topics found. Default: "# {count} topic(s) found."

Correcting message — This message is shown when the search engine suggests a search string correction to the user. It can contain a #{query} macro that is replaced with a link to the corrected search string. Default: "Did you mean #{query}?"

More text — The text for the link that displays if the found topics don't fit on the screen and aren't scrolled. Default: "More…"

Buttons

The tooltip labels and icons for the **Search**(go), **Help**, and **Highlight** buttons in the Search box.

- Default tooltip for the go button: "Search"
- Default tooltip for the help button: "Help"
- Default tooltip for the highlight button: "Highlight Search Hits"

You can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Operators

Operators — The text name of the search operators “AND” , “NOT”, and “OR”.

Splitter

Position — The initial position of the splitter, in pixels. Default: 300

Collapsed — If selected, the side panel (TOC/Index/Search) is hidden by default. The user can reopen the side panel by clicking the “Show side panel” icon.

Disabled — If selected, the splitter position cannot be changed by the end user.

Strings

Tooltip: show side panel — The tooltip displayed for the show side panel icon. Default: “Show side panel”

Tooltip: hide side panel — The tooltip displayed for the hide side panel icon. Default: “Hide side panel”

Icons

The icons displayed for the **Splitter**, **Show side panel**, and **Hide side panel**. You can choose alternate icons, upload your own (see Adding Custom Icons to a NetHelp 2.0 Theme on page 217), or have no icon display.

Adding Custom Icons to a NetHelp 2.0 Theme

All icons in a NetHelp 2.0 Theme can be changed. You can replace the default icon with a different jQuery or FontAwesome icon (FontAwesome icons available only in the Responsive Theme), upload your own custom icon images, or have no icon display.

To create a Custom Icon Set

1. Click on the icon you wish to change in the **Settings editor** of the Theme Designer. The **Choose Icon** dialog box will open.
2. Choose the **Standalone custom icon files** radio button. **Custom icon set #1** displays by default.
3. Choose the next empty icon set from the drop down (there are 20 available).
4. Click on the button next to the **Default state**. The **Open** dialog will open, where you can choose a .png file from your machine.
5. If desired, repeat for the **Focus state**, **Active state**, and **In content** icons. If you leave these empty, the image chosen for the **Default state** will be used for all states.
6. Click **OK**.

The custom images will automatically be copied to your Theme folder \\My Doc-To-Help Projects\Doc-To-Help\Themes\NetHelp 2.0\(Name of Theme)\images

Using the Settings List Editor for Localization and more

The **Settings List Editor** manages all of the settings for your Theme customizations and localization strings.

When you make Theme customizations — add a logo to the header, rearrange the toolbar buttons, etc. — those changes are saved to the **settings.xml** file in the **Theme Settings**. You can reuse this file in other Themes.

Doc-To-Help includes 14 localization files that can be used to change all of the text strings within a NetHelp 2.0 Theme. Once a configuration file is chosen, all of the strings will displayed within the Theme Designer and can be edited if desired.

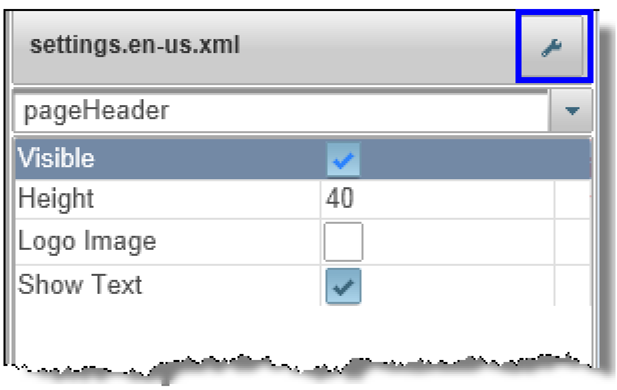
English = settings.en-us.xml
Danish = settings.da-dk.xml
German = settings.de-de.xml
Spanish = settings.es-es.xml
Norwegian = settings.nb-no.xml
Portuguese = settings.pt-pt.xml
Russian = settings.ru-ru.xml
Swedish = settings.sv-se.xml
Hebrew = settings.he-il.xml
Chinese = settings.zh-cn.xml
Japanese = settings.ja-jp.xml
Italian = settings.it-it.xml
French = settings.fr-fr.xml
Dutch = settings.nl-nl.xml

Doc-To-Help merges the Theme Settings and the Target Settings together when your Target is built.

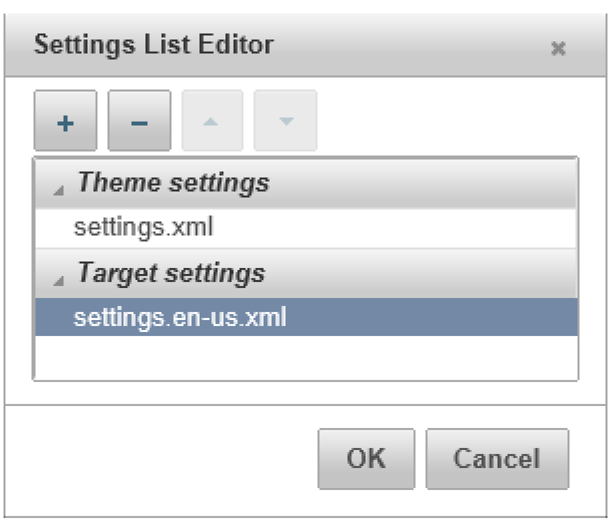
By saving the Theme settings separately from the Target settings, you can make your main customizations in one place (the Theme settings) and keep your localizations in another (the Target settings). Then, you can build different versions of the same theme in different languages. For example, you can add a logo to your Theme, rearrange the buttons, and hide the breadcrumbs. This will be saved to **settings.xml** in the Theme settings. Then you can add the French localization file (**settings.fr-fr.xml**) to your Target settings. You can then build an English version and a French version of the same Theme.

To manage Theme settings

1. Open the **Theme Designer** and click the Wrench button on the upper right.



The **Settings List Editor** will open.



The Settings List Editor has two sections, **Theme Settings** and **Target Settings**. This means you can manage the settings for your Theme and Target separately. (Plus, you can keep several Theme and Target settings in your list.)

Doc-To-Help merges all the files together when you build your Target.

2. To add or remove settings files, click in the appropriate area and use the + or – buttons. You can rearrange the files by using the up and down arrows. If you make changes to one of the default configuration files using the Theme Designer, a copy of that file will be created in your Theme's folder. This serves two purposes – first of all, the changes won't affect other Themes **—** plus, the edited settings file you create can be imported into other Themes.

   You can also create new files if you wish. After clicking the **+** button, choose the **Add New File** radio button in the **Add Settings** dialog box.

3. Select the settings file you would like to use and/or edit in the Theme Designer and click **OK**. That file name will display at the top of the **Settings** editor.

The **Settings List Editor** can store a "chain" of files for both your Theme and for the current Target. It merges the files from start to end, with the latest file overriding each previous file. If the Settings file you are editing is not the last in the chain, what you see in the Theme preview will not be the same as what you see in the Settings editor, because the Settings editor displays the settings up to the current point in the chain, while the Theme preview will reflect the entire chain. If you choose the last file in the chain, the Theme preview will match the Settings editor.

Edited configuration files are stored in the \\My Doc-To-Help Projects\Doc-To-Help\Themes\NetHelp 2.0\Themes\name of new theme folder. These files can be used in other Themes. Just copy them into the folder of the new Theme and add them using the **Settings List Editor**.

Creating and Adding New Style Sheets

A Style sheet is a jQuery or Wijmo theme that controls the appearance (colors, fonts, etc.) of your NetHelp 2.0 Theme.

Doc-To-Help includes 16 predefined style sheets, but you can download additional themes, or create your own, with the jQuery Theme Roller.

To add a new jQuery or jQuery Mobile theme

1. In the **Theme Designer**, click the **Add New Style Sheet** button. The **Add New Style Sheet** dialog box will open.

- If you already have a theme you would like to add, click the **Browse** button next to the **Browse to the file path here** field and navigate to your file.
- If you want to create or download a Theme for Tabs or Accordion, open the **jQuery UI Theme Roller** website at http://jqueryui.com/themeroller/. For the Responsive Theme, use the jQuery Mobile Theme Roller website at http://jquerymobile.com/themeroller/.

  In Doc-To-Help's **Add New Style Sheet** dialog box, click the **Browse** button next to the **Browse to the file path here** field and navigate to your file.

2. Click **Next**. The file name will display in the **Style Sheet name** field.
3. Click **Finish**. In the Theme Designer, the **Style sheet** field will automatically choose the new theme and display it.

After you add a style sheet using the **Add New Style sheet** dialog box, it will be stored on your machine here: \\My Doc-To-Help Projects\Doc-To-Help\Themes\Common\NetHelp 2.0\css\jquery-ui\name of new theme.

To learn more about using the jQuery UI Theme Roller, see http://jqueryui.com/docs/Theming/Themeroller. To learn more about the jQuery Mobile Theme Roller, see http://jquerymobile.com/resources/ .

Additional Theme Customizations

NetHelp 2.0 Themes are designed to be fully customized. You can change virtually any aspect of a theme and add whatever effects and widgets you can think of. Many customizations can be done with the **Theme Designer** (see Using the NetHelp 2.0 Theme Designer on page 201), but others can be done by manually editing files within the Theme.

Before starting a customization, create a new theme. See Customizing NetHelp 2.0 Themes on page 200.

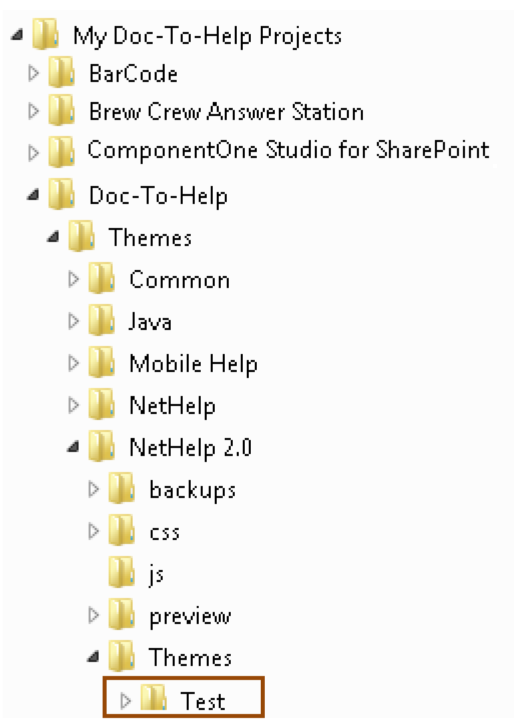
Manual customization is restricted to the following files (Tabs and Accordion Themes only):

- **layout.html** to change the theme layout.
- **user.css** to change styles in the theme.
- **user.js** to add new scripts to the theme.
- **settings*.xml**: This file should only be modified in an advanced scenario where you add new aspects to the theme and you want those aspects to be customizable by other users in the Theme Designer.

Other files in the theme should not be edited because Doc-To-Help will overwrite them during an upgrade. All changes should be restricted to the four files above.

Note: Customized Themes (which include the files above) are stored by default in the \\My Doc-To-Help Projects\Doc-To-Help\Themes\NetHelp 2.0\Themes\(name of theme) folder. **Make sure to edit the files stored in the folder for your custom theme and not one of the default themes.**

For example, if you create a custom NetHelp 2.0 Theme named "Test", this is where it will be located. Manually edit the files in this folder.



Examples: Changing the Style sheet

Tabs and Accordion Themes only.

Changing the header font

Change in **user.css:**

```
#c1headerText {
    margin: 0;
    font-family: Georgia, Times, "Times New Roman";
    font-family: Tahoma, Arial, sans-serif;
    font-style: italic;
}
```

Changing the header alignment

Add to **user.css**:

```
#c1headerPanel {
    width: 100%;
}
```

Change in **user.css**:

```
#c1headerText {
    margin: 0;
    font-family: Georgia, Times, "Times New Roman";
    font-style: italic;
    text-align: right;
    margin-right: 10px;
}
```

Changing the font of the left panel tabs

Add to **user.css**:

```
#c1sideTabsHeader li {
    font-weight: bold;
    font-style: italic;
}
```

Changing the Contents panel font

Add to **user.css**:

```
#c1toc li {
    font-family: Monotype Corsiva;
}
```

Changing the Index panel font

Add to **user.css**:

```
#c1index li {
    font-family: Monotype Corsiva;
}
```

Changing the Search panel font

Add to **user.css**:

```
#c1search li {
    font-family: Monotype Corsiva;
}
```

Changing the highlighting color of search hits

Add to **user.css**:

```
.search-highlight {
    background-color: Lime;
}
```

Changing the breadcrumbs font

Change in **user.css**:

```
#c1breadcrumbs {
    font-size: .9em;
    font-family: Courier New;
}
```

Removing the line under the breadcrumbs

Change in **user.css**:

```
#c1breadcrumbs {
    border: none;
}
```

Changing the link font

Change in **user.css:**

```
#c1topicPanel a[href],
#c1topicPanel a[data-ref],
.topic-popup a[href],
.topic-popup a[data-ref],
```

```
.aklinks-menu a {
    color: #1B75BB;
    text-decoration: none;
    font-style: italic;
}
```

Examples: Manual Customization

Tabs and Accordion Themes only.

Showing some text to the right of the header

Change in **layout.html**:

```
<table id="c1headerPanel" border="0" cellpadding="0" cellspacing="0"><tr
id="c1headerPanelRow0">
  <td id="c1headerLogoCell">
    <img id="c1headerLogo" src="images/d2h_logo_placeholder.png" alt="Logo"
class="logo" />
    </td><td id="c1headerTextCell">
      <h1 id="c1headerText"></h1>
    </td><td id="headerRightCell">
      <em>Powered by Doc-To-Help</em>
    </td>
  </tr></table>
```

Add to **user.css**:

```
#c1headerPanel {
    width: 100%;
}
#headerRightCell {
    text-align: right;
    padding-right: 10px;
}
```

Adding a footer

Add a <div> element to **layout.html:**

```
<div id="c1page" class="abs fill-h fill-v crop">
    <div id="c1header" class="abs fill-h crop ui-widget ui-state-hover">
    ....
```

```
    </div>
    <div id="c1main" class="abs fill-h">
    ....
    </div>
    <div id="mainFooter" class="crop ui-widget ui-state-hover">
        Footer text
    </div>
</div>
```

Add to **user.css**:

```
#mainFooter {
    position: absolute;
    bottom: 0;
    left: 0;
    right: 0;
    height: 30px;
}
#c1main {
    bottom: 32px;
}
```

Adding a header and footer to the topic panel

Add two <div> elements to HTML in **layout.html**:

```
<div id="c1topicPanelInner" class="abs fill-h fill-v scroll">
    <div id="topicHeader" class="ui-widget-content">
        Topic header text
    </div>
    <div id="c1topicBar">
        <div id="c1breadcrumbs" style="display:none;"></div>
        <div id="c1collapsiblePanel" style="display:none;">
            <span id="c1expandAll"><span class="icon"></span><a
class="label">Expand All</a></span>
            <span id="c1collapseAll"><span class="icon"></span><a
class="label">Collapse All</a></span>
        </div>
    </div>
    <div id="c1topic"></div>
    <div id="topicFooter" class="ui-widget-content">
        Topic footer text
```

```
    </div>
</div>
```

Add to **user.css**:

```
#topicHeader {
    margin: 5px 20px;
    border-width: 0 0 1px 0;
}
#topicFooter {
    margin: 25px 20px 5px 20px;
    border-width: 1px 0 0 0;
}
```

Adding nonscrolling header and footer into the topic panel

Add two <div> elements to HTML in **layout.html**:

```
<div id="c1topicPanel" class="content-topic abs fill-h ui-corner-all topiconly-container" style="display:none;" data-c1-role="topic">
    <div id="topicHeader" class="ui-widget-content">Header text</div>
    <div id="c1topicPanelInner" class="abs fill-h fill-v scroll">
        <div id="c1topicBar">
            <div id="c1breadcrumbs" style="display:none;"></div>
            <div id="c1collapsiblePanel" style="display:none;">
                <button id="c1expandAll" class="flat ui-widget-content">
                    <span id="c1expandAllIcon" class="icon"></span>
                    <span id="c1expandAllLabel" class="label">Expand All</span>
                </button>
                <button id="c1collapseAll" class="flat ui-widget-content">
                    <span id="c1collapseAllIcon" class="icon"></span>
                    <span id="c1collapseAllLabel" class="label">Collapse
All</span>
                </button>
            </div>
        </div>
        <div id="c1topic"></div>
    </div>
    <div id="topicFooter" class="ui-widget-content">Footer text</div>
</div>
```

Add to **user.css**:

```
#topicHeader {
    position: absolute;
```

```
    top: 0;
    left: 0;
    right: 0;
    height: 2em;
    border-width: 0 0 1px 0;
}
#topicFooter {
    position: absolute;
    bottom: 0;
    left: 0;
    right: 0;
    height: 2em;
    border-width: 1px 0 0 0;
}
#c1topicPanelInner {
    top: 2.2em;
    bottom: 2.2em;
}
```

Extending the toolbar to the left to the entire page width

Change in **layout.html**:

- Move the element `<div id="c1topBar">` from inside the element `<div id="c1content">` into the element `<div id="c1main">`.
- Create a new element `<div id="mainContent">` inside the element `<div id="c1main">`.

```
<div id="c1main" class="abs fill-h">
    <div id="c1topBar" class="ui-widget ui-widget-header ui-corner-all"
style="text-align: right;">
    ....
    </div>
    <div id="mainContent">
        <div id="c1side" class="abs fill-v"><div id="c1sideInner">
        ....
        </div>
        <div id="c1splitter" class="c1-splitter abs fill-v">
        ....
        </div>
        <div id="c1content" class="content abs fill-v"><div id="c1contentInner">
```

```
~~            <div id="c1topBar" class="ui-widget ui-widget-header ui-corner-all" style="text-align: right;">~~
~~            ....~~
~~            </div>~~
        ....
        </div>
    </div>
</div>
```

Add to **user.css**:

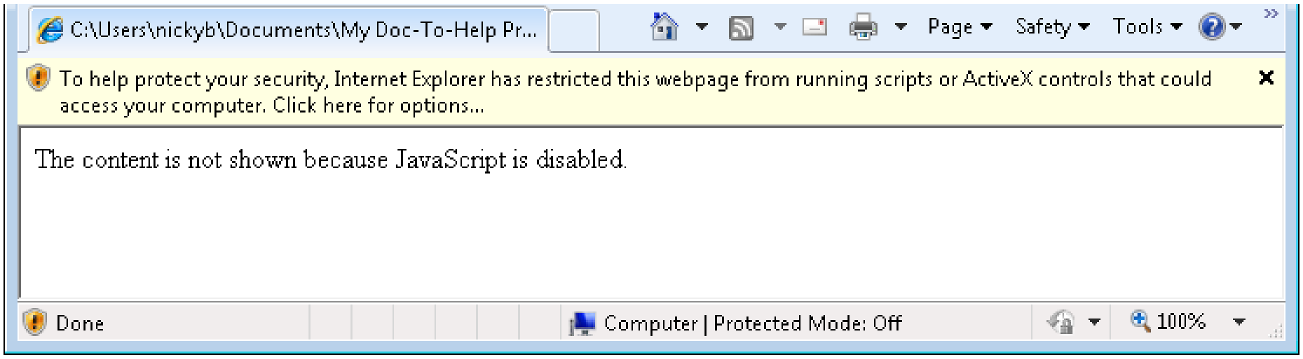
```
#c1topBar {
    margin: 0.2em;
}
#mainContent {
    position: absolute;
    top: 2.7em;
    bottom: 0;
    left: 0;
    right: 0;
}
#c1topicPanel {
    top: 0;
}
```

Theme Configuration FAQ

Here are the answers to some frequently asked questions about customizing NetHelp 2.0.

General

Why do I get the error message "The content is not shown because JavaScript is disabled" when I try to view my NetHelp 2.0 Targets?



NetHelp Targets that are opened locally (on your machine) may display this message, depending on your security settings. You could add the "Mark of the Web" (MOTW) to your project using the "Mark of the Web" check box in the **Help Targets** dialog box, see NetHelp Target for more information about MOTW.

What happens when I upgrade my project from NetHelp Classic to NetHelp 2.0?

Your project's theme will be set to the NetHelp 2.0 default **Tabs** theme. If you created a custom theme for NetHelp Classic, those customizations are lost. But it is easy to customize a NetHelp 2.0 theme, see Using the NetHelp 2.0 Theme Designer on page 201. If your NetHelp Classic project had context-sensitivity implemented, minor changes will need to be made when upgrading to NetHelp 2.0. See Context Sensitive Help in NetHelp for details.

When you switch between NetHelp Classic and NetHelp 2.0, the following properties will be reset: Theme, Localization, Background, Search Type, Dynamic Table of Contents, Frameset, Generate XHTML, XML Transformation, and Strings.

I had NetHelp projects and I upgraded them to NetHelp 2.0. Can I change them back to NetHelp Classic if I want to?

Yes, open the **Help Targets** dialog box and clear the **NetHelp 2.0 Target** check box. Click **OK** and build your Target.

When you switch between NetHelp Classic and NetHelp 2.0, the following properties will be reset: Theme, Localization, Background, Search Type, Dynamic Table of Contents, Frameset, Generate XHTML, XML Transformation, and Strings.

How do I change the position of items within a NetHelp 2.0 theme?

Some things are changed in the Theme Designer; some in the **layout.html** file. See Using the NetHelp 2.0 Theme Designer on page 201 and Additional Theme Customizations on page 220. (Tabs and Accordion Themes only.)

Can I incorporate Google Translate, Google Analytics, or Google Custom Search into NetHelp 2.0?

Yes, you can incorporate any, or all of them. See the following blog posts for details:

Instant Localization: Integrating Google Translate into NetHelp

Track Your Readership: Integrating Google Analytics into Your NetHelp

Integrating Google Custom Search into Your NetHelp Outputs

Header area

How do I add a logo to the header of a NetHelp 2.0 Theme?

In the Theme Designer, click on the header to open the **Page Header** settings and choose **logoImage** (Tabs and Accordion Themes only). For the Responsive Theme, **logoImage** is in the sidePanel settings. You can specify a logo there. It will appear in the upper left of your header. See NetHelp 2.0 Theme Designer Settings.

I want to change the font and color of the Title in my NetHelp 2.0 header, how do I do that?

You need to edit a section of the **user.css** file in your theme. This file can be found in the \\My Doc-To-Help Projects\Doc-To-Help\Themes\NetHelp 2.0\Themes\Name of Theme folder. (Tabs and Accordion Themes only.) Here is the section:

```
}
#c1headerTextCellTextCell h1{
                vertical-align: bottom;
                color:#ff9933;
                font-size:60x;
                font-family:"Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
                font-weight:normal;
                font-style:normal
}
```

How do I turn the buttons on the right — previous, next, home, print, email — on and off, or change the images?

In the Theme Designer, click on the button to open its settings.Click on the **Icon** button to change the image. To turn buttons on or off, or create new ones, click in the toolbar area to open the **Toolbar** settings. See NetHelp 2.0 Theme Designer Settings.

Can I add text to the previous, next, home, print, and email buttons?

In the Theme Designer, click on the button to open its settings. Select the **Show Label** check box to add a text label to the button.

How do I turn off the “Created with Doc-To-Help” button in my NetHelp 2.0 Theme?

In the Theme Designer, use the drop-down in the **Settings** area and choose **toolbars.topicTop** (Tabs and Accordion Themes only). Click the Wrench icon next **Content** and delete the “Created with Doc-To-Help” button off in the **Edit Toolbar Buttons** dialog box.

TOC panel

How do I change the Table of Contents, Index, and Search tabs to accordion buttons?

The default theme for NetHelp 2.0 is **Tabs**. You can choose the **Accordion** theme instead (or the Responsive theme). In the Theme Designer, click the **Theme** drop-down. Choose **Accordion**.

Can I remove the Index or Search tabs (or accordion buttons)?

Yes, in the Theme Designer, click on the Index or Search areas to open their settings. Clear the **Visible** check box. (Only the Tabs and Accordion themes have a Search tab.)

Can I change the icons used next to the topics in the Table of Contents?

Yes, in the Theme Designer, click on the Table of Contents to open its toc settings. Click on the appropriate icon to change the image or add a new one. See NetHelp 2.0 Theme Designer Settings.

Color scheme

I would like to change the color theming of my NetHelp 2.0 target.

In the Theme Designer, you can use the Style Sheet drop down to choose another style sheet. If you wish to create additional style sheets, click the **Add New Style Sheet** button. See Creating a New Style Sheet.

Icons

How to I change the “Related Topics” icon (the automatic subtopic links after “More” at the end of topics; by default an arrow icon)

In the Theme Designer, click in the topic area to open the Topic settings. Click on the **Related Topics** icon to change the image or add a new one. See NetHelp 2.0 Theme Designer Settings.

Localization

I would like to change the language used in my NetHelp 2.0 theme.

Doc-To-Help includes configuration files for 14 languages. See Using the Settings List Editor for Localization on page 217.

Customizing Eclipse Help and EPUB Themes

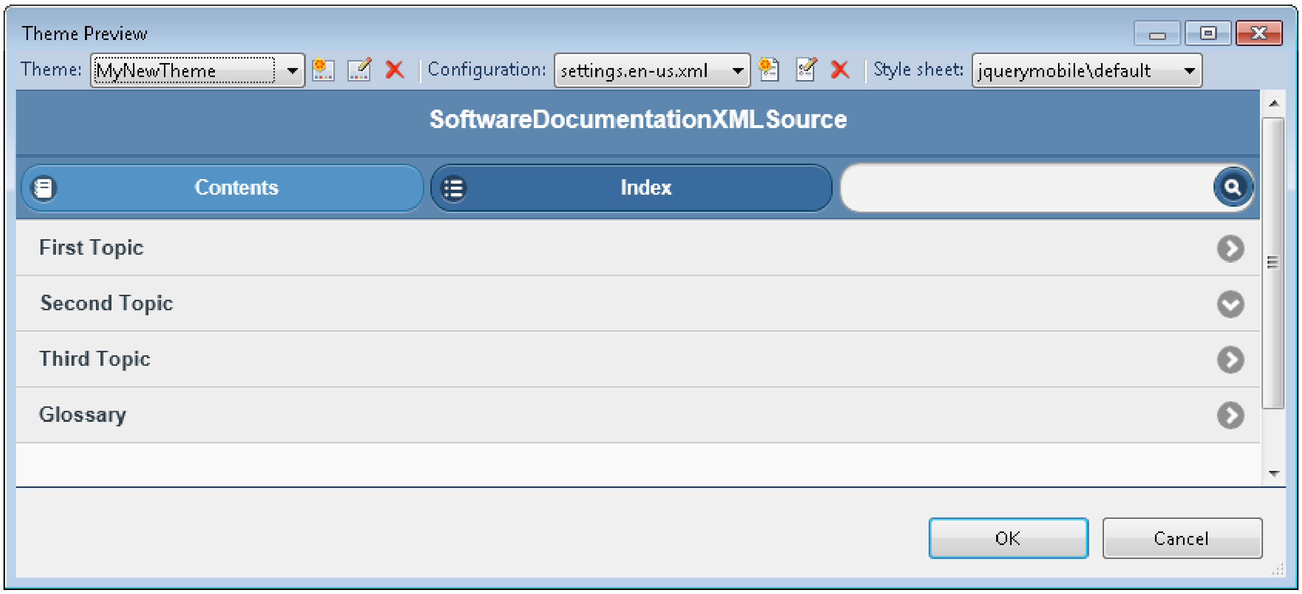
The following Themes are included with **Eclipse Help**, and **EPUB** Targets. These themes are the starting point for creating your customized Themes.

Target	Themes Available	Description
Eclipse Help	**Basic**	In Eclipse Help, the Target Theme only controls the topic frame, all other elements (TOC, Index, etc.) are controlled by the Eclipse Help System scripts.
EPUB	**Basic**	In EPUB, the Target Theme controls only items inside the frame, such as the option to display text Right-To-Left.

To create a new theme

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.**
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.

   You can now customize your Theme. See Using the Theme Editor on page 231.



Note: You can also create a new Theme by clicking the **Add New Theme** button in the **Theme Preview** dialog box.

For information on customizing:

- **NetHelp 2.0** Targets – see Customizing NetHelp 2.0 Themes on page 200.
- **NetHelp Classic**, **HTML Help**, **Microsoft Help Viewer**, **Microsoft Help 2.0**, or **JavaHelp** Targets – see Customizing NetHelp Classic, HTML Help, Microsoft Help Viewer, Microsoft Help 2.0, and JavaHelp Themes on page 235.

Using the Theme Editor

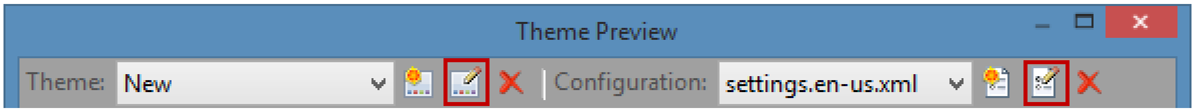
For Eclipse Help and EPUB Themes, the Theme Editor is used to view and edit **Themes** and **Configurations**, and to choose the **Style Sheet**. All three of these options work together to create the look, behavior, text, color scheme, accessibility options, and language you want for your Theme.

For other Targets, see Customizing NetHelp 2.0 Themes on page 200 and Customizing NetHelp Classic, HTML Help, Microsoft Help Viewer, Microsoft Help 2.0, and JavaHelp Themes on page 235.

To view or customize a theme or theme configuration

1. On the Home tab, click the **Theme** button. The **Theme Preview** dialog box will open. If you have chosen a default Theme or Theme Configuration, you can view them using the **View** buttons.

2. If you would like to create a custom Theme or configuration, create a new Theme first. See Adding a Theme or Theme Configuration on page 199 for instructions.
3. Click the **Edit Theme** or **Edit Configuration** buttons. A full list of options for each appears in Theme Options on page 232 and Configuration Options on page 233.



To choose a Style Sheet (Eclipse Help only)

1. Open the **Theme Preview** dialog box.
2. Select a style sheet using the **Style sheet** drop down. Doc-To-Help includes style sheets for Eclipse Help, and you can create more using the jQuery and Wijmo theme rollers.

jQuery UI themes can be downloaded from and created at http://jqueryui.com/themeroller/

Wijmo themes can be downloaded from http://wijmo.com/theming/

Copy the new themes to:

- jQuery: \\My Doc-To-Help Projects\Doc-To-Help\Themes\Common\(type of theme)\css\jquery-ui
- Wijmo: \\My Doc-To-Help Projects\Doc-To-Help\Themes\Common\(type of theme)\css\wijmo

Note: Customized Themes and Configurations are stored by default in the \\My Doc-To-Help Projects\Doc-To-Help\Themes\ folder.

Theme options

These options are available by clicking the **Edit Theme** button and are used to turn specific items in the Theme on and off (the header, plus tabs and buttons), to change icons in the navigation panes (TOC, Index, Search), to insert a logo in the header, and to change the icons for related topics and expanding/collapsing sections.

theme
The **name**, **baseTheme**, and **baseVersion** of the Theme. These fields are read-only for default themes.

general
updateTitle – Switches the header title displayed on the browser tab from the Topic title to the project title.

header
You can modify or even turn off the header at the top of the Theme.

visible — Turns the header on or off.

topic
You can change the icons used for automatic subtopic links and expanding/collapsing sections.

applyStylesheet (Eclipse Help only) — Set to true by default.

relatedTopics — Choose the icon that will appear before Subtopic Links.

collapsibleSections (Eclipse Help only) — Choose the expand all, collapse all, expanded, and collapsed icons for expanding/collapsing sections. See Creating an Expanding/Collapsing Section on page 316.

toc
You can change the icons for the Table of Contents.

icons — Choose the **leaf** (topic with no subtopics), **expanded**, and **collapsed** icons for the topics in the table of contents. You can also select the **No icon** option.

search
You can turn the **Search** tab or accordion button on or off, as well as change the icons for the Search features.

buttons — Choose the icons for **go** (begin search), **help**, and **highlight** (turns search highlight on/off).

buttons
You can turn the buttons in the Theme on or off, change the button icons, as well as display text on the buttons.

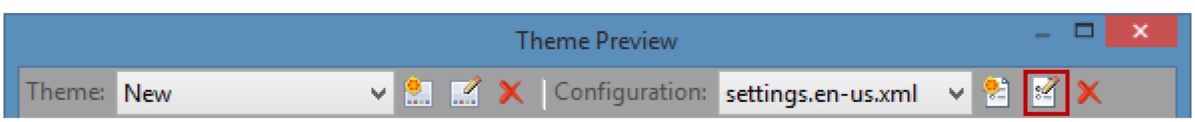
The button labels and tooltips can be edited using the **Configuration Options**.

Each button has an **id** that matches the id element in layout.html.

prev — Turn the Previous button on or off, choose the icon, and display button text (showLabel).

next — Turn the Next button on or off, choose the icon, and display button text (showLabel).

Configuration (language) options



These options are available by clicking the **Edit Configuration** button and are used to edit the language strings used in the Theme. Doc-To-Help includes configuration files for thirteen languages. There are also options for right-to-left languages (under **general**) and accessibility strings.

Configuration files
English = settings.en-us.xml
Danish = settings.da-dk.xml
German = settings.de-de.xml

Spanish = settings.es-es.xml
Norwegian = settings.nb-no.xml
Portuguese = settings.pt-pt.xml
Russian = settings.ru-ru.xml
Swedish = settings.sv-se.xml
Hebrew = settings.he-il.xml
Chinese = settings.zh-cn.xml
Japanese = settings.ja-jp.xml
Italian = settings.it-it.xml
French = settings.fr-fr.xml
Dutch = settings.nl-nl.xml

general

rightToLeft — The Theme text direction will be set for right-to-left languages. If you choose the Hebrew configuration file (settings.he-il.xml) this check box will be selected automatically.

theme

The **name** of the Theme this configuration is used in.

strings

topicSpinnerText — The text that displays when the topic is loading.

splitter

strings — The tooltips for the splitter button: "Show Side Panel" and "Hide Side Panel."

topic

index (EPUB only) — The header label for the Index.(Value).

strings — "Topic Not Found" text (notfoundText, notfoundTitle).

collapsibleSections — The "Expand All" and "Collapse All" text displayed for expanding/collapsing all sections. See Creating an Expanding/Collapsing Section on page 316.

Index

The text displayed on the **Index** tab or accordion button. Also the text for the following:

- "Filter keywords" (filterTooltip) – Tooltip for the Index text box.
- "No keywords found" (notfound) – This message is shown when no keywords are found.
- "#{count} keywords found" (found). This message is shown when one or more keywords are found. It can contain a {count} macro that is replaced with the number of keywords found.
- "More…" (more), the text for the link that displays if the keywords don't fit on the screen and aren't scrolled.

Search

The text displayed within the **Search** tab or accordion button.

The text for the following:

- "Search topics" (filterTooltip) – Tooltip for the Search text box.
- "Logical Operators" (helpMessage) – This text will be shown in a popup when the user clicks the "Help" button on the Search tab.
- "Loading Search Engine" (loading) – This message is shown when the data is loading.
- "Search engine failed to load" (loaderror) – This message is shown if the data fails to load.
- "Search disabled" (disabled) – This message is shown when search is disabled.
- "No topics found" (notfound) – This message is shown when no keywords are found.
- "# {count} topic(s) found." (found). This message is shown when one or more topics are found. It can contain a {count} macro that is replaced with the number of topics found.
- "Did you mean #{query}?" (correcting) –This message is shown when the search engine suggests a search string correction to the user. It can contain a #{query} macro that is replaced with a link to the corrected search string.
- "More…" (more) – The text for the link that displays if the found topics don't fit on the screen and aren't scrolled.

The text for the "Go", "Search", "Help", and "Highlight Search Hits" button tooltips.

The name of the search operators "AND" , "NOT", and "OR".

buttons

These buttons and their labels can be turned on and off using the **Theme Options**.

Previous (prev) — The label text and tooltip text for the "Previous" button (label).

Next (next) — The label text and tooltip text for the "Next" button (label).

accessibility

These strings are only editable if the **Accessibility Mode** for this Target is set to **Section 508**. This is done in the **Help Targets** dialog box, in the Advanced section (**Home** tab > **Targets** ribbon group dialog box launcher.)

toc — The titles for all of the icons in the table of contents.

collapsibleSections — The titles for the collapsible section icons.

akLinksMenuHeader — The header for the list of topics (#{count} topics found) if a keyword or group link has more than one topic to display.

Customizing NetHelp Classic, HTML Help, Microsoft Help Viewer, Microsoft Help 2.0, and JavaHelp Themes

The following Themes are included with **NetHelp Classic**, **HTML Help**, **Microsoft Help Viewer**, **Microsoft Help 2.0**, and **JavaHelp** Targets. These themes are the starting point for creating your customized Themes.

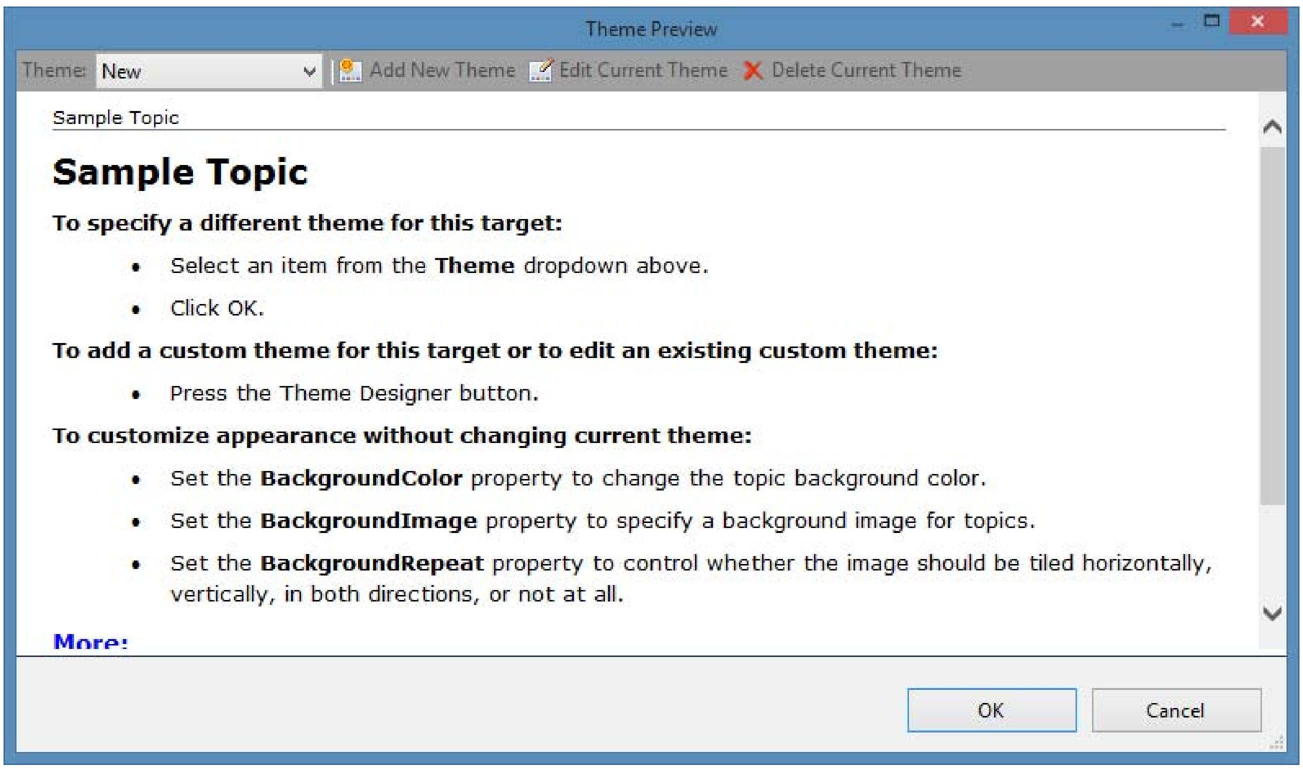
Target	Themes Available	Description
HTML Help or **Microsoft Help Viewer** or **Microsoft Help 2.0**	**Default**	• Tripane window with Contents, Index, Search tabs at left • Breadcrumbs above Help content
	Top Navigation	• Tripane window with Contents, Index, Search tabs at left • Previous/next icons 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 Classic	**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.
	Deutsche Themenvorlage (German)	Default theme with German text.
	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
	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

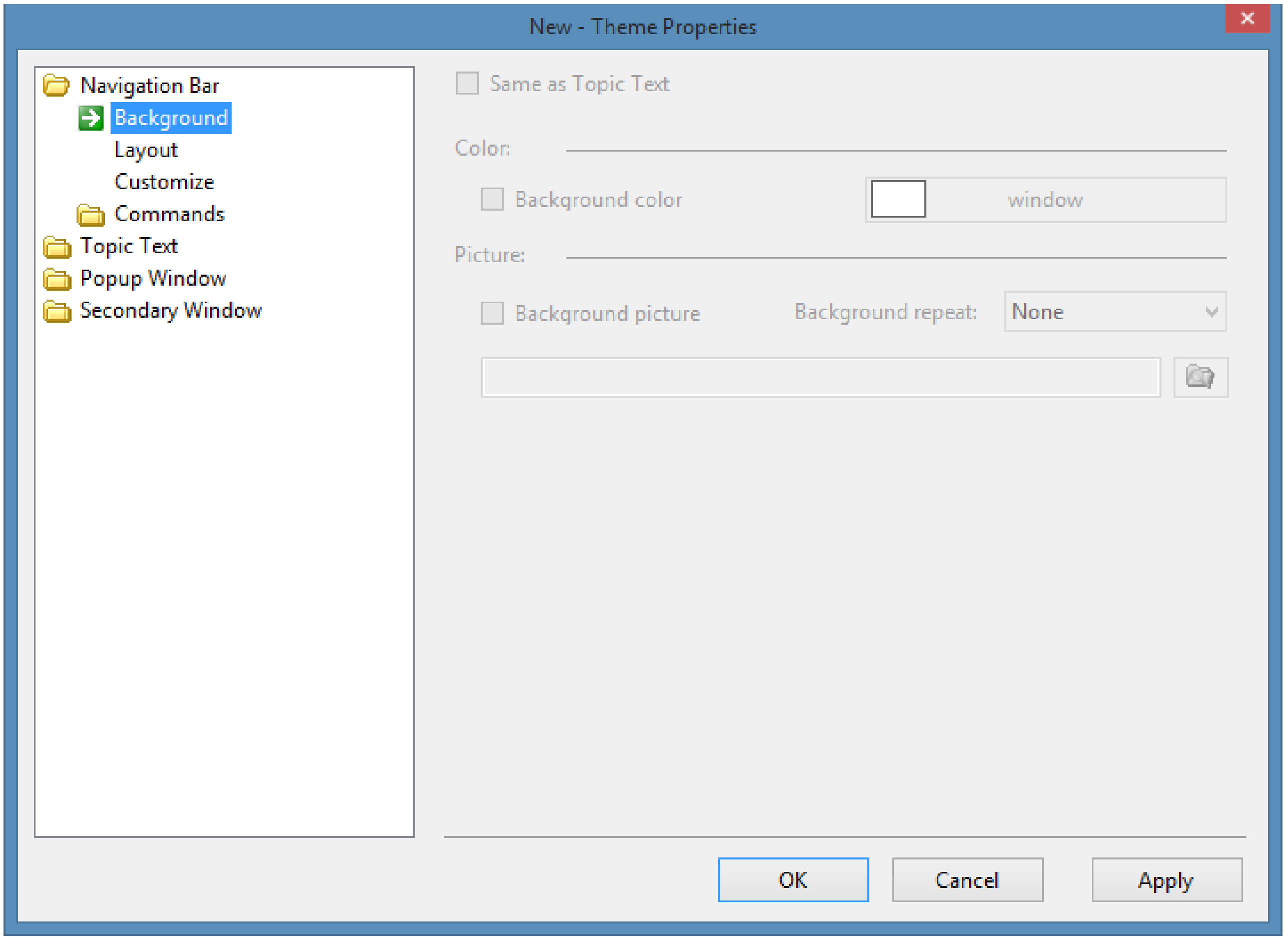
Note: To switch a **NetHelp Classic** Target to **NetHelp 2.0**, open the **Help Targets** dialog box and click the **NetHelp 2.0** check box.

To create a new theme and customize it

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.**
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.



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



5. Make your changes and click **OK**.

Note: Customized Themes are stored by default in the \\My Doc-To-Help Projects\Doc-To-Help\Themes folder. You can change this location using the Doc-To-Help **Options** dialog box, **Files** button. (To open the **Options** dialog box, choose the **File** tab > **Tools** > **Options**.)

For information on customizing:

- **NetHelp 2.0** Targets – see Customizing NetHelp 2.0 Themes on page 200.
- **Eclipse Help** and **EPUB** Targets – see Customizing Eclipse Help and EPUB Themes on page 230.

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

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

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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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.

Commands Fields

The **Previous** and **Next** controls are available for HTML Help, Microsoft Help Viewer 1.x, 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) **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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Buttons and Icons

Disabled picture — Defines the disabled image for the button. Note: Images should be stored in the following folder: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Buttons and Icons

General Properties (NetHelp Classic 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 Classic 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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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 163 — 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) 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 (HTML, Microsoft Help Viewer, Microsoft Help 2.0, JavaHelp, NetHelp Classic)

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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Backgrounds

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

Customize Fields

See Including Customized HTML Content in Themes.

Breadcrumbs Controls

Breadcrumbs are added by default to HTML Help, NetHelp, Microsoft Help Viewer 1.x, 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) **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 163 (**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 (HTML, MS Help Viewer 1.x, Help 2.0, JavaHelp, NetHelp Classic)

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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Backgrounds

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

Customize Fields

See Including Customized HTML Content in Themes.

Secondary Window Properties (HTML, MS Help Viewer 1.x, Help 2.0, JavaHelp, NetHelp Classic)

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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Backgrounds

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

Customize Fields

See Including Customized HTML Content in Themes.

Table of Contents Properties (NetHelp Classic 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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Buttons and Icons

Open book picture — Defines the opened image for the TOC. Note: Images should be stored in the following folder: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Buttons and Icons

Topic picture — Defines the topic image for the TOC. Note: Images should be stored in the following folder: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\Images\Buttons and Icons

Index Properties (NetHelp Classic 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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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 Classic 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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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 Classic 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 163.

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: \\My-Doc-To-Help Projects\DocToHelp\Themes\theme type\theme name\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 Classic 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 133 for more information.

Menu Properties (NetHelp Classic 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 Classic 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 (HTML, Microsoft Help Viewer, Microsoft Help 2.0, JavaHelp, NetHelp Classic)

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.

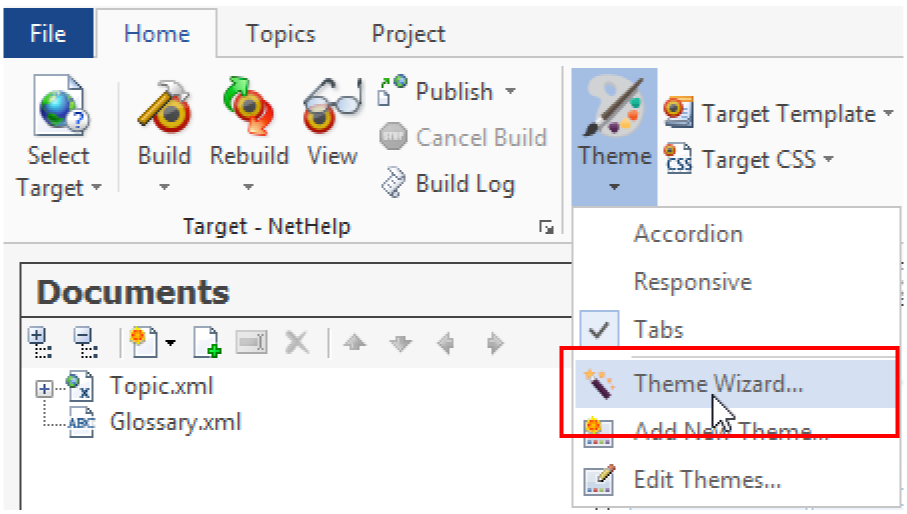
NetHelp Theme Wizard

The NetHelp Theme Wizard takes you step-by-step through the process of creating your own custom theme. It allows you to:

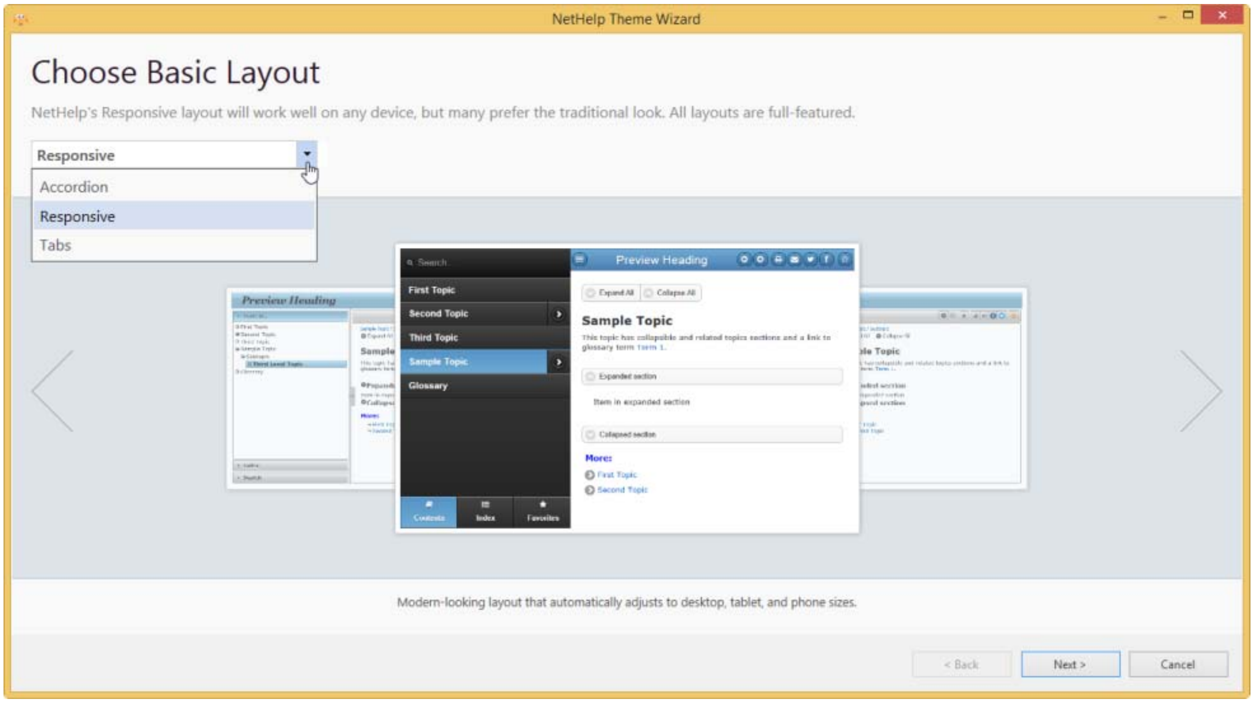
- create a new or update an existing theme,
- specify a layout and style sheet,
- brand your output, and
- set some of the most commonly used properties for the NetHelp target.

To use the wizard, follow these steps:

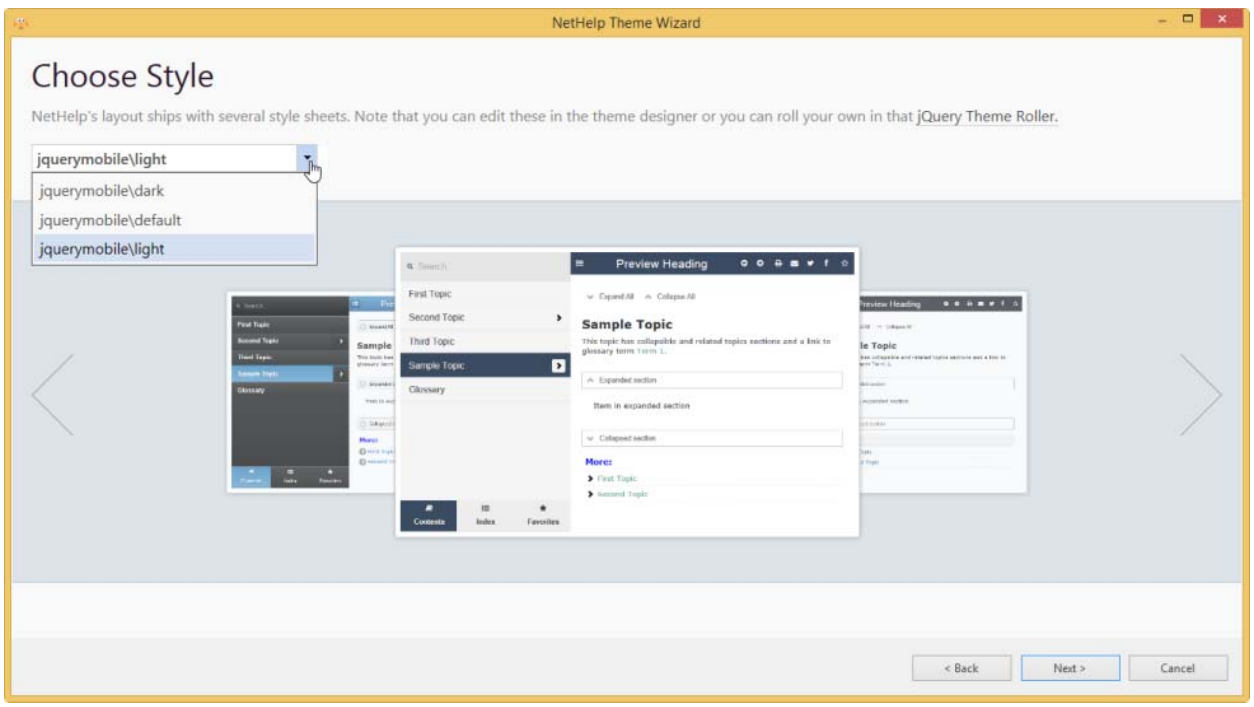
1. On the Doc-To-Help **Home** tab, click **Select Target** and choose a NetHelp target.
2. Click the **Theme button** drop-down arrow and select **Theme Wizard**. If you click the paintbrush and paint palette icon in the top of the **Theme** button, this also opens the wizard.



3. In the first window, choose a layout for the theme by selecting *Accordion*, *Responsive*, or *Tabs* from the drop-down text box. A preview of the layout appears below the box, or you can use the arrows on the left and right side of the window to scroll and choose a layout.



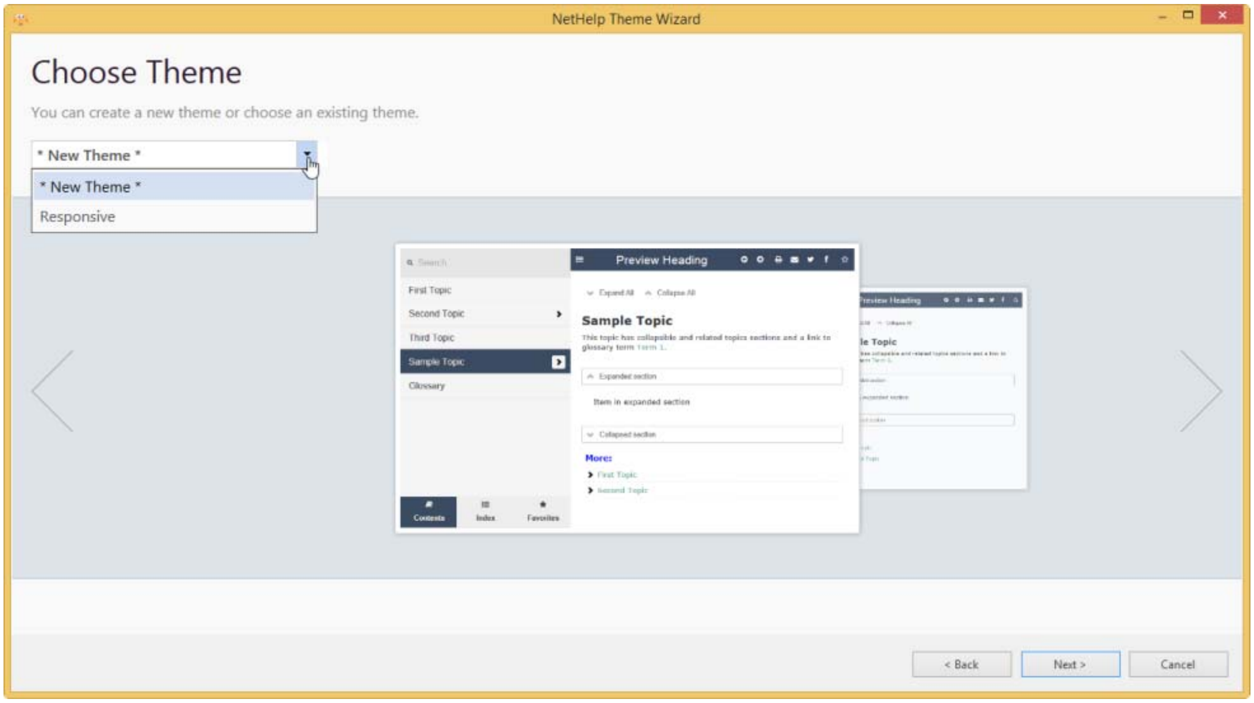
4. Click **Next**.
5. Choose a style sheet from the drop-down box. A preview of the style sheet appears below the box, or you can use the arrows on the left and right side of the window to scroll and choose a style sheet.



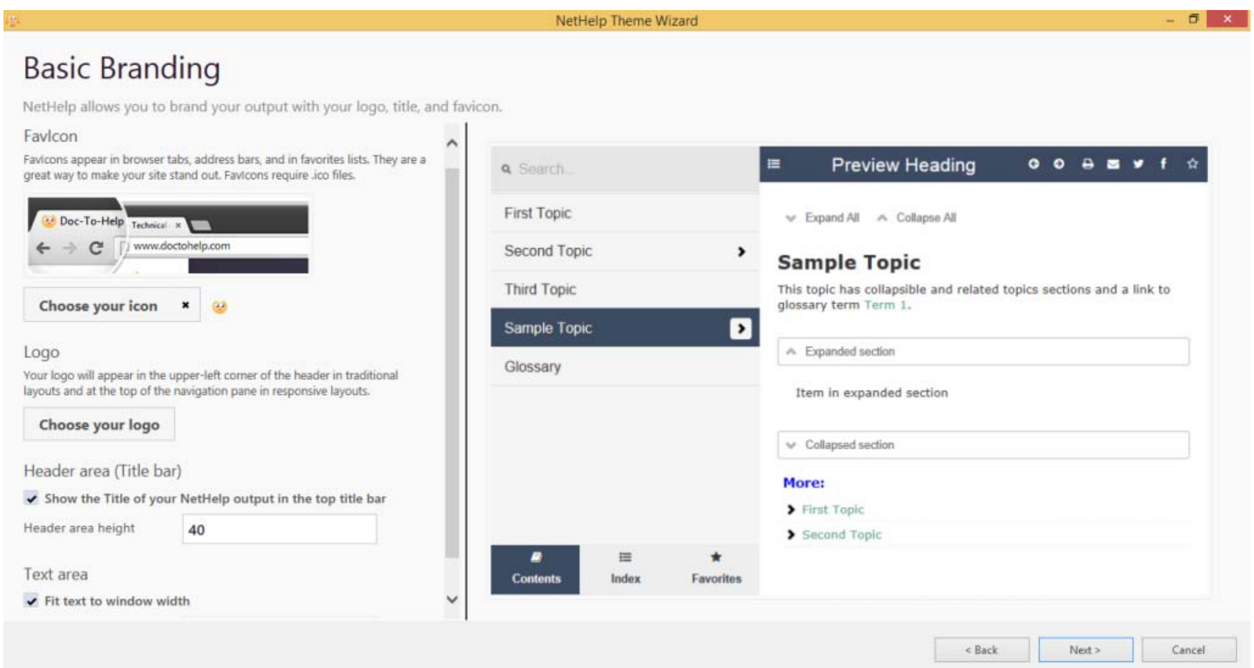
Note: You can edit the style sheet in the theme designer later or you can roll your own with the jQuery Theme Roller.

6. Click **Next**.

7. Choose ***New Theme*** to create a new theme or select an existing theme from the drop-down box.



8. Click **Next**.
9. In the **Basic Branding** window, select a FavIcon and logo, specify whether to show the title in the output, and customize the text area of your output. A preview of your output appears on the right side of the window as you set the properties.

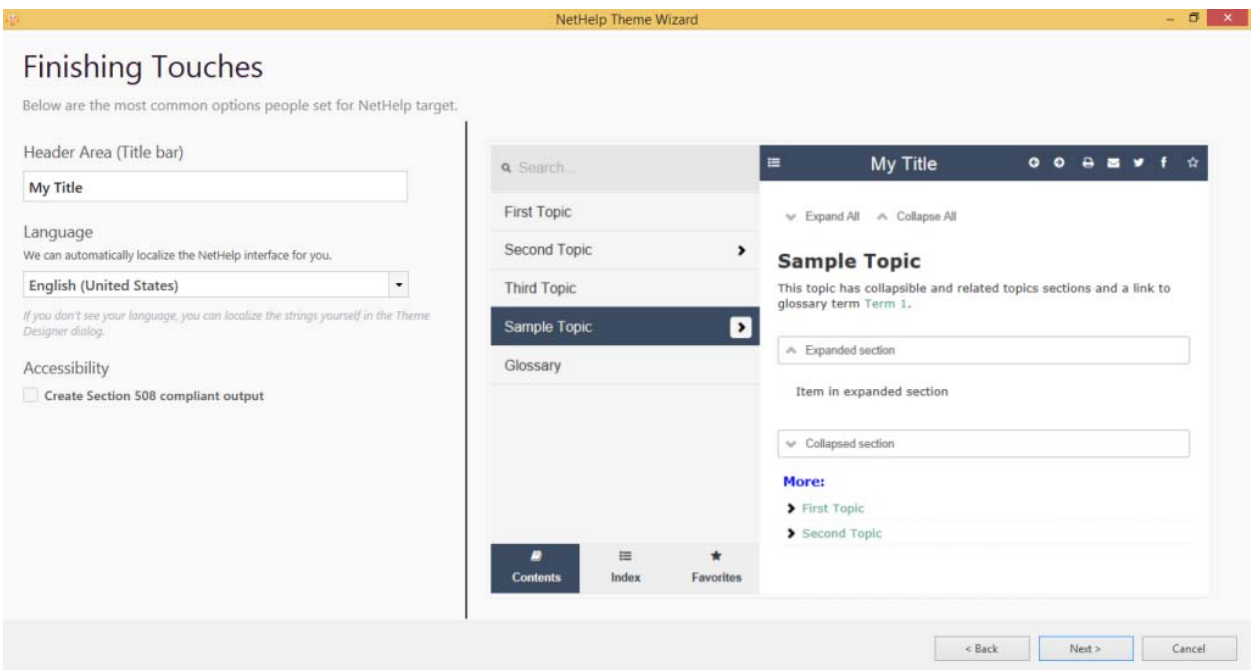


FavIcon	Click **Choose your icon** and select an icon (.ico). The icon will appear in

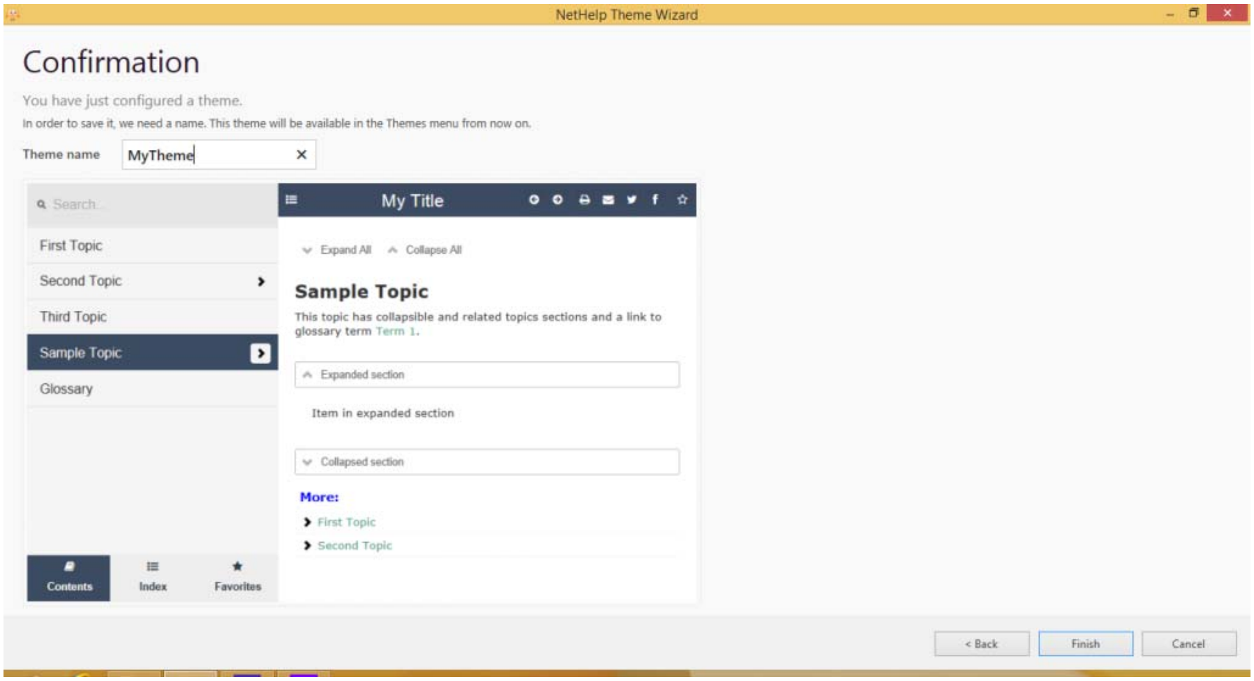
	browser tabs, address bars, and in **Favorites** lists.
Logo	Click **Choose your logo** and select an image. It will appear in the upper-left corner of your theme header.
Header area (Title bar)	To make a title appear in the title bar of your output, select **Show the Title of your NetHelp output in the top title bar** and enter a height for the header area.
Text area	Select **Fit text to window width** or enter a **Maximum text width** in the text box.

Note: If you chose a pre-defined theme in the **Choose Theme** window, you cannot change the branding properties in this window.

10. Click **Next**.
11. In the **Finishing Touches** window, enter a title to appear in the output **Title bar**, select a localization language for the NetHelp interface, and select the **Create Section 508 compliant output** check box if you need compliancy.



12. Click **Next**.
13. In the **Confirmation** window, enter a theme name and review the sample theme to make sure it is correct. You will not be able to enter a theme name if you chose an existing theme in the **Choose Theme** window.



14. Click **Finish**. If you used an existing theme, a dialog box appears, explaining the **Properties of the "NetHelp" target have been updated**. If you created a new theme, a dialog box appears, explaining the theme has been created and assigned to your NetHelp target.
15. Click **OK**. The theme appears in the **Theme button** drop-down list on the **Home** tab. You can select it and build the target to apply the theme.

Customizing Topics with the Transformation Wizard

Using the **Transformation Wizard**, you can easily insert boilerplate content or code into every Help topic in your Targets. Examples include: headers, footers, or code needed for technology such as analytics.

You can add transformations to **NetHelp 2.0**, **NetHelp Classic**, **HTML Help**, **EPUB**, **JavaHelp**, **Eclipse Help**, **Microsoft Help Viewer**, and **Microsoft Help 2.0** Help Targets. Your source documents can be Word, HTML5, or HTML.

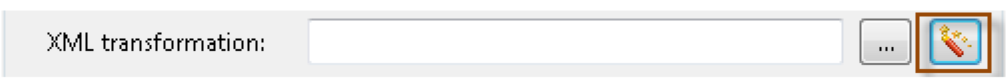
Transformation files created with the Wizard can be used in any project.

Custom HTML5 code can be defined for the following places in a topic:

- Inside the <head> element (for example: scripts, CSS, metadata, etc.)
- At the beginning of topic text (above the topic title)
- At the end of topic text

To open the Transformation Wizard

1. First, open the **Help Targets** dialog box from the **Home** tab. Click the **Target** ribbon group dialog box launcher. The **Help Targets** dialog box will open.
2. In the Advanced section, click the **Wizard** button next to the **XML Transformation** field. The **Transformation Wizard** will open.



To insert a transformation

1. Enter your custom code in the appropriate section(s): <head> element, beginning of topic, or end of topic.
2. Click **Save**. You will be prompted to name and save the file. The transformation file created (*.config) will be stored at the root of your Doc-To-Help project folder. (This location is the default, you can store it anywhere.)
3. Build your Target. The code will be inserted in each topic of the Target.

If would like to reuse the file in another project, simply open the **Help Target** dialog box in that project and click the **ellipsis** button next to the **XML Transformation** field. You can then navigate to the *.config file and select it.

There is an example transformation named timestamp.config that will add timestamp text to each topic of a Target. It is located at:

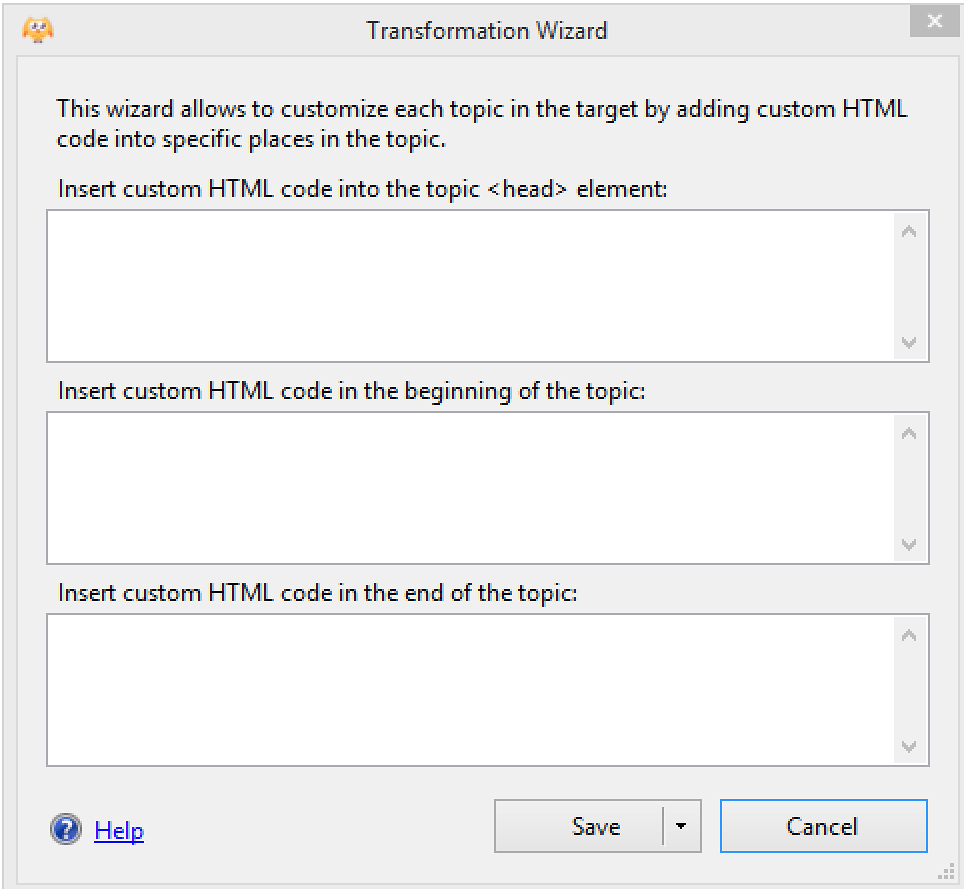
- 32-bit machine: C:\Program Files\MadCap Software\DocToHelp\Tranforms\Examples\Timestamp
- 64-bit machine: C:\Program Files (x86)\MadCap Software\DocToHelp\Tranforms\Examples\Timestamp

Note: Verify that the **Generate XHTML** check box is selected in the **Help Targets** dialog box before building your Target. It is selected by default in most Targets. Transformations cannot be used with WinHelp or Manual Targets.

Code examples

To insert keywords in the <head> element of each topic, add this to the first field of the Transformation Wizard:

```
<meta name="keywords" content="help authoring tool, doc-to-help, doctohelp,
online help"/>
```



To insert a distinctive header (green, bold, and italic) in each topic, add this to the middle field of the Transformation Wizard:

```
<div style="font-size:large; font-style:italic; color:green; font-weight:bold" >
Draft Version 1.1, for Internal Review Only
</div>
```

To insert a footer with a logo and hyperlink in each topic, add this to the last field of the Transformation Wizard:

```
<div style="width:90%">
      <hr style="color:#CCCCCC" />
      <div style="float: left">
            <img
src="http://www.componentone.com/newimages/company/nav_c1icon.png"
alt="ComponentOne Logo" />
      </div>
      <div style="float: right">
```

```
        ComponentOne, a division of GrapeCity, inc. <br />
        Pittsburgh, PA <br />
    <a href="http://www.componentone.com" target="_blank">ComponentOne.com</a>
    </div>
</div>
```

You can add programmatic and XSLT transformations to your projects also, see Advanced XML Transformations on page 259.

Advanced XML Transformations

Programmatic and XSLT transformations can be added to outputs to manipulate content. Programmatic transformations can be written in any .NET language, such as C# or VB.NET. This advanced feature gives you great flexibility, because it allows for post-build modification in virtually any imaginable custom way.

You can use the Transformation Wizard to insert boilerplate content or code into every Help topic in your Targets, see Customizing Topics with the Transformation Wizard on page 257 for details.

Examples of advanced transformations include:

- Replacing some text (tag) with variable text depending on a parameter specified in that tag.
- The Expanding/Collapsing Sections on page 316 feature in Doc-To-Help.

To use a custom transformation

1. Open the **Help Targets** dialog box from the **Home** tab. Click the **Target** ribbon group dialog box launcher. The **Help Targets** dialog box will open.
2. In the Advanced section, click the ellipsis button in the **XML Transformation** field and navigate to the file. Click **Open**.
3. Click **OK** in the **Help Targets** dialog box and build your Target.

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

- The expanding sections functionality (see Creating Expanding/Collapsing Sections on page 316) is implemented as an XML transformation (a programmatic one). Its full source code can be found in \\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp\Transforms\BuiltIn\Source\XMLInternal. (Don't modify these files; they are used in Target compilation.)
- There is an example transformation named **timestamp.config** that will add timestamp text to each topic of a Target. It is located at C:\Program Files [or Program Files (x86)]\ MadCap Software\DocToHelp\Tranforms\Examples\Timestamp

Notes on creating XML transformations:

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

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

- **<description>** — Arbitrary string describing the transformation, for explanatory purpose only.
- **<assembly>** — File name of the transformation assembly (including the .dll extension). For an XSLT transformation that assembly should always be C1D2HXMLInternal.dll. For programmatic transformations, it is

a path to the file, not the file name. The path is relative to the configuration file (where transform assembly is specified).

- **<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 transformation that element should always be as follows: <type>C1.D2H.XMLTransform.XSLTTransform</type>
- **<params>** — This element contains whatever elements the transformation may need as its parameters. All specific parameters will be passed to the transformation assembly when it is executed. For programmatic transformations, their interpretation is entirely dependent on the code implementing the transformation methods. XSLT transformations should be contained in one XSLT file that receives parameter values. The path to an XSL file is relative to the configuration file, so only the file name needs to be specified, for example: <xsl-file>C1D2HXMLInternal.xsl</xsl-file>.

Notes on creating programmatic transformations:

A programmatic transformation 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 transformation; creating XSLT transformations requires programming only in XSLT; no .NET programming is required):

- **void IXMLTransform.ReadParams(XmlNode node)**
  This method is called before executing the transformation, to initialize the transformation 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 transformation 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 transformation, applies it to a topic. The 'doc' parameter is the topic XML before applying the transformation (all transformations preceding this transformation 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 transformation'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).

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
HTML5	.xml	Doc-To-Help Content editor	**Editor**, **Insert**, **Table** (in Doc-To-Help)
Microsoft® Word	.doc .docx (Word 2007/2010)	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	**Doc-To-Help D2HML Styles** (added when Doc-To-Help is installed; Microsoft FrontPage and Adobe Dreamweaver only)

You edit .xml documents using the Content Editor window 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 285 for more information

HTML documents are edited using the **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 293 for more information.

Three ribbons (**Editor**, **Insert**, and **Table**) in Doc-To-Help are used to edit your HTML5 documents in the **Content Editor** window. See Editing HTML5 Documents for more information. Doc-To-Help includes a converter you can use to convert existing documents to HTML5. The converted documents are edited in the Content Editor window. For more information on converting documents to HTML5, see Converting Existing Documents to HTML5.

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 303 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, HTML5, or HTML.

After adding a new document to your project, you should build the Target (see Building a Target on page 341). 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 122.
2. If you are adding a new HTML or HTML5 (.xml) 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 **HTML5 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 **HTML5** 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 HTML5 and HTML documents. The **Default CSS** is specified in the Project Settings dialog box on page 182. 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 HTML5

1. Open the Documents pane on page 122.
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 — **HTML5 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 HTML5 while they are being imported. (HTML5 documents can be edited in Doc-To-Help's Content editor.) See Converting Existing Documents to HTML5 for details.

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 HTML5, 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 182 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 299.

To convert an existing Word or HTML document in your project to HTML5, see Converting Existing Documents to HTML5 on page 265.

Converting Existing Documents to HTML5

You can convert a Microsoft® Word or HTML document that is already part of your project to HTML5, or you can import a Microsoft Word or HTML document into your project as HTML5. The file extension of the converted document will be .xml.

To convert existing project documents to HTML5

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

1. Open the **Documents** pane.
2. Click the **Convert Multiple Documents to HTML5** button . The **Convert Documents to HTML5** 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 (by default documents are saved to the **Documents** folder), choose **Save converted files to** and click the **Browse** button). If the documents are Word files, confirm the **Word Style Map** (the Word to HTML5 styles map file). Click the **Browse** button to change it.

4. 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.

5. Click the **Convert** button.
6. 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 HTML5**. The **Convert Documents to HTML5** dialog box will open, with the single document selected in the list.

Importing an existing Word or HTML document into your project as HTML5

1. Open the Documents pane on page 122.
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 HTML5**.
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 HTML5, 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. The file will be converted and appear in the **Documents** pane.

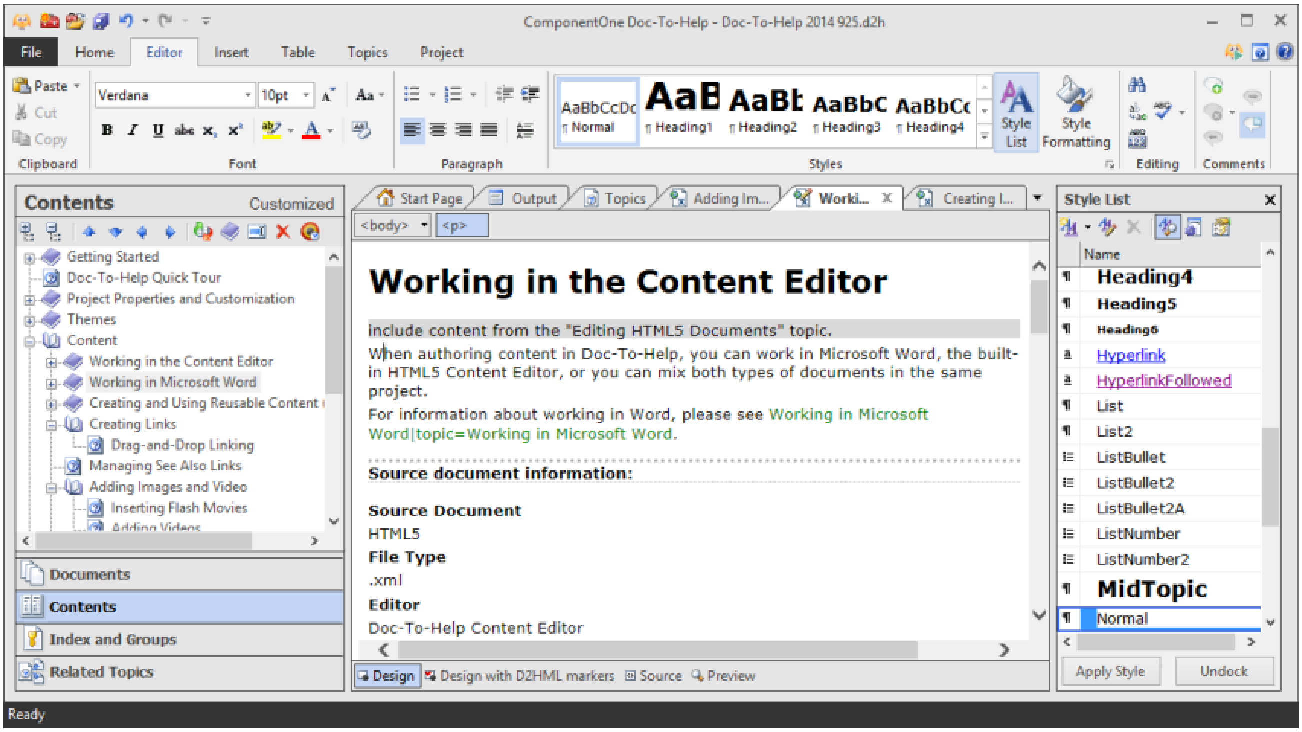
See the Editor on page 110, Table on page 113, and Insert on page 115 tabs, as well as Editing HTML5 Documents on page 266 for information on editing HTML5.

See Working with Source Documents on page 263 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 264.

Editing HTML5 Documents

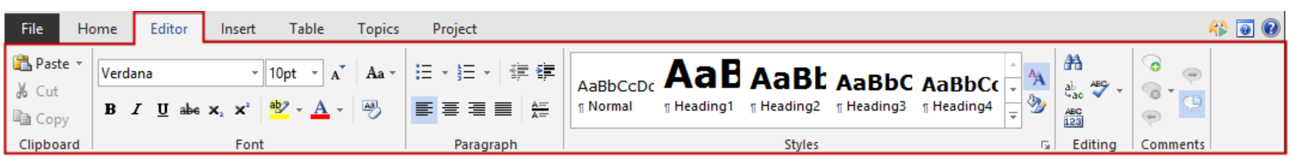
HTML5 documents are edited in the Content Editor window on page 128.

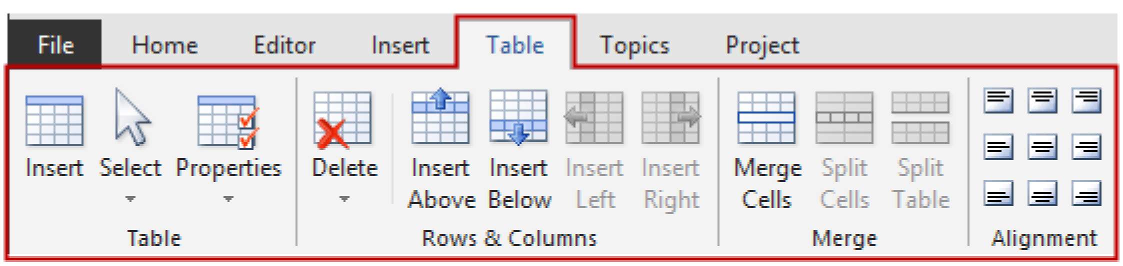


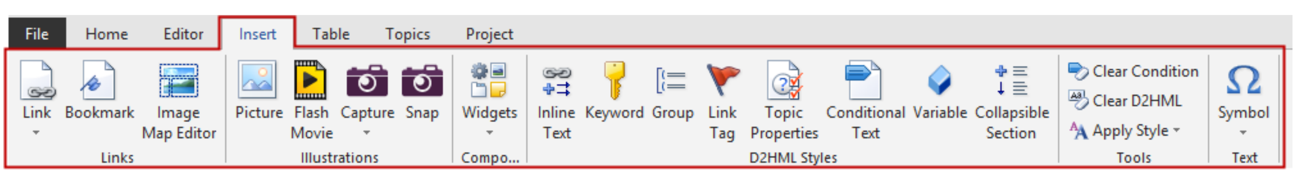
In this standards-compliant editor, you can work in WYSIWYG (Design View), or in Source View, which allows you to edit the code as you wish. Source View features IntelliSense, line numbering, and more (see Working in the HTML5 Source Code View on page 284).

The **Content Editor** opens when you open an .xml document from the Documents pane on page 122.

The Editor on page 110, Table on page 113, and Insert on page 115 ribbons are used to apply styles, insert tables/images/movies, add comments, create links, and more in .xml documents. These tabs appear only when the **Content Editor** is open.







Adding Widgets in the Content Editor

Adding widgets to your project can do a variety of things: add interactivity, improve the display of images and other content, save space, provide alternate navigation, and save you time.

All the Widgets have full interactivity in NetHelp and Eclipse Help, but interactivity varies in other Targets.

There are seven widgets available: LightBox on page 268, Gallery on page 270, Carousel on page 270, TopicContents on page 273, CodeHighlighter on page 273, Tabs on page 272, and Note on page 271.

To add a Widget

1. Click the **Insert** tab.
2. Put your cursor where you would like to insert a widget.
3. Click the **Widgets** button in the **Components** ribbon group.
4. Choose a Widget. The Widget Properties dialog box will open. If you choose Carousel, Gallery, or LightBox, the **Select Images** dialog box will open first. You can choose the images you'd like to use at this point, or click Cancel and configure the other widget properties first.

For each Widget, you can specify the dimensions, the float, and an optional CSS class. Additional properties are discussed in the details for each Widget.

A few notes about Widgets:

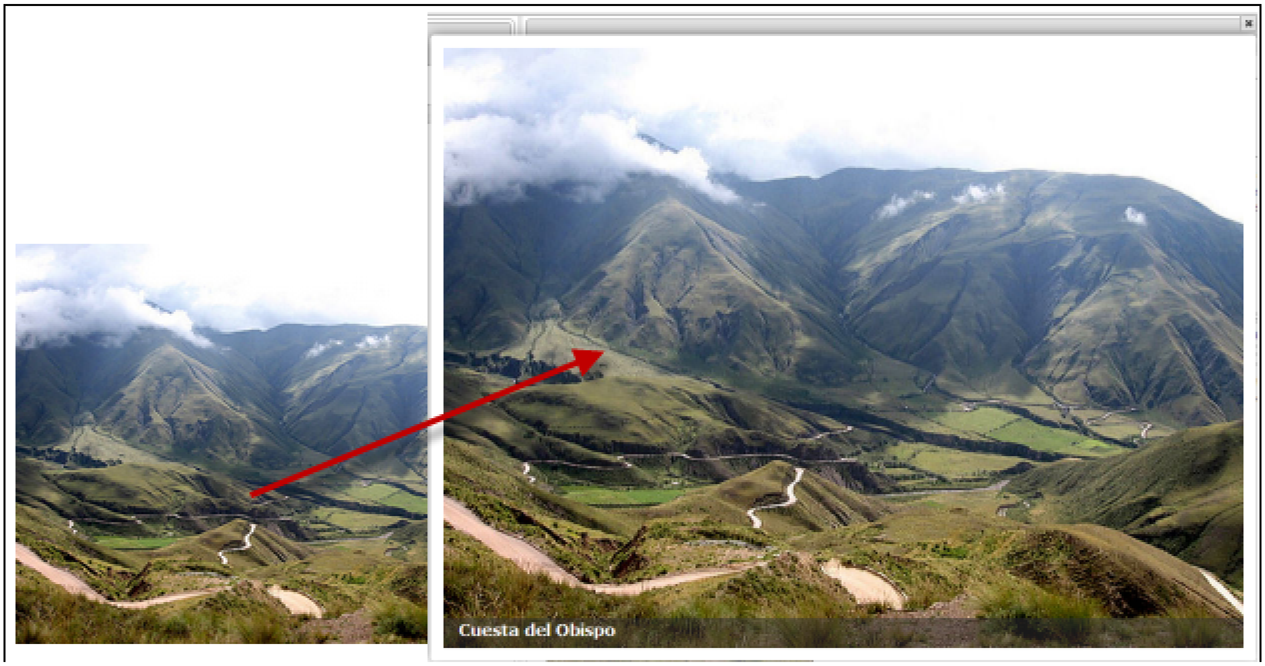
- If you leave the dimensions unspecified, they will be determined automatically by the widget's contents.
- By default, **Float** is set to **None**, which means that text will not flow around the Widget. You can change the float so that text flows around the Widget on the right or left.
- You can also specify a CSS class (style) if you want to further customize widget's appearance, for example, change its margins, border, etc. The class name can be any of the style names present in the project's CSS, or it can be defined in a CSS in the widget itself or in the widget's customization. The widget's style (cssClass) can also be set from the **Style List** as it can be for any other element in Doc-To-Help Content Editor. Note that if a CSS Class is specified, Float is ignored.

See the following blog posts for more information on customizing widgets: Meet the Widgets: The Note Widget, Meet the Widgets: The Image Widgets.

You can edit or delete a Widget by right-clicking on it and choosing the appropriate option. Or you can select the Widget and click the **Widgets** button, where **Edit** and **Delete** options are also available.

LightBox

With the LightBox widget, you can display an image (or other external content) in a pop-up interface. Your image will display as a thumbnail that the user can click on to display the larger image. A caption and comment text can be added.



This widget has full interactive functionality only in NetHelp and Eclipse targets. In JavaHelp, HTMLHelp, MS Help Viewer, and MS Help 2.0 targets the widget is shown in a simplified form.

In Manual, EPUB, and WinHelp Targets, it will not appear.

LightBox widget properties:

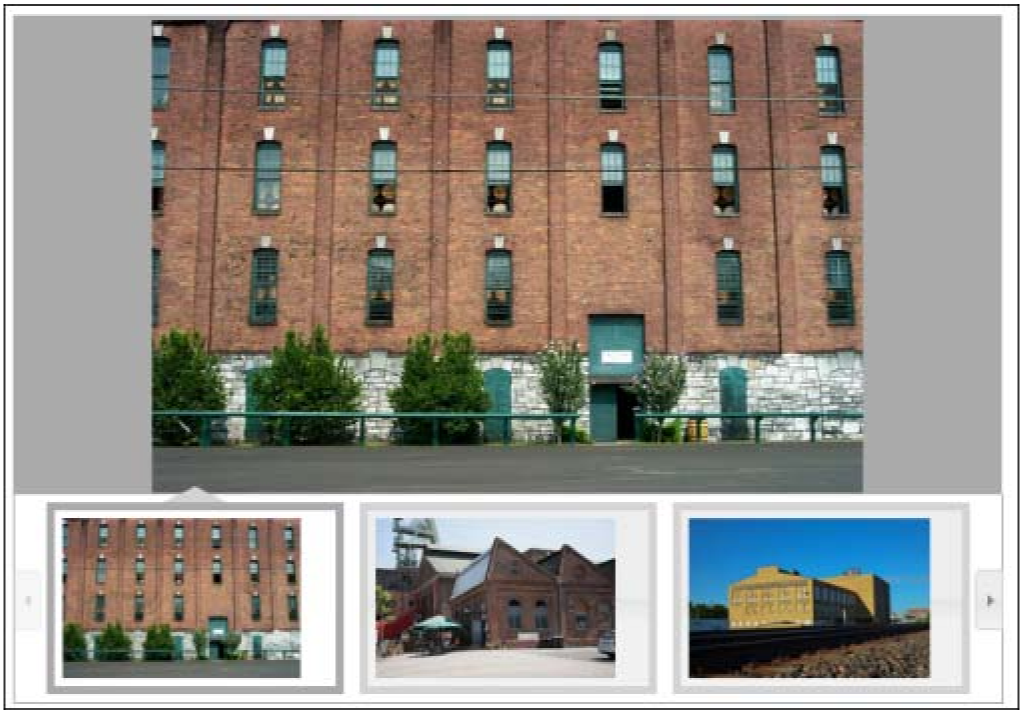
- **autoPlay —** By default, this is not turned on (false). The user must click on the thumbnail to display the larger image.
- **delay —** If autoPlay is turned on, the time span in milliseconds between image displays.
- **fullWidth —** The maximum panel width.
- **fullHeight —** The maximum panel height.
- **thumbHeight —** The height of the thumbnail. If not set, the widget height is used.
- **images —** The image(s) (and/or external content) used in the widget. Click the ellipsis to open the **Add/Remove Items** dialog box, where you can set additional image properties.

Image properties:

- **caption —** The caption text for the image, which appears as a tooltip on the thumbnail version and is shown in bold text on the widget.
- **comment —** The comment text displayed below the image.
- **thumbnail —** The path to the image file that contains the thumbnail version of the image. If a thumbnail image is not supplied, the original image will be resized and used.
- **image —** The path to the full size image file.
- **url —** The path to external content.
- **type —** The type of content (image or external content).
- **player —** The method used for displaying external content. Options include: img, iframe, swf, flv, wmp, qt, wijvideo.

Gallery

The Gallery widget displays a set of images with their thumbnails below. The user can select the thumbnail images (or use the forward or back buttons) and the larger images will be displayed with transition effects.



The Gallery widget has full interactive functionality only in NetHelp and Eclipse targets. In JavaHelp, HTMLHelp, MS Help Viewer, and MS Help 2.0 targets, the widget is shown in a simplified form.

In Manual, EPUB, and WinHelp Targets, it will not appear.

Gallery widget properties:

- **autoPlay —** Determines whether the images will be automatically displayed in order.
- **delay —** If autoPlay is turned on, the time span in milliseconds between image displays.
- **thumbHeight —** The height of the thumbnail. If not set, the widget height is used.
- **thumbsDisplay —** The number of thumbnails displayed.
- **images —** The image(s) (and/or external content) used in the widget. Click the ellipsis to open the **Add/Remove Items** dialog box, where you can set additional image properties.

Image properties:

- **comment —** The comment text displayed below the image. Each image can have its own comment.
- **thumbnail —** The path to the image file that contains the thumbnail version of the image. If a thumbnail image is not supplied, the original image will be resized and used.
- **image —** The path to the full size image file.

Carousel

The Carousel widget displays a set of images dynamically. The images will rotate automatically, or the user can select the forward or back buttons (or the Pager buttons) and the next image will be displayed with transition effects.

The Carousel widget has full interactive functionality only in NetHelp and Eclipse targets. In JavaHelp, HTMLHelp, MS Help Viewer, and MS Help 2.0 targets, the widget is shown in a simplified form.

In Manual, EPUB, and WinHelp Targets, it will not appear.

Carousel widget properties:

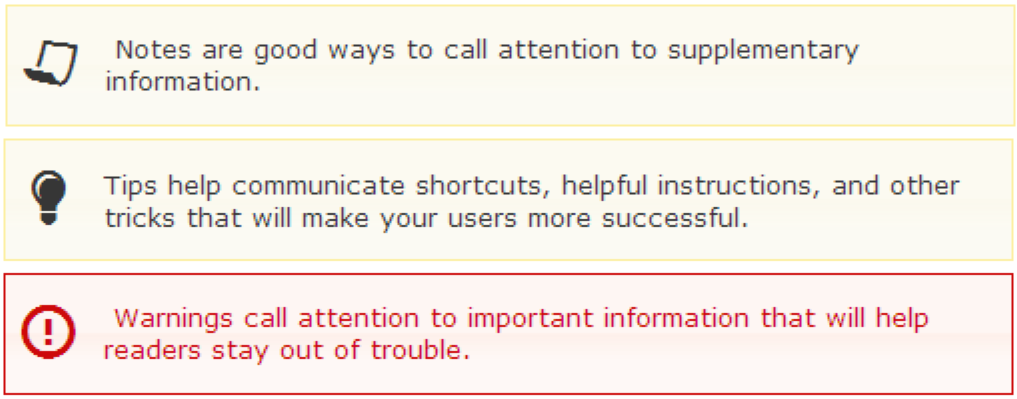
- **autoPlay —** Determines whether the images are automatically displayed in order.
- **delay —** If autoPlay is turned on, the time span in milliseconds between image displays.
- **pagerPosition —** Use this setting to determine the location of the Pager buttons that display below the widget (left, center, or right). These buttons are an alternate way the user can navigate through the images.
- **images —** The image(s) (and/or external content) used in the widget. Click the ellipsis to open the **Add/Remove Items** dialog box, where you can set additional image properties.

Image properties:

- **comment —** The comment text displayed below the image. Each image can have its own comment.
- **image —** The path to the image file.

Note

Using the Note widget is a great way to draw attention to a Tip, Note, or Warning message. The Widget will display the content in a frame, with an icon on the left. A custom image can be substituted for the default icon.



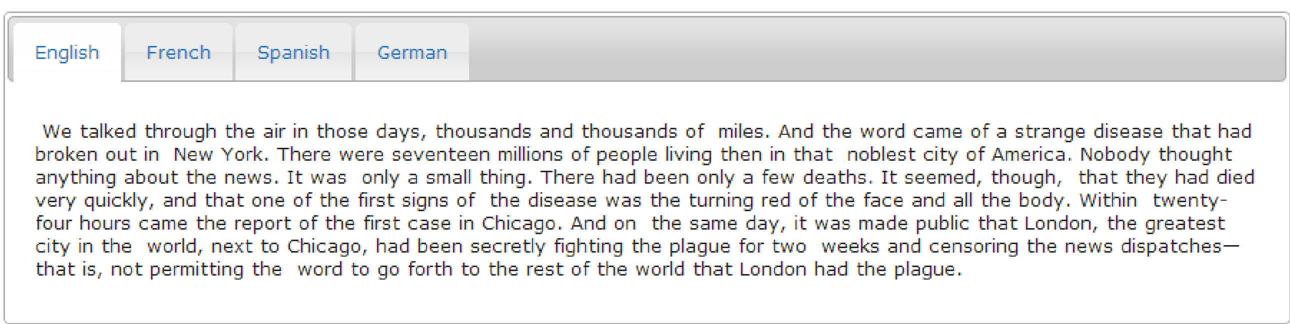
The Note widget works in all targets (except WinHelp). After the Note widget is inserted, you can enter your text by clicking in the widget.

Note widget properties:

- **type** — The type of Note you want to create. The options are: note, tip, or warning. Each has a different icon, and the warning note is red.
- **customImage —** The image file to use for the icon in the Note. If not specified, the default icons will be used. If customImage is specified, it will override the **type** property.
- **customStyle —** Optional. The CSS class name used to specify the widget colors and other style properties, instead of the standard ones defined by the **type** property. The icon is not affected by this property.

Tabs

The Tabs widget is used to break content into multiple “tabbed” sections that can be swapped to save screen real estate. It is ideal for code samples – you can display the same sample in different languages in a compact space.



The Tabs widget has full (interactive) functionality in NetHelp, HTMLHelp, and Eclipse Help targets. In other targets (including Manual) it shows the tab contents for every tab in a sequence, with the tab title used as the section heading for each. The Tabs widget is not supported in WinHelp Targets.

In the Content editor, the Tabs widget is fully interactive in Design as well as in Preview mode.

When you create a new Tabs widget, it will have two tabs by default. You can add more using the **tabs** property. After the Tabs widget is inserted, you can enter the content in each section; simply click on the tabs to navigate each one.

Tabs widget properties:

- **alignment —** Use this field to specify where the tabs should be located (top, bottom, left, or right). The default is top.

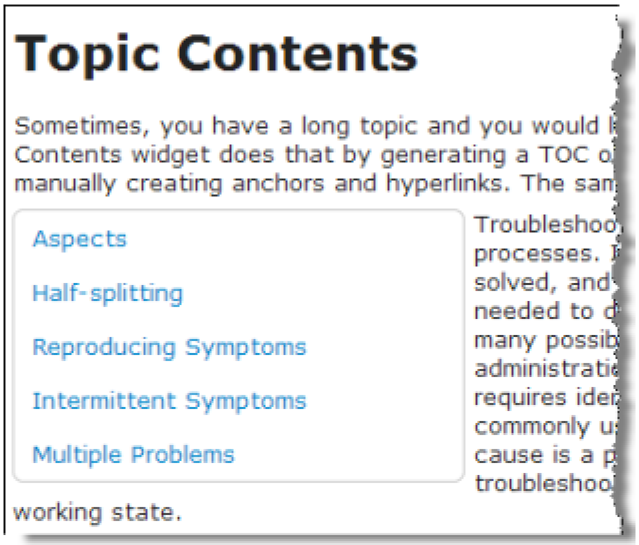
- **tabs** — Click the ellipsis to open the **Add/Remove Items** dialog box, where you can set additional tab properties, as well as rearrange tabs.

Tab properties:

- **id** — the id number of the tab.
- **title** — Use this field to add a name to each tab.

TopicContents

The TopicContents widget displays the headers inside the current document. (It resembles the navigation menu in many wikis.) The headers displayed in the widget are the H1 – H6 and MidTopic styles (you can change this by using widget customization).



This widget does not have any properties.

This widget works only in NetHelp. HTMLHelp, and Eclipse Help targets; it will not appear in the other targets. In the editor, it works in Preview mode, but in Design mode it always displays a rectangle with the caption "Topic Contents" inside).

CodeHighlighter

The CodeHighlighter widget is a frame for adding syntax highlighting to programming code (usually a code sample).


```
$("#pager").wijpager({
   pageCount: $("#pages").wijwizard('count'),
   pageIndex: $("#pages").wijwizard('option', 'activeIndex'),
   mode: "numeric",
   pageIndexChanged: function () {
       var pageIndex = $("#pager").wijpager("option", "pageIndex");
       $("#pages").wijwizard({ activeIndex: pageIndex });
   }
   });
```

It is based on the popular open source code prettifier from Google (google-code-prettify): https://code.google.com/p/google-code-prettify/

It also includes a **Copy to Clipboard** button that works in IE and in any browser supporting Flash (it uses the ZeroClipboard library that utilizes Flash; which is necessary in all browsers except IE because other browsers don't have native clipboard access due to security concerns). Note that the **Copy to Clipboard** button does not appear in local NetHelp (NetHelp deployed in file system).

CodeHighlighter works in all targets (except WinHelp), but **Copy to Clipboard** works only in NetHelp, HTMLHelp, and Eclipse Help.

In the Content editor, CodeHighlighter highlightes the code in Preview mode but not in Design mode.

CodeHighlighter has the following properties:

- **lang —** This is an advanced property. Use it only if you are very familiar with google-code-prettify documentation. For most common programming languages, the language used in the code is detected automatically. If you are using one of the languages that are not supported by default, you can specify the language here. See google-code-prettify documentation for details.
- **linenums —** Turns on line numbering. To start numbering from 1, set this property to 1. To start from an alternate number, change the number.

Customizing Widgets

If you want to customize the internal style and behavior of a widget in your project, you should use WidgetThemes.

To Create a Widget theme

1. Create a folder called **WidgetThemes** in the same folder where your Widgets folder is located in the user documents folder:

   Users\<user-name>\Documents\My Doc-To-Help Projects\Doc-To-Help\Themes\WidgetThemes

2. Inside that folder, create widget theme folders, for example:

   ...\WidgetThemes\Theme1

   ...\WidgetThemes\Theme2

3. Create a text XML file called map.xml containing <map> elements each defining a widget theme to use for a certain Doc-To-Help theme belonging to a certain platform. For example:

   <maps>

     <map platform="nethelp" theme="Tabs" widget-theme="Theme1" />

     <map platform="nethelp" theme="Responsive" widget-theme="Theme2" />

     <map platform="nethelp" theme="MyTheme" widget-theme="MyWidgetTheme1" />

   </maps>

The map.xml file effectively specifies which widget theme to use with each theme. So, a widget theme can be considered a part of a Doc-To-Help theme, an addition to it.

For every Doc-To-Help theme, you can (optionally) create a separate widget theme, or you can use a single widget theme for multiple Doc-To-Help themes, depending on what you specify in the map.xml file.

A widget theme is a folder containing folders with the widget names (same names as in the ...\Themes\Widgets folder), for example:

```
...\WidgetThemes\Theme1\CodeHighlighter
...\WidgetThemes\Theme1\TopicContents
```

You don't have to create folders for all widgets there, only for those that you want to customize.

If a widget is present in a widget theme, create a text XML file called widget.xml there. It can contain two kinds of elements, <css> and <script>, for example:

```
<widget>
  <script>..\WidgetThemes\Theme1\TopicContents\myscript.js</script>
  <css>..\WidgetThemes\Theme1\TopicContents\mystyles.css</css>
</widget>
```

Script files referred to by the widget.xml file must be inside the widget theme folder (Theme1 in the example above). CSS files referred to by the widget.xml file can be either inside the widget theme folder or in the original widget definition folder (if a file is present in both places, the former overrides the latter).

Script files are **added** to the script defined by the widget. They are executed before the scripts defined by the widget are executed, thus giving you an opportunity to set options that modify widget behavior. This is done (in JavaScript) using the following pattern:

```
window.c1WidgetOptions = window.c1WidgetOptions || {};
window.c1WidgetOptions.Tabs = {
  collapsible: true
};
```

where **Tabs** is an example of a widget name, and **collapsible** is the name of one of the Tab widget's options.

CSS files **replace** the CSS files specified in the widget's definition.

Important: if a widget is customized, that is, it is included in a widget theme that is used in a Doc-To-Help theme, then all CSS file names are removed from the original widget definition and replaced with CSS file names specified in widget.xml in the widget theme. So, unless you want to remove some CSS files from the widget definition completely, you need to repeat all CSS files from the widget definition in your widget theme's widget.xml. You can add a CSS file there, and you can replace any file with your own, but be aware that only the CSS files listed there will be used.

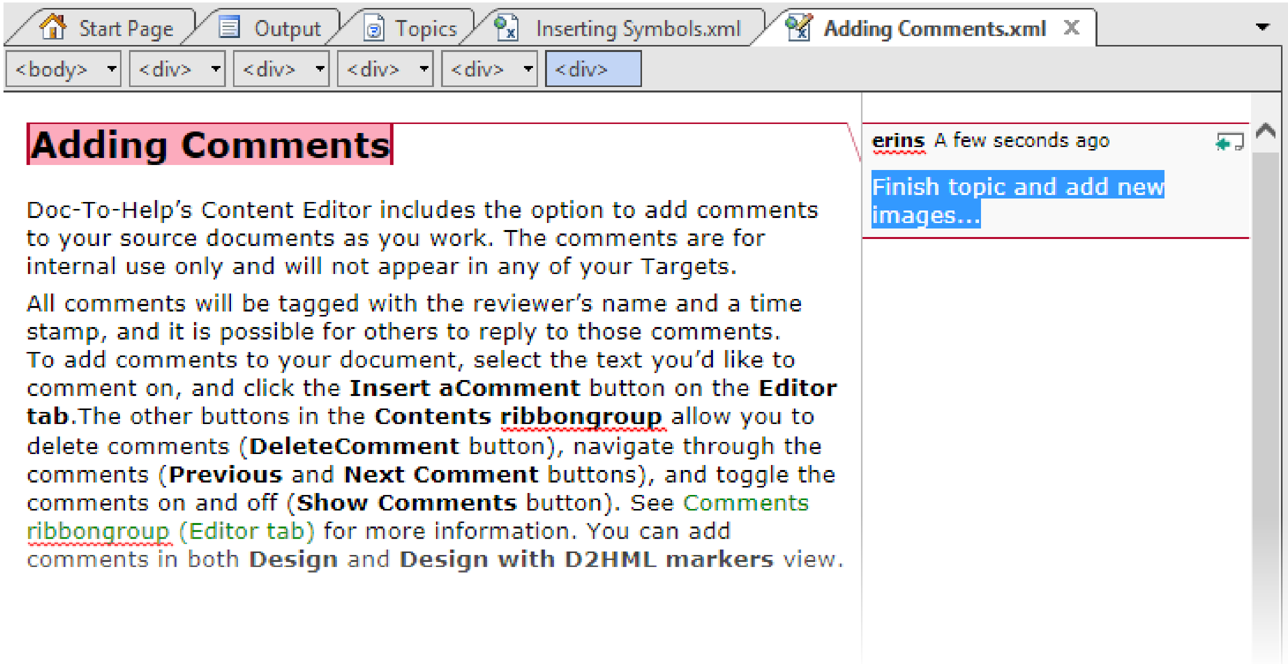
Note: Manually changing files in the widget definition is strongly discouraged, as is any manual change of an internal Doc-To-Help file. Manually changing that file will also change the widget's behavior in all projects on your computer.

Adding Comments in the Content Editor

Doc-To-Help's Content Editor includes the option to add comments to your source documents as you work. The comments are for internal use only and will not appear in any of your Targets.

All comments will be tagged with the reviewer's name and a time stamp, and it is possible for others to reply to those comments.

To add comments to your document, select the text you'd like to comment on, and click the **Insert a Comment** button on the **Editor tab**. The other buttons in the **Contents ribbon group** allow you to delete comments (**Delete Comment** button), navigate through the comments (**Previous** and **Next Comment** buttons), and toggle the comments on and off (**Show Comments** button). See Comments ribbon group (Editor tab) on page 113 for more information. You can add comments in both **Design** and **Design with D2HML markers** view.



To reply to any comment, click the "reply" icon on the upper right of the comment. The comments will be threaded.

If you'd like other members of your team to add comments to your documents, you could collaborate using Doc-To-Help's Team Authoring on page 381, or store your project on a network. Anyone with a Doc-To-Help License can collaborate on projects stored on a network.

Applying Styles in the Content Editor

You can apply styles and formatting in the Content Editor window on page 128 several ways.

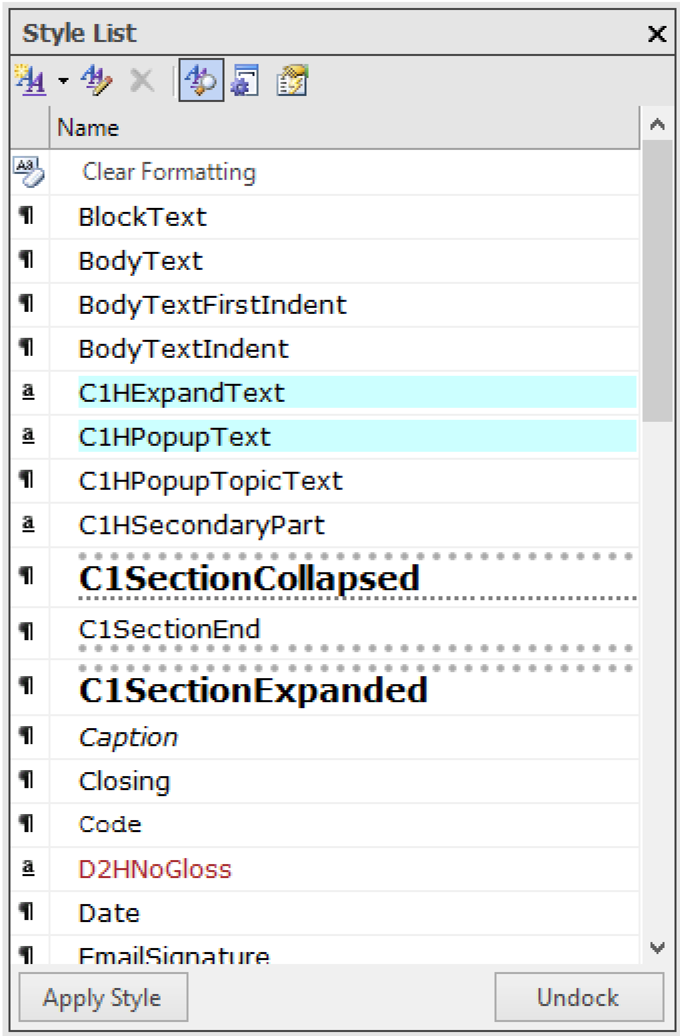
See Editing a CSS on page 176 for information about editing and changing your project's CSS files.

To apply a style using the Style List

1. Click the **Editor** tab.
2. Select the text in the **Content 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 a. 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 178 for more information about the **Style List** window.

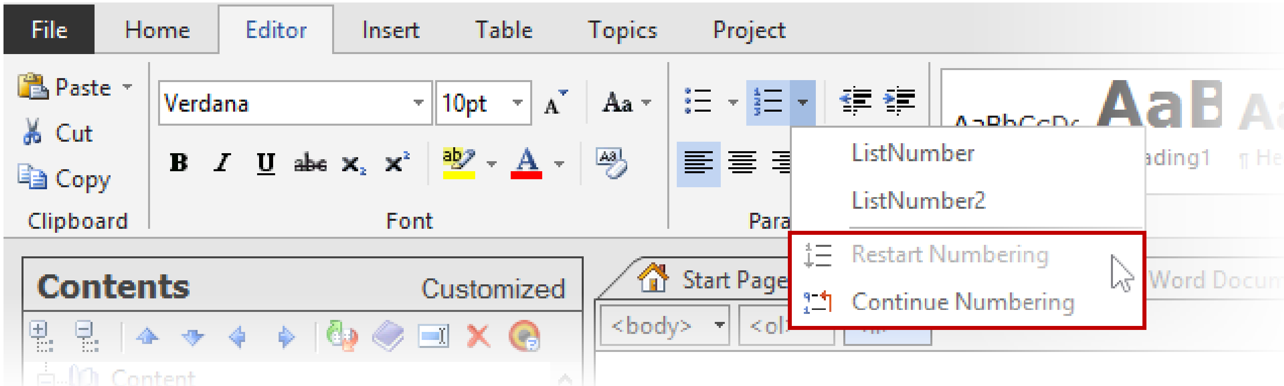


Bullets, Numbering, and Nested Lists

You can apply bullets and numbering two ways: either using the **Style List** (choose the appropriate styles: **ListBullet** or **ListNumber**), or the **Bullets** or **Numbers** toolbar buttons on the **Editor tab**.

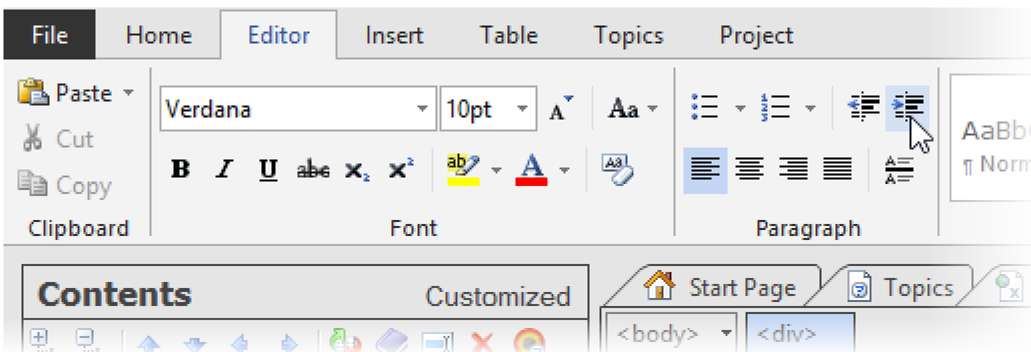
To restart or continue a numbered list, choose the appropriate option from the **Numbers** toolbar button drop-down. You can also right-click in the **Content Editor** window and choose from the menu.

Restart Numbering stops the current list and begins a new list. **Continue Numbering** creates a new item in the list and reorders the numbering starting at that point.



To create nested lists, use **Increase Indent** toolbar buttons. (You may also use the **ListBullet2** and **ListBullet2A** styles in the **Style List**, but do not use styles and toolbar buttons together.)

If the selected text is inside a list, clicking on the **Increase Indent** toolbar button will create a new nested list with the selected items. If the selected text is part of a nested list, then clicking on the **Decrease Indent** toolbar button will unnest those items.



To apply a Style using the Style Gallery

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

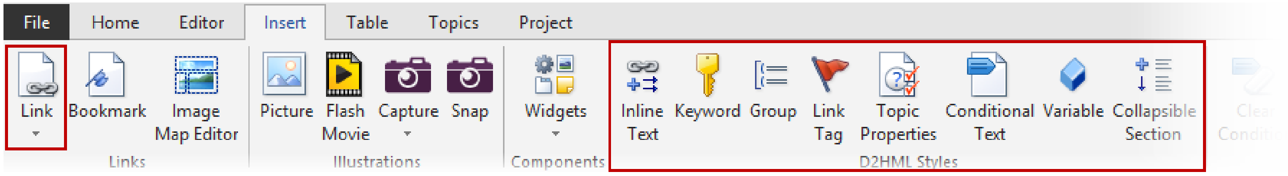
You can change the list of styles displayed in the **Style Gallery**. See Managing Style Gallery Styles on page 280 for details.

To apply a D2HML Style

Doc-To-Help Markup Language (D2HML) styles are used to add links, keywords, groups, variables, etc. in your HTML5 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 115. See Using D2HML on page 303 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 **Content 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 176 for more information on creating styles.

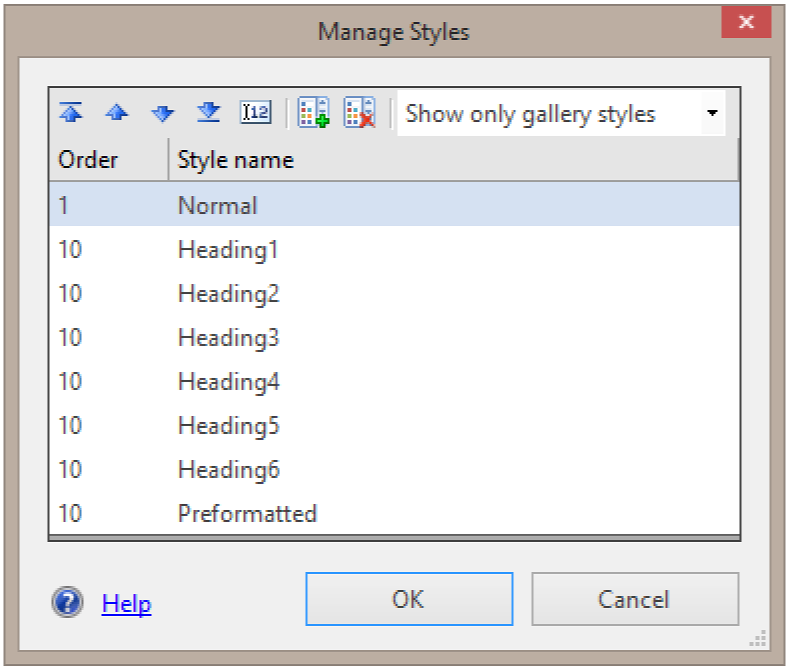
Applying Local Formatting

If you would like to apply formatting without using a style (local formatting), you may do so in the Content Editor window on page 128, using the Font on page 111 and Paragraph on page 112 ribbon groups of the **Editor** tab.

There are many factors to consider before using local formatting. Using local formatting 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.

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.

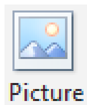
Select a **Style name,** then…

- 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 Content Editor

Using the **Picture Properties** dialog box, you may insert .gif, .jpg, .png, .jpeg, .bmp, .wmf, or .emf image files into the Content Editor window on page 128.

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 Creating Section 508 Compliant NetHelp on page 19.
6. If desired, specify the **Width** and **Height** of the image. The image will appear in the **Preview** area.

   If you select the **Keep Aspect Ratio** check box, you can change a single dimension of the picture (either height or width), and the other dimension will automatically be adjusted to maintain the aspect ratio.

   Click the **Reset** button to return the image to its original size.
7. Click **OK**.

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.

Creating Hyperlinks in the Content Editor

Using the **Hyperlink Properties** dialog box, you may create a hyperlink to a URL or another file from the Content Editor window on page 128. 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**.

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 305.

Creating Bookmarks in the Content Editor

Using the **Bookmark Properties** dialog box, you may insert or delete a bookmark in your document in the Content Editor window on page 128. 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 305.

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 editor.

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.

Inserting Symbols in the Content Editor

You can insert symbols — copyright symbols, trademark symbols, Unicode characters, etc. — as well as line breaks and non-breaking spaces using the **Symbol** button on the **Insert** tab.

Editing and Proofing in the Content Editor

The **Editor** tab includes several different functions to edit your document in the Content Editor window on page 128.

The Clipboard ribbon group on page 111 can be used to cut/copy/paste. The Editing ribbon group on page 112 includes **Find** and **Replace** functions. You can find/replace within a single document, or across the entire project. You can set the language of the spell checker by document using the drop-down of the **Spelling** button; to change it for all documents in the project click the **File** tab > **Tools** > **Options**, and click the **Spelling** button. Click the **Word Count** button to receive detailed word and character count information.

Inserting Tables in the Content Editor

A table may be configured and inserted in the Content Editor window on page 128 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.

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** text (for screen readers). For more on accessibility, see Creating Section 508 Compliant NetHelp. on page 19
6. If desired, specify the cell width, table alignment (left, center, right), and border width.
7. Click **OK**.

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

A table may be modified in the Content Editor window on page 128 using the **Table** tab.

To insert a table see Inserting Tables.

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

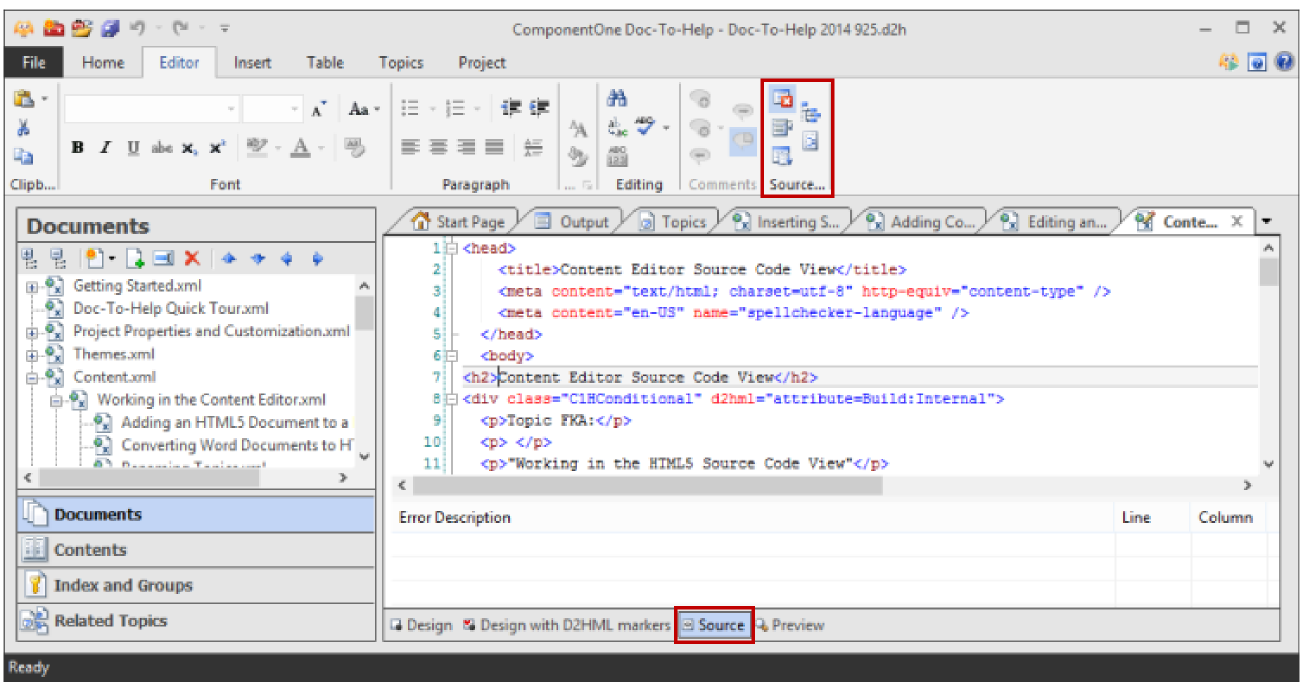
1. Open the **Table** tab.
2. Place your cursor in the table you wish to edit.
3. Click the **Select** button drop-down list.
4. Choose **Select Cell**, **Select Row**, or **Select Table**.
5. Choose the **Merge Cells** or **Split Cells** button from the Merge ribbon group on page 114. If the option 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 114.

Working in the HTML5 Source Code View

Clicking on the **Source** button at the bottom of the **Content Editor** will open the Source Code view.



In this view, you can edit the HTML5 code natively. The editor will automatically validate your content as you work. If it finds errors, they will be displayed so that you can fix them, or Doc-To-Help can fix them for you.

Working in this view enables the **Source Code ribbon group** on the **Editor tab**, which includes options that control the display, as well as provides tools to fix errors and formatting.

- **Error List** — This button toggles the error list on and off.
- **Fix Errors** — Click this button to fixes all of the validation errors in the document.
- **Word Wrap** — This button Toggles word wrap on and off.
- **IntelliSense** — This button will display the IntelliSense (autocomplete) menu.
- **Format Selection** — Well formatted HTML is nested appropriately so that is human-readable and easier to troubleshoot. Select your code and click this button to formats the selected section.
- **Format Document** — Click this button to format the entire document.

Several other useful options are available on the **Editor tab** in Source view, specifically the Bold, Italic, Subscript and Superscript buttons, as well as the Spelling button so that you can use the spell checker.

The editor also has two other features that are useful for troubleshooting and readability: line numbering and collapsible outlining. Error messages are referred to by line number, and collapsible outlining (the + and – signs next to the numbers) allows you to collapse elements and their children for easier viewing.

For more information about HTML5, see the following:

Getting Started with HTML5 in Tech Comm

HTML 5.1 Specification

W3 Schools HTML5 tutorial

Saving HTML5 Documents

There are several options for saving HTML5 documents. These options are available only when an HTML5 document is open in the **Content Editor** window.

To save the document you are currently editing: File tab > Save.

To save all open documents: File tab > Save All. (Or use the button on the Quick Access Toolbar on page 106.)

To save a document with another name: File tab > 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.

To close all open HTML5 documents at the same time, use the drop-down arrow on the far right of the document tabs. There is an option to **Close All Documents**.

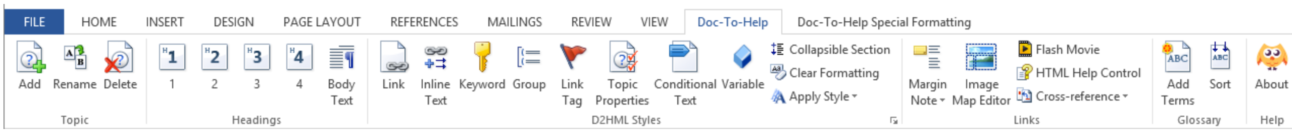
Open Microsoft® Word documents and HTML documents must be saved in those applications.

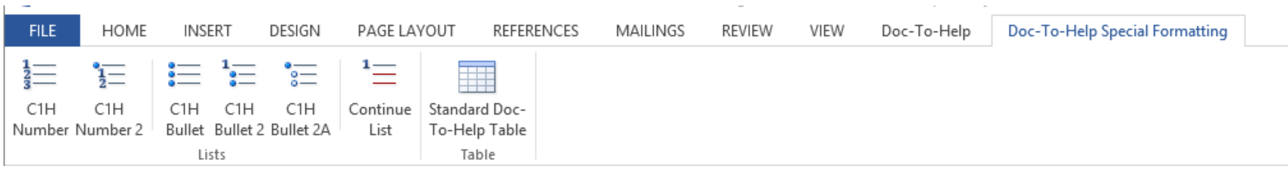
Editing Word Documents

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.

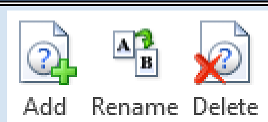
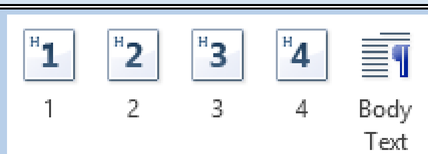
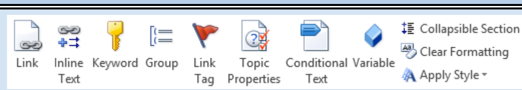
These toolbars or ribbons are: **Doc-To-Help** and **Doc-To-Help Special Formatting**.

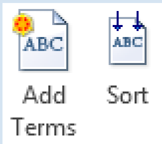
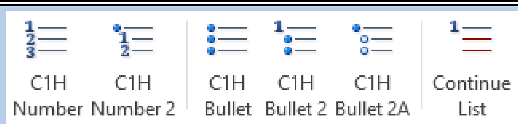
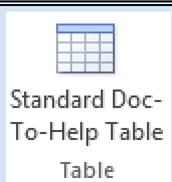
Doc-To-Help ribbons in Microsoft Word 2007/2010/2013:





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
 on page **Error! Bookmark not defined.**	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 288.
	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 303.
	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 289.
	Image Map Editor	In Help Targets, use the Image Map Editor to create hyperlinks within a graphic. See Creating Image Maps on page 296.
	Insert Flash Movie	You may insert Flash movies (.swf) into Help Targets — see Inserting Flash Movies on page 295.
	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 290.
	Insert Cross Reference/ Complete Cross	Use to insert cross-references that will appear as page numbers in printed targets and hyperlinks in Help Targets.

	Reference	See Inserting Cross References on page 289.
	Add/Sort Glossary Terms	Use to add terms to your glossary from any Source document. See Adding Terms to the Glossary on page 291.
	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	Use to insert a Table in your Doc-To-Help project. See Inserting a Standard Doc-To-Help Table on page 291.

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 7 for more information.

For more on applying styles, see Applying Styles in Microsoft Word on page 287.

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 303 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 2007/2010/2013, 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.

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 303 for the list of Word Styles that correspond to the D2HML toolbar buttons.

See Editing a Template on page 10 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 rename a single-topic HTML document in the Content Editor, see Renaming Topics on page 299.

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 to use the new title (see Topic Properties dialog box on page 321). 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.

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 2007/2010/2013, 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, 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 182.
11. Type a period after the page number, if necessary.

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 313 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**.

Shortcut Keys

If you prefer to use shortcut keys when authoring in Microsoft Word, here are available shortcuts for the **Doc-To-Help** and **Doc-To-Help Special Formatting** ribbons or toolbars.

Doc-To-Help ribbon or toolbar

Topic ribbon group
Add — Alt+Insert
Rename — Alt+F2
Delete — Alt+Del

Headings ribbon group
1 — Ctrl+Shift+1
2 — Ctrl+Shift+2
3 — Ctrl+Shift+3
4 — Ctrl+Shift+4
Body Text — Ctrl+Shift+B

D2HML Styles ribbon group
Link — Alt+Ctrl+J
Inline Text — Alt+Ctrl+E
Keyword — Alt+Ctrl+W
Group — Alt+Ctrl+R
Link Tag — Alt+Ctrl+L
Topic Properties — Alt+Ctrl+T
Conditional Text — Alt+Ctrl+C
Variable — Alt+Ctrl+B
Collapsible Section — Alt+Ctrl+P
Clear Formatting — Alt+Ctrl+F
Apply Style — no shortcut

Links ribbon group

Margin Note
Create — Alt+M
Remove — Alt+Shift+T

Set Link — Alt+Shift+M
Delete Link — Alt+Shift+L
Add Definition — Alt+D
Delete Definition — Alt+Shift+D
Clear Definition Links — Ctrl+Shift+D
Highlight Definition Links — Alt+H

Image Map Editor — Ctrl+Shift+R
Flash Movie — no shortcut
HTML Help Control — no shortcut
Insert Cross-reference — Alt+R
Complete Cross-reference — Alt+Shift+R

Glossary ribbon group
Add Terms — Alt+Shift+G
Sort — Ctrl+Shift+G

Doc-To-Help Special Formatting ribbon or toolbar

Lists ribbon group
C1H Number — Alt+0
C1H Number 2 — Alt+1
C1H Bullet — Alt+2
C1H Bullet 2 — Alt+3
C1H Bullet 2A — Alt+4
Continue Lists — Alt+5

Table ribbon group
Standard Doc-To-Help Table — no shortcut

The same keys will work for the **Doc-To-Help** and **Doc-To-Help Special Formatting** ribbons or toolbars in a Manual Target (.doc or .docx), with the addition of **Master Print** — Ctrl+P.

Editing HTML Documents

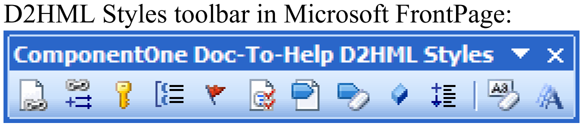
Note: Doc-To-Help includes a built-in HTML5 editor. For more information see Content Editor Window on page 128.

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 **Doc-To-Help D2HML Styles** toolbar or menu. This toolbar allows you to use D2HML while using FrontPage or Dreamweaver as your editor.

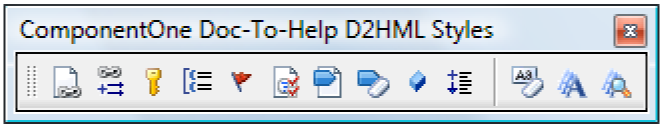
The **D2HML toolbar** will be installed on your machine automatically for Microsoft FrontPage; you will be prompted to install the toolbar in Adobe Dreamweaver during Doc-To-Help installation.

For information about using each toolbar button, see Using D2HML (Doc-To-Help Markup Language) on page 303.

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 294 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 **File** tab.
2. Click on the **Doc-To-Help Options** button. The **Options** dialog box will open.
3. Click the **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 108. See Guide to Templates and Styles on page 7 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 303 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 168.

To apply Character Styles, use the **Doc-To-Help D2HML Styles** toolbar or menu in FrontPage or Dreamweaver. See Using D2HML on page 303 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 176 for more information.

Inserting Flash Movies

It is easy to insert Flash movies into your projects.

To insert a Flash movie

1. Place your cursor at the point you'd like to insert a movie.
2. Click the **Flash Movie** button  (In Word, on the **Doc-To-Help** ribbon; in the **Content Editor** on the **Insert** ribbon). The **Movie in Flash Format** dialog box will open.
3. 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. (Storing it in your project will increase the size of your Help file.)
4. If desired, enter the **Alternative text** (for screen readers) for this movie. For more on accessibility, see Creating Section 508 Compliant NetHelp on page 19.
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**.

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 Image Maps (graphics with links)

For Help Targets, you can create hotspots from a graphic to topics, keywords, or groups using the **Image Map Editor**. These interactive graphics create "shortcuts" for the reader to relevant information. A sample can be found here: Navigating Doc-To-Help on page 105.

Since the hotspots are only available in online Help Targets, you may want to add the same links under the graphic for Manual Targets, then mark that text with the "Printed Manual" condition so it does not appear in online Help (see Marking Text as Conditional on page 313).

To create an image map

1. Select a graphic in Word or the **Content Editor**.
2. Click the **Image Map Editor** button 

(In Word, on the **Doc-To-Help** ribbon; in the Content Editor on the **Insert** ribbon). The **Image Map Editor** dialog box will open.
3. Click the **Draw hot region** button at the top left and select a region. After you have done so, a default name for the region ("Hot Region #X") will appear in the box on the right. It is in this area that you will create links to topics.
4. If you would like to split the graphic into equal regions, click the **Split into Hot Regions** button instead. See Splitting a Graphic into Equal Regions on page 296.
5. Click the **Link** button on the upper right. The **Link** dialog box will open.
6. Choose the **Topic**, **Keyword**, or **Group** to link the region to. Click **OK** to close the **Link** dialog box. (The specified link will display in the lower right of the **Image Map Editor** dialog box). If you want to delete the link later, click the **Remove Hot link** button in the **Image Map Editor** dialog box.

   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.

   You can bring a region to the front or back, change the default line color for a region, and zoom in or out using the toolbar on the top left.

   If you would like to adjust the coordinates of a region, select it and drag it on the graphic, or for more precise adjustments use the Layout section on the right.
7. Click **OK**.

Note: In Word, there is the option to make hot regions visible. This will make your hot regions visible in WinHelp Targets only.

Splitting a Graphic into Equal Regions

You can split a graphic (or a region you have already created) into equal regions using the **Split into Hot Regions** dialog box. This dialog box is opened by clicking the **Split into Hot Regions** button on the **Image Map Editor** dialog box. (See Creating Image Maps on page 296.)

To split an entire graphic, or another region into equal sections

1. Select a graphic in Word or the Content Editor.
2. Click the **Image Map Editor** button  (In Word, on the **Doc-To-Help** ribbon; in the Content Editor on the **Insert** ribbon). The **Image Map Editor** dialog box will open.
3. If you would like to split a previously created region, select that region in the graphic.
4. Click the **Split into Hot Regions** button. In the **Split Into Hot Regions** dialog box:
   - To split the entire graphic, select the **Split Entire Region** radio button and enter the **Row** and **Column** dimensions.
   - To split a region, select the **Split selected hot region** radio button and enter the **Row** and **Column** dimensions. If you would like the original region to be deleted, select the **Delete the region after splitting it** check box.
5. Click **OK**.

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 321.

To open the Document Properties dialog box

1. Select a document in the Documents pane on page 122.
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 109.

Document

Name: The read-only name of the document.

Type: Notes the document type — Word, HTML5, or HTML.

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. HTML5 and HTML documents can be either multi-topic or single-topic.

Title: If this is a single-topic HTML5 or HTML source document, the title will be displayed here. To rename it, click the **Rename** button next to this field.

Style: Determines the style of the topic contained in a single-topic document. This property applies only to single-topic HTML5 or HTML source documents.

Condition

These can also be set and will appear in the Source ribbon group on page 109.

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** (**Project** tab > **Project** ribbon group > **Attributes** button) to create custom attributes.

Sharing

These fields only apply to documents that have been shared to a SharePoint Library.

Document URL: The location of this document on the SharePoint Library server.

Auxiliary Files: The location of this document's auxiliary files (usually graphics and video) on the SharePoint Library server.

First Uploaded by: The name of the person who first uploaded this document to the SharePoint Library server.

First Upload date: The date this document was first uploaded to the SharePoint Library server.

Last changed by: The name of the last person to edit this document.

Last download date: The latest date this document was downloaded from the SharePoint Library server.

Checked out by: If this document is currently checked out from the SharePoint Library, the name of the person who has it checked out. If it is you, it will read "me."

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 Search: If selected, Search is enabled for this document in NetHelp Targets. 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.

Renaming Topics

Single-Topic HTML source documents are renamed using the **Document Properties** dialog box.

To rename a topic

1. Select the document in the **Documents pane**. Open the **Document Properties** dialog box (see Setting Document Properties on page 297).
2. Click the **Rename** button located next to the **Title** field. The **Rename Topic** dialog box will open.
3. Change the **Title** of the topic. The **URL**, **ASCII Name**, **Link tag**, and **Keyword** will be changed by default to use the new title (see Topic Properties dialog box on page 321). If you would prefer any or all of these properties do not change, clear the check box(es).
4. Click **OK**.

To rename a topic authored in a Word document, see Adding, Renaming, and Deleting Topics on page 288.

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 on page 449 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 122. 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 of the Doc-To-Help 2014 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 168.

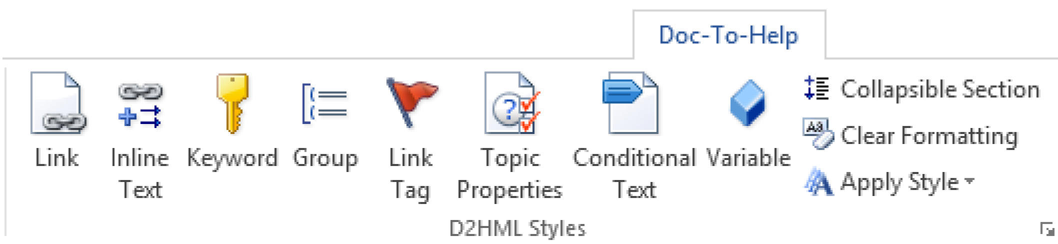
Doc-To-Help can create automatic links to each Glossary item, or you can create manual links to Glossary items. To create a manual link, see Creating Links on page 305. Choose the **Link Type** of **Glossary Term**. To set your project to automatically create glossary links, select the **Auto Glossary Links** check box in the **Project Styles** dialog box for the **Glossary Heading** Paragraph Style. See Paragraph Styles on page 171 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 291 for more information.

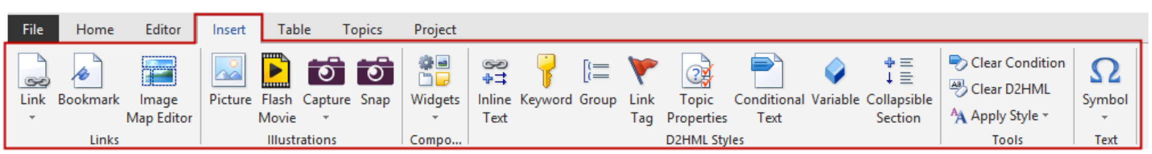
Using D2HML (Doc-To-Help Markup Language)

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.

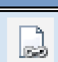
When editing documents in Microsoft® Word, Microsoft® FrontPage®, or Adobe® Dreamweaver®, you can use the appropriate **Doc-To-Help** ribbon, toolbar, or menu to quickly and easily apply D2HML styles:



When editing HTML5 documents in the Content Editor, 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 305 Create a hotspot link to a topic (jump or popup) Create a	C1HJump C1HPopup C1HKeywordLink C1HGroupLink	 	The topic list opened by a Keyword or G link can be displayed in a dialog box or menu. Index entries and groups can also be cre the Index and Groups pane on page 125 Doc-To-Help. Link Tags are created with the **Link Ta**

	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.			button. See Adding a Link Tag on page
Inline Text	Inline Text on page 308 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 "hot style/HTML; the text that will be display the hotspot is tagged with the "expandin style/HTML. The inline/expanding/drop-down text is invisible by default. See Showing Hidde Hotspots on page 317 for more informat
Keyword	Keyword on page 309 Designates a hotspot as an index keyword.	C1HIndex C1HIndexInvisible	 	The hotspot can be visible or invisible. U "Invisible" style/HTML to make the hot invisible. (The "Visible" check box in th Keyword dialog box controls visibility.) See Showing Hidden Hotspots on page 3 more information.
Group	Group on page 310 Designates a hotspot as a member of a group.	C1HGroup C1HGroupInvisible	 	If invisible the hotspot does not appear i output. See Showing Hidden Hotspots on page 3 more information.
Link Tag	Link Tag on page 311 Designates a hotspot as a link target.	C1HLinkTag C1HLinkTagInvisible	 	If invisible the hotspot does not appear i output. See Showing Hidden Hotspots on page 3 more information.
Topic Properties	D2HML Topic Properties on page 312 Sets a subset of the topic properties.	C1HTopicProperties		All topic properties can be controlled us Topic Properties dialog box in Doc-To-H The Topic Properties are invisible by de See Showing Hidden Hotspots on page 3 more information
Conditional Text	Conditional Text on page 313 Used to mark text as conditional by platform, target, attribute or any combination of	C1HConditional		Conditional text will also be tagged with platform, target, and/or attribute informa (platform=, target=, attribute=)

	all three.			
	[Variable](#) on page 315 Creates a hotspot to a text or rich content variable.	C1HVariable		The hotspot chosen is replaced by the va
	[Collapsible Section](#) on page 316 Creates a section that is either collapsed or expanded by default.	C1H Section Collapsed C1H Section Expanded	 	The section header will have either the C Section Collapsed or C1H Section Expa style applied. The text underneath the he will retain its original style.

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 168 for more information on editing and creating styles.
- See Editing a CSS on page 176 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 108. See Guide to Templates and Styles on page 7 for more information on Doc-To-Help's predefined templates and styles and how to work with them.

Creating Links

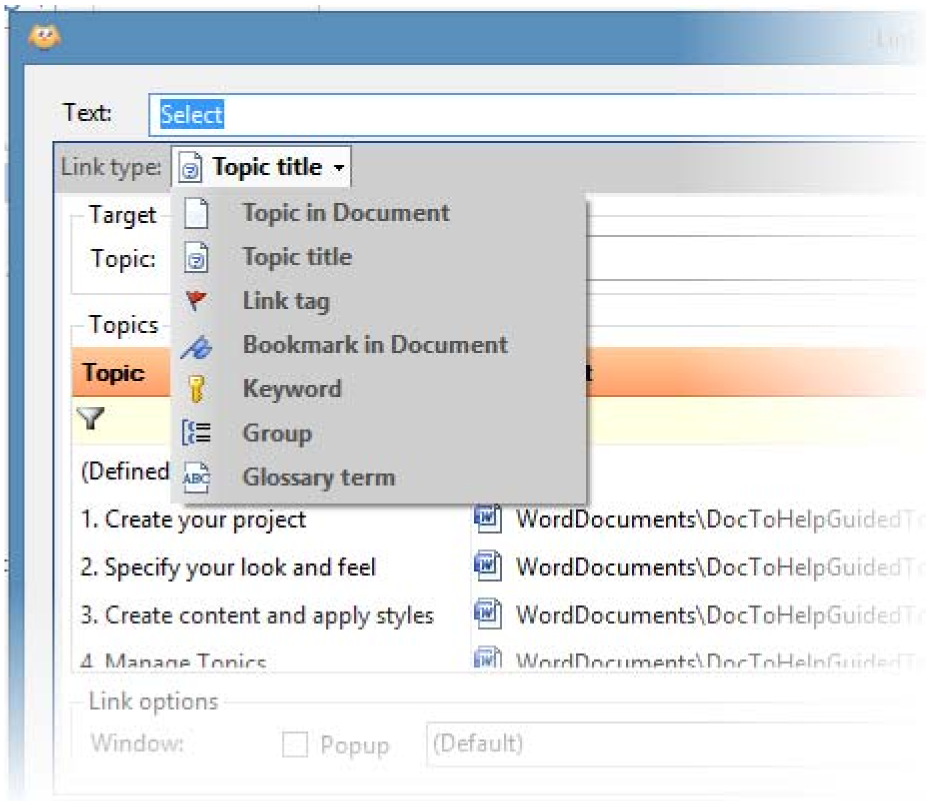
The **Link dialog box** is used to create links to topics, bookmarks, link tags, keywords, groups, and glossary entries.

Where to find the Link button 

- **Doc-To-Help's Content Editor:** the Insert tab on page 115 (Choose **In Project** from the drop-down.)
- **Microsoft® Word:** the Doc-To-Help toolbar or ribbon on page 285
- **Microsoft® FrontPage®,** and **Adobe® Dreamweaver®:** the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 Doc-To-Help's Content Editor and HTML Editors:

   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 309, Adding a Topic to Group on page 310, or Adding a Link Tag on page 311. You can also create/edit keywords and groups using the Index and Groups Pane on page 125 in Doc-To-Help.

Topic/Link Tag/Bookmark 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 168.

Keyword/Group link options

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 168.

Drag-and-Drop Linking

If you would prefer, you don't need to use the Link button on page 305 to create links to topics, keywords, groups, and documents — or the Variable button on page 315 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
- HTML5 documents (Doc-To-Help's **Content 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 305.

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 **Content 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 **Content** window tab. The **Content Editor** window will open.
3. Drop the item at the desired place in the **Content** 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 Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 317.

Note: In EPUB Targets, Expanding Text and Dropdown Text may be displayed or hidden. The option you prefer can be set in the **Help Targets** dialog box using the **Show expanding text** and **Show dropdown text** check boxes.

It is also possible to create expanding/collapsing sections in HTML Help, NetHelp, Microsoft Help Viewer, and Help 2.0 targets, see Creating Expanding/Collapsing Sections on page 316 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 125 in Doc-To-Help.
Links to keywords are created using the Link dialog box on page 305. 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 Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 Doc-To-Help's Content Editor and HTML Editors:

   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 317.)

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 317.

5. Click **OK**.

Please note that any new keywords created using this method will be added to the Index and Groups pane on page 125 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 327 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 125 in Doc-To-Help.

Links to groups are created using the Link dialog box on page 305. 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 Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 125 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 Doc-To-Help's Content Editor and HTML Editors:

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 317.)

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 317.

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 168.) 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 127 (right-click and choose **Link Tag** from the **Columns** menu).

Where to find the Link Tag button

- Doc-To-Help's Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 Doc-To-Help's Content Editor and HTML Editors:

   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 317.)

   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 317.

5. Click **OK**.

Setting D2HML Topic Properties

Specific topic properties may be set using D2HML; please see Viewing/Changing Topic Properties on page 321 for information about making adjustments to other topic properties.

Where to find the Topic Properties button

- Doc-To-Help's Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

Topic Properties will be hidden by default, see Showing Hidden Hotspots on page 317 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 321.

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. You can edit the name manually, or you can set a rule for it in the **Project Settings** dialog box, **URL mode** field. You can also limit the character length of this URL using the **Truncate file name length** field in the **Project Settings**.

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** 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, 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) 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. The default label is **More:**

Include in Search: If selected, Search is enabled for this topic in NetHelp Targets. 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 117.

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.

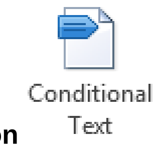
Attributes: Set an attribute-based condition for the selected topic. The topic will be included or excluded when creating conditional builds (for example, internal or external.) Use the **Attributes** dialog box (**Project** tab > **Project** ribbon group > **Attributes** button) to create custom attributes.

Disable comments: If checked, DISQUS commenting will be disabled for this topic. See Adding DISQUS Commenting to NetHelp 2.0 Targets on page 335 for more information.

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 3 easy to accomplish.

You can also mark entire documents and topics as conditional, see Setting Document Properties on page 297 and Setting Topic Conditions on page 325.



Where to find the Conditional Text button

- Doc-To-Help's Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 162 (Project tab > Project ribbon group on page 119 > **Attributes** button) to create custom attributes.
4. Set your **Options**. Options will vary slightly depending on your choice of editor.

    In Doc-To-Help's Content Editor and HTML Editors:

    - **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 129 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 290 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 Content 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 3 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 for more information about creating variables.

Where to find the Variable button

- Doc-To-Help's Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

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 hotspot 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 Doc-To-Help's Content Editor and HTML Editors:

   - **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 307 for more information.

Please note that if you are working in the Content 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.

Creating an Expanding/Collapsing Section

In online Help Targets, you can create sections that are expanded or collapsed by default. This is an ideal way to subdivide a long topic in online Help. The sections can be displayed as expanded or collapsed by default, and include “+” and “-” controls for end users next to each section and at the top of the topic.

Where to find the Collapsible Section button 

- Doc-To-Help’s Content Editor: the Insert tab on page 115
- Microsoft® Word: the Doc-To-Help toolbar or ribbon on page 285
- Microsoft® FrontPage®, and Adobe® Dreamweaver®: the Doc-To-Help D2HML Styles toolbar or ribbon on page 293.

To create expanding/collapsing sections

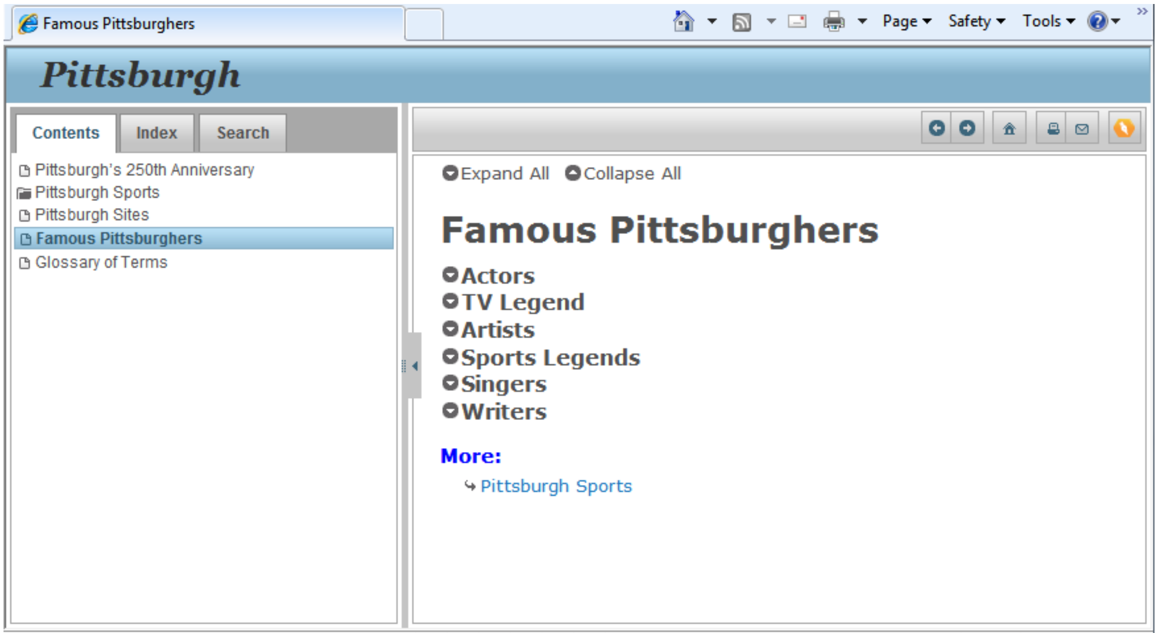
1. Select the header and the text that you would like to make into a section. (An example header: "Penn State" and then the text could be a description of the university.)
2. Click the **Collapsible Section** button. The **Collapsible Section** dialog box will open. The selected text will display in the **Header** and **Text** areas.
3. If you would like the section to be collapsed by default, select the **Collapsed** radio button in the **Options** area.

   For text expanded by default, select the **Expanded** radio button.
4. Click **OK**.
5. Continue creating sections. Note that the Heading for each section now has the **C1H Section Collapsed** or **C1H Section Expanded** style applied to it.

Note: The **Generate XHTML** check box in the **Help Targets** dialog box must be selected to enable expanding/collapsing sections.

The header will be displayed in the Target with a “+” and “-” control next to it. When the user clicks on the control, the text will be displayed or hidden. The "Expand All" or "Collapse All" control at the top of the topic will allow the user to expand/collapse all of the text at once.

This is an example of a topic with sections collapsed by default; Doc-To-Help automatically added the "Expand All" control at the top of the topic.



Note: In EPUB Targets, all content in Collapsible Sections will be displayed (just as it would in Manual Targets).

Clearing D2HML Styles

In Doc-To-Help's Editor, the **Clear D2HML** button will be on the Insert tab on page 115; 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 303 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

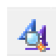
Invisible hot spot types, such as Invisible Keyword and Topic Properties, are not visible in help targets and certain source documents. The **Show Hidden Hot Spots** button available on the **D2HML Styles** toolbar in Microsoft® Word and Adobe® Dreamweaver® allows you to show any invisible hot spots 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 127, which works along with the Topics tab on page 117 and Index and Groups on page 125, Contents on page 124, and Related Topics panes on page 126.

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 127 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 297.

To open the Topic Properties dialog box

1. Choose the desired topic from the Topics window on page 127.
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. To change a topic title in a Word Source document, use the **Rename** button on the **Doc-To-Help** ribbon. In a single-topic HTML5 or HTML source document, use the **Document Properties** dialog box.

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).

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. You can edit the name manually, or you can set a rule for it in the **Project Settings** dialog box, **URL mode** field. You can also limit the character length of this URL using the **Truncate file name length** field in the **Project Settings**.

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** 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, 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) 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. The default label is **More:**

Hide Subtopic Links: If selected, the subtopic links will be hidden for this topic.

Include in Search: If selected, Search is enabled for this topic in NetHelp Targets. If cleared, this topic is excluded from the search.

Disable comments: If checked, DISQUS commenting will be disabled for this topic. See Adding DISQUS Commenting to NetHelp 2.0 Targets on page 335 for more information.

Condition

These can also be set and will appear in the Condition ribbon group on page 117. See Utilizing Conditions on page 160 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 (**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

Target menu: Before setting your module link properties, choose the appropriate modular help Target from the drop-down.

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, NetHelp, and Microsoft Help Viewer 1.x 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.
Use first topic as parent: For Microsoft Help Viewer modular hub projects, sets the first topic in the child project as the parent of the other topics in the child project.

Inherit Product name: For Microsoft Help Viewer modular hub projects, sets the **Product name** for the child projects to the same value as that of the hub project. The **Product name** value is set in the **Help Targets** dialog box of the hub project.

Inherit Book name: For Microsoft Help Viewer modular hub projects, sets the **Book name** for the child projects to the same value as that of the hub project. The **Book name** value is set in the **Help Targets** dialog box of the hub project.

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 186 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 117.
4. Choose **Print** from the list of options. The **Print** dialog box will open.
5. Select the options you'd prefer 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 117.
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 117.
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 117.
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**.

Note: Only the columns displayed will be printed. See Topics window on page 127 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 186 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 117.
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**.

Note: Only the columns displayed will be exported. See Topics window on page 127 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 117. 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 321 and the Condition ribbon group on page 117.

Attributes must be specified in the **Help Targets** dialog box (see Creating Help Targets on page 133) to be properly included/excluded from the final output. See Defining Attributes on page 162.

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 3 for more information.

By default, Related Topics will display at the bottom of an online Help topic like this:

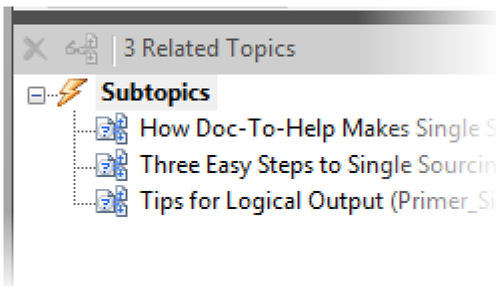
Using the **Related Topics** pane, you can add additional subtopic links to a topic, as well as disable or hide links.

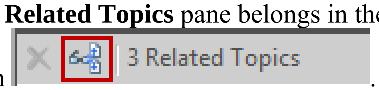
Automatic Subtopic Links

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 168 using the **Auto Subtopic Links** check box.

In the bottom section of the **Related Topics** pane, these automatic relations are flagged as "Subtopics."



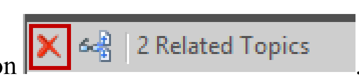
You can note where any topic displayed in the lower half of the **Related Topics** pane belongs in the topic hierarchy by selecting it and clicking the **Go to Related Topic** toolbar button 

.

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."

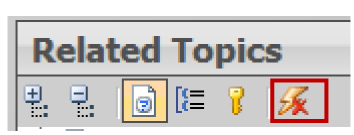
   This custom relation will display in the **Related Topics** list for that topic after the Target is rebuilt.

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 117 (**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 133 for more information.

If you would like to change the label for individual topics:

1. Open the **Topics** pane.
2. Select the topic in the **Topics** pane.
3. In the Topics tab on page 117 (**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 133 for more information.

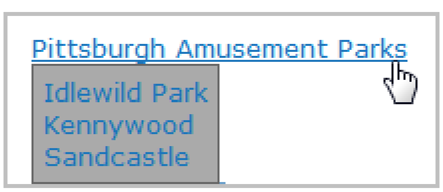
See Related Topics pane on page 126 for more information.

Creating an Index or Groups

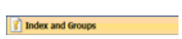
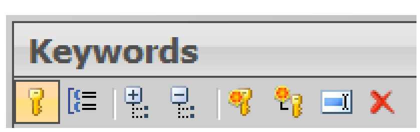
Indexes and Groups can be built quickly and easily using the **Index and Groups** pane and the **Topics** window.

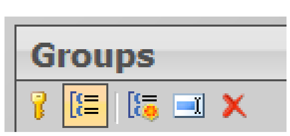
Index entries are also called keywords (or K-links). They will display in your Help or manual targets in an index. You can also create links to keywords.

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:

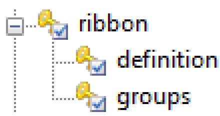
- Click **Index and Groups** 
- Click the **Keywords button** 

- 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 Content Editor). See Using D2HML on page 303 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 127 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 119 (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 168.

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 Content Editor. See Creating Links on page 305 for more information.

See Index and Groups pane on page 125 for a complete listing of options.

Watch the video: How to Create an Index (1:46)

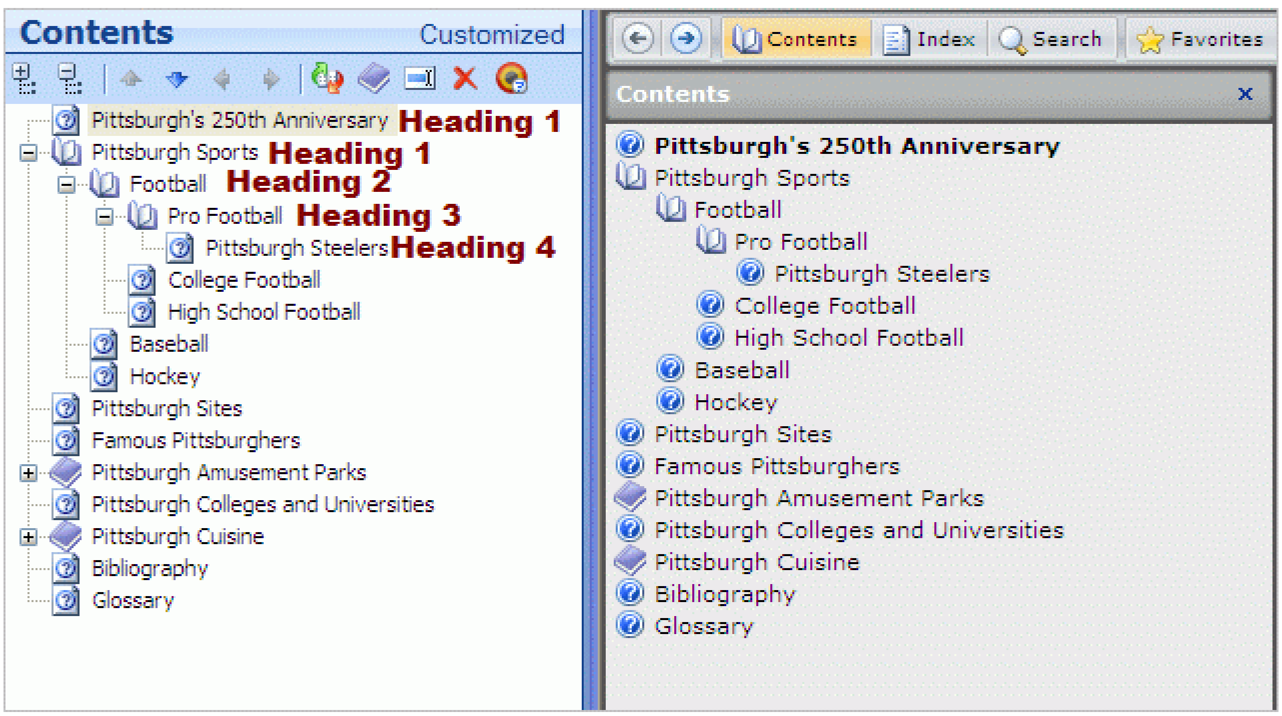
Creating a Table of Contents

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 122.
- 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 168 for more information.

You can see which topics are included in your table of contents by opening the Topics window on page 127 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 124.
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 must be added manually, unless in the Project Settings on page 182 dialog box the **Update customized table of contents in build** check box is selected. Choosing this option will automatically add newly-created topics to your customized table of contents when your project is built, although you will want to open the **Contents Pane** and confirm their location. To view the new topics added after a build, in the **Topics** tab of Doc-To-Help, click the **Filter View** button and choose **Show only topics added in the last build**. If you have deleted any topics, those will need to be deleted manually from the Table of Contents. **Target-Specific Tables of Contents** will not be updated.

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 124 for a complete listing of TOC options and instructions.

Watch the video: Customizing a Table of Contents (2:08)

Extending Targets

With Doc-To-Help, you can extend your NetHelp 2.0 Targets further in two ways. You can gather user feedback and build a user community by incorporating DISQUS into NetHelp 2.0 systems posted on the web. Plus you can embed NetHelp 2.0 Targets (the full target or single topics) into web pages with or without iframes.

Adding DISQUS Commenting to NetHelp 2.0 Targets

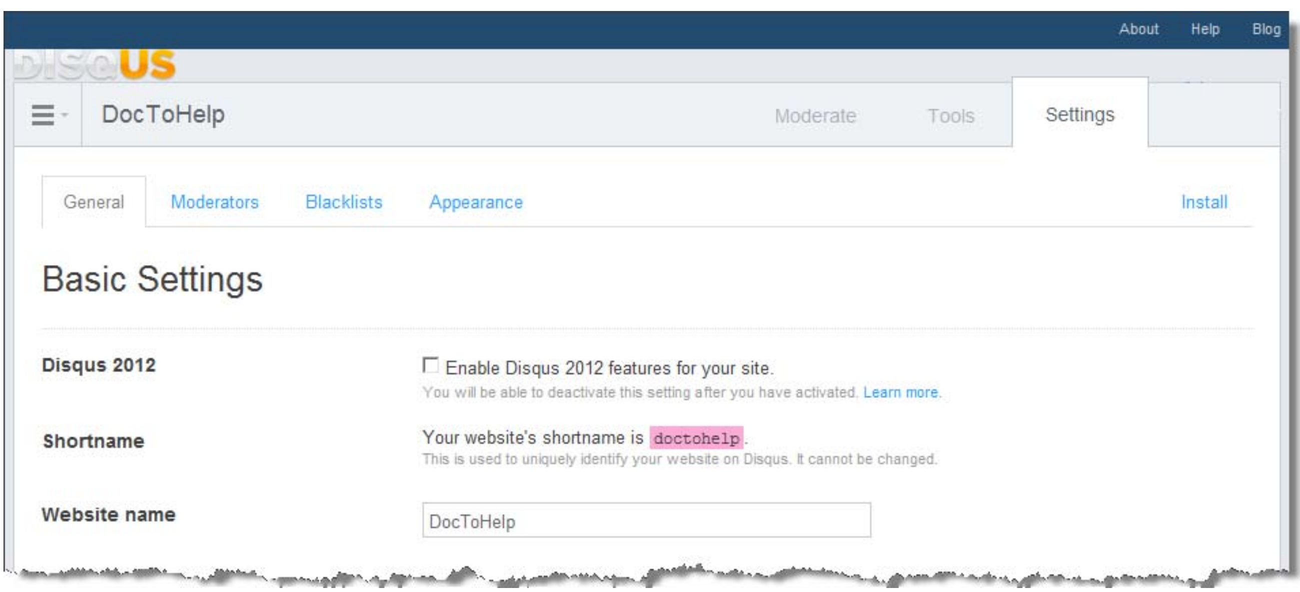
If you would like to gather user feedback for your Help system — such as commenting and ratings — and build a user community, you can do so by incorporating DISQUS in NetHelp 2.0 Targets. DISQUS (http://disqus.com/) is an online discussion and commenting service for websites that uses a networked platform and works on the following browsers: Internet Explorer, Firefox, Chrome, Safari, and Opera.

All settings, comment moderation, appearance, etc. are managed through your DISQUS account.

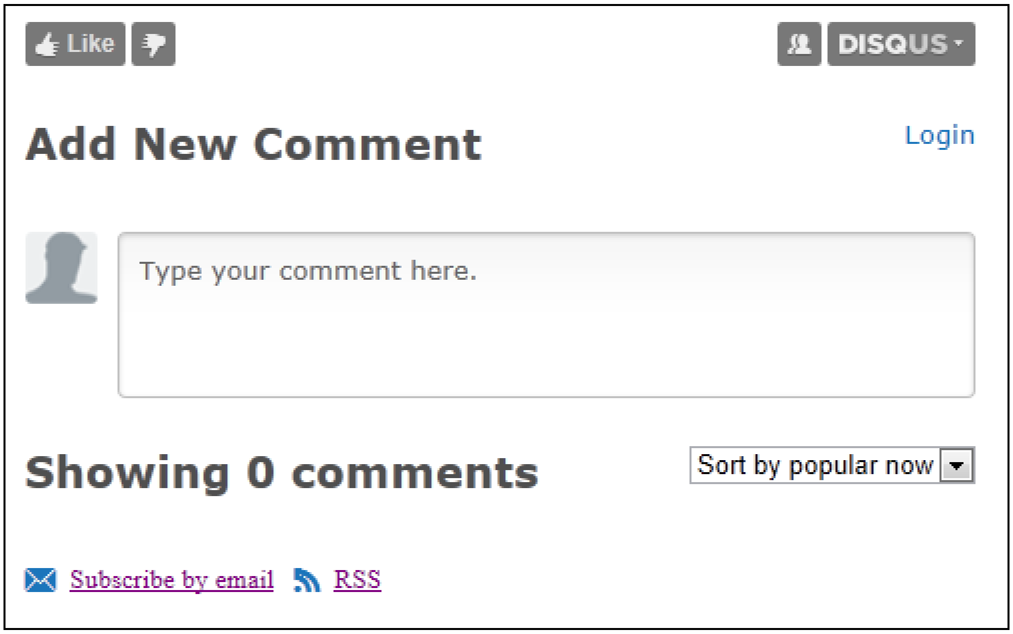
By default, DISQUS comments are enabled for every topic in your NetHelp 2.0 Target, but is possible to exclude specific topics, or entire topic types from commenting. Instructions to exclude topics or topic types are below.

To Enable DISQUS in NetHelp 2.0 Targets

1. Create a DISQUS account at http://disqus.com/register and define your DISQUS shortname.



2. In Doc-To-Help, open the **Help Targets** dialog box (Doc-To-Help **Home** tab, click the dialog box launcher on the **Target** ribbon group) and select the NetHelp 2.0 Target on the left. In the **Comments** section, set the **Use Comments** field to **DISQUS**, and enter your **DISQUS shortname**.
3. Click **OK** and click the **Rebuild Target** button.
4. Post your NetHelp 2.0 on your website. At the bottom of each topic, DISQUS commenting and rating options will appear.



If you would like to change the skin of your DISQUS comments; set up comment moderators and moderation rules; set the language, Twitter and Facebook features; or create blacklists, you can do so using your DISQUS account.

To Exclude Topic Types from DISQUS Commenting

1. In Doc-To-Help, open the **Project Styles** dialog. (**Project** ribbon group, click the **Topic Types** button.)
2. Select the **Topic Type** you would like to exclude from the pane on the left.
3. Clear the **Use Comments** check box.
4. Click **OK** and click the **Rebuild Target** button.

To Exclude Individual Topics from DISQUS Commenting

1. In Doc-To-Help, choose the desired topic from the Topics window on page 127. Right-click on the topic and choose **Properties**. The **Topic Properties** dialog box will open.
2. Check the **Disable Comments** check box.
3. Click **OK** and click the **Rebuild Target** button.

You can also exclude Topics while working in your Source Documents. If working in Word, click the **Topic Properties** button in the **Doc-To-Help** ribbon. In the **D2HML Topic Properties** dialog box, select the **Disable Comments** check box. In the Content Editor, the **Topic Properties** button is on the **Insert** ribbon.

See Incorporating the Wisdom of the Crowd with Doc-To-Help and DISQUS for more information.

Embedding NetHelp 2.0 Targets into Web Pages

Doc-To-Help's NetHelp 2.0 is architected to make it easy to embed a single topic, or an entire Help system, into a web page. This allows you to stream updated content to your website without web development.

There is some lightweight front-end configuration that must be made to your website. Your Help system does not need to be hosted on the same web server as your main site. In that case, you would add an iframe, because using an iframe makes it possible to place the web site and the NetHelp 2.0 target on different servers (domains). If both your Help system and your website will be hosted on the same web server, an iframe is not needed.

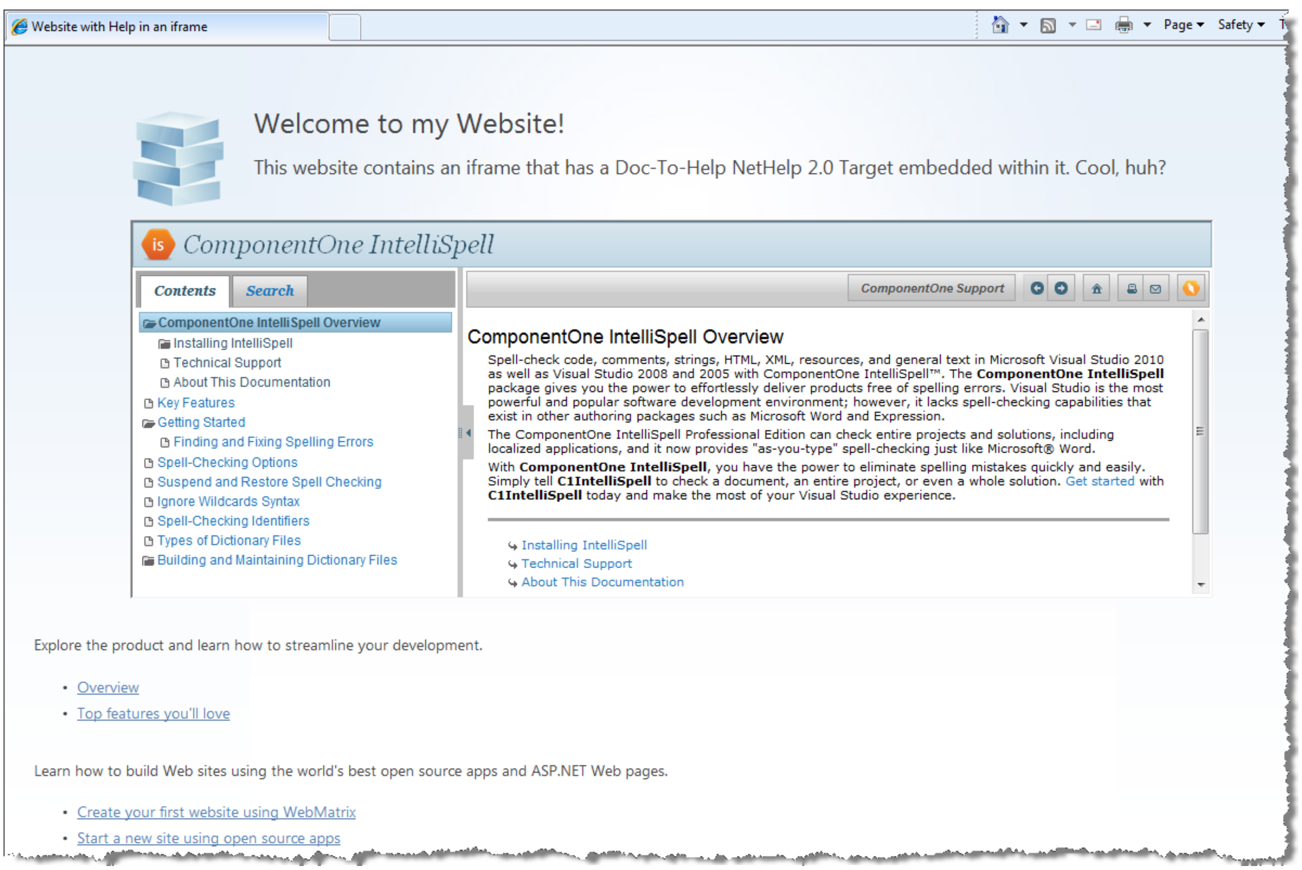
To Embed a NetHelp 2.0 Target in an iframe

This method allows to place the web site and NetHelp 2.0 target on different servers (domains).

1. Publish your built NetHelp 2.0 target to a server, for example, http://mysite.com/help/mytarget/ (The full Target URL to NetHelp 2.0 is: http://mysite.com/help/mytarget/index.html)
2. Add an "iframe" element to your webpage that contains the NetHelp 2.0 target. The iframe tag must have the "src" attribute set to the default NetHelp 2.0 page ("index.html") in the target folder. For example, to embed a full NetHelp 2.0 Target without parameters add:<iframe src="http://mysite.com/help/mytarget/index.html"></iframe>

The URL in the "src" attribute can be used to set NetHelp 2.0 target parameters if desired. These parameters are explained in NetHelp 2.0: Supported Parameters and How to Use Them on page 196

Here is an example iframe embedded in a website:



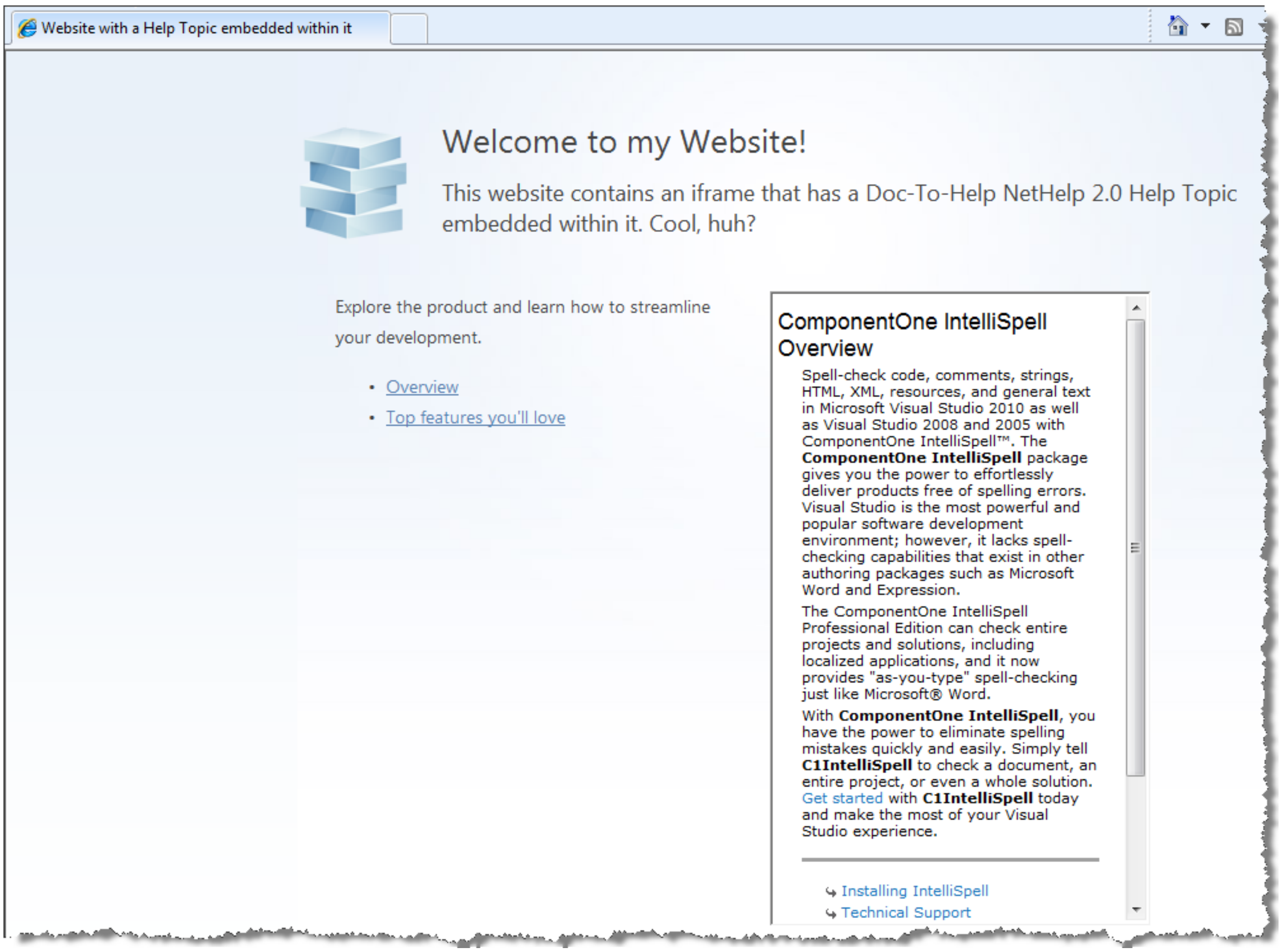
This is the code used:

```
<iframe
src="http://helpcentral.componentone.com/nethelp/c1intellispell/index.html"
width="1000px" height="350px"></iframe>
```

To embed a single topic on a web page, use:

```
<iframe
src="http://mysite.com/help/mytarget/index.html?topiconly=true"></iframe>
```

Here is an example single topic iframe embedded in a website:



This is the code used:

```
<iframe
src="http://helpcentral.componentone.com/nethelp/c1intellispell/index.html?topic
only=true" width="300px" height="500px"></iframe>
```

Generic example:

```
<!DOCTYPE html>
<html>
<head>
    <title>IFrame example</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
</head>
<body style="padding: 10px;">
    <p>Some text before help frame.</p>
    <!-- HELP FRAME -->
    <iframe src="http://mysite.com/help/mytarget/index.html"></iframe
width="800px" height="400px" >
    <!-- END OF HELP FRAME -->
    <p>Some text after help frame.</p>
</body>
</html>
```

To Embed a NetHelp 2.0 Target without an iframe

You can embed a NetHelp 2.0 Target into any HTML element (for example, into a DIV element).
This option is only available if the NetHelp 2.0 target and the container page (into which you embed the NetHelp 2.0 target) are placed in the same domain.

1. Publish your built NetHelp 2.0 target to a server, for example, http://mysite.com/help/mytarget/
   (The full Target URL to NetHelp 2.0 is: http://mysite.com/help/mytarget/index.html)
2. Add a container element (any HTML element) to your page, which will contain the NetHelp 2.0 target.

   The container must have the CSS property "position" set to either the "relative", "absolute" or "fixed" value. This is a restriction of the Doc-To-Help standard themes (Tabs and Accordion). You must also set the height for the container if you want to display NetHelp 2.0 target in full mode (you do not need to do this for topic only mode).

   For example, <div id="mytarget" style="position:relative; height:500px;"></div>

   If you want to display the NetHelp 2.0 target in topic only mode, the container element may look like this: <div id="mytarget"></div>
3. Add the "script" tag to the "head" section of the page. This script tag must have the "src" attribute set to the "js/nethelp.connect.js" file located in the NetHelp 2.0 target folder. You can also set the following optional attributes for this tag to set up the target:

   - "data-placeholder"—specifies the jQuery-selector of the container element created in step 2. If this attribute is missed then the NetHelp 2.0 engine will search for an element with the attribute "data-c1-role" equal to the "nethelp" value, if no such element is found the "body" element will be used as a container for the NetHelp 2.0 target. The NetHelp 2.0 target is placed into a container element and replaces all its content.
   - "data-start"—specifies the URL of the topic that will be shown instead of the default topic when the NetHelp 2.0 target is loaded.
   - "data-topiconly"—if set to "true" the NetHelp 2.0 target will be displayed in the topic only mode (without header, toolbars, and Contents, Index and Search tabs).
   - "data-settings"—specifies URL of the NetHelp 2.0 target settings file (by default it is the "settings.xml" file in the root target folder).

- “data-responsive”—used only for Responsive themes, set to “true”.

For example:

For the Tabs and Accordion Themes

```
<script type="text/javascript" src="/help/mytarget/js/nethelp.connect.js" data-
placeholder="#mytarget"></script>
```

For the Responsive Theme

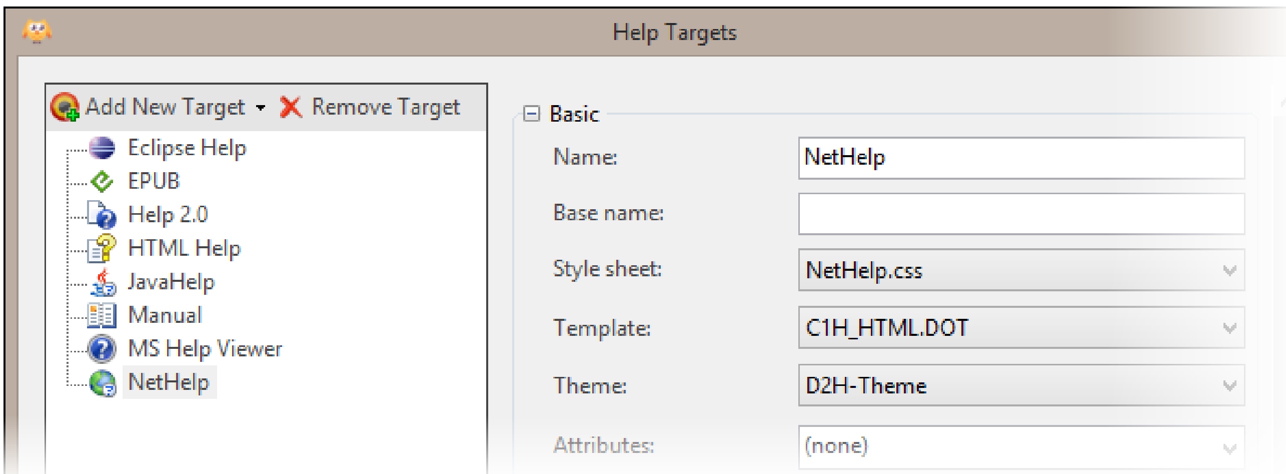
```
<script type="text/javascript" src="/help/mytarget/js/nethelp.connect.js"data-
placeholder="#mytarget" data-responsive="true"></script>
```

Note: If you are viewing the web site from the local file system, the NetHelp 2.0 target folder must be located in a subfolder of your site. Otherwise your browser may block the NetHelp 2.0 files.

Generic Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline example</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <script type="text/javascript" src="/help/mytarget/js/nethelp.connect.js"
        data-settings="/help/mytarget/settings.xml"
        data-start="/help/mytarget/Documents/home_topic.htm"></script>
</head>
<body style="padding: 10px;">
    <p>Some text before help frame.</p>
    <!-- HELP FRAME -->
    <div data-c1-role="nethelp" style="position:relative; width:800px;
height:400px;"></div>
    <!-- END OF HELP FRAME -->
    <p>Some text after help frame.</p>
</body>
</html>
```

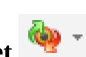
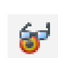
Building a Target



Doc-To-Help can create nine different types of Targets (outputs): NetHelp (including Responsive), HTML Help, EPUB, Eclipse Help, JavaHelp, WinHelp, Manual, Microsoft Help Viewer, and Microsoft Help 2.0. For details about each type of Target, see Doc-To-Help Outputs and Deliverables on page 13.

Target names and other details are customized using the **Help Targets** dialog box. See Creating Help Targets on page 133 for more information.

To Build a Target

1. Save your source documents. You may close them if you wish.
2. On the **Home** tab, click the **Select Target** button and choose the Target you would like to build.
3. Click the Build **Target**  button, or the **Rebuild Target**  button.
4. To view the Target, click the **View Target** button . The **Output window** will display the build status and any unresolved links and errors.

Use the **Cancel Build** button to stop a build, and the **Build Log**  button to open the **Output** window.

You can schedule your builds so that they run on a one-time, daily, or weekly basis. See Scheduling Builds with the Build Scheduler on page 342.

Viewing NetHelp Targets in Chrome and Opera: If your default browser is Chrome or Opera, Doc-To-Help will open your project when you click the **View Target** button, but if you already have Chrome or Opera open, it will prompt you to close them before displaying your NetHelp project. If you attempt to open the NetHelp Target locally outside of Doc-To-Help (by double-clicking on the index.html file) the Target will not display in Chrome or Opera. This is due to limitations with those browsers. Once NetHelp is deployed on a web server, it will open in Chrome and Opera.

Viewing EPUB Targets: Your EPUBs will open in your default reader (such as Calibre) after they are built. To set the default EPUB reader in Doc-To-Help, go to **File** tab > **Tools** > **Options**. In the **Options** dialog box, select the **Viewers** button and choose the EPUB viewer executable.

Note about Word Documents: You can edit Word documents — even the source documents in your project — while Doc-To-Help builds a Target. If you choose to edit a source document that has been shared to SharePoint, it can only be edited locally and all automatic operations with the SharePoint server are disabled.

"Build" vs. "Rebuild"

- When you choose the **Build Target** button, Doc-To-Help recognizes all the Source documents that have been edited since the last build and makes those changes in the Target. This is the quickest way to build a project.
- When you choose the **Rebuild Target** button, Doc-To-Help deletes all of the content in the output folder of your Target, then builds everything from scratch (not just Source documents). If you make any global changes to your project — changing project settings, creating variables, Glossary entries, etc., you should do a rebuild. A rebuild will always take longer than a build, especially on large projects. You should always do a rebuild before reviews and releases.

Some notes on Manual Targets:

- When you build your Manual Target, you can build a PDF version at the same time. To do so, open the **Help Targets** dialog box and select the **Generate PDF** check box for your Manual Target. You can then view your PDF by clicking the **View PDF** button on the **Home** tab after the build. Projects authored in Microsoft Word 2007/2010/2013 will use Word's built-in PDF converter. If using Word 2007, the built-in converter will only be used if the "2007 Microsoft Office Add-in: Microsoft Save as PDF" is installed http://www.microsoft.com/downloads/en/details.aspx?FamilyID=f1fc413c-6d89-4f15-991b-63b07ba5f2e5&displaylang=en.
- Please note that when you view the Word version of your Manual target, the resulting file will have a different **Doc-To-Help** ribbon than the Source documents. In the Target document, the ribbon will contain only the basic styles and tools necessary to make final adjustments to your manual before printing or conversion to PDF, such as the **Cross-Reference** button (see Inserting Cross References on page 289 for more information) and the **Margin Notes** button (see Creating Margin Notes on page 289 for more information).

Watch the videos: How to Build a Target (1:53); How to Deliver a Target (1:01)

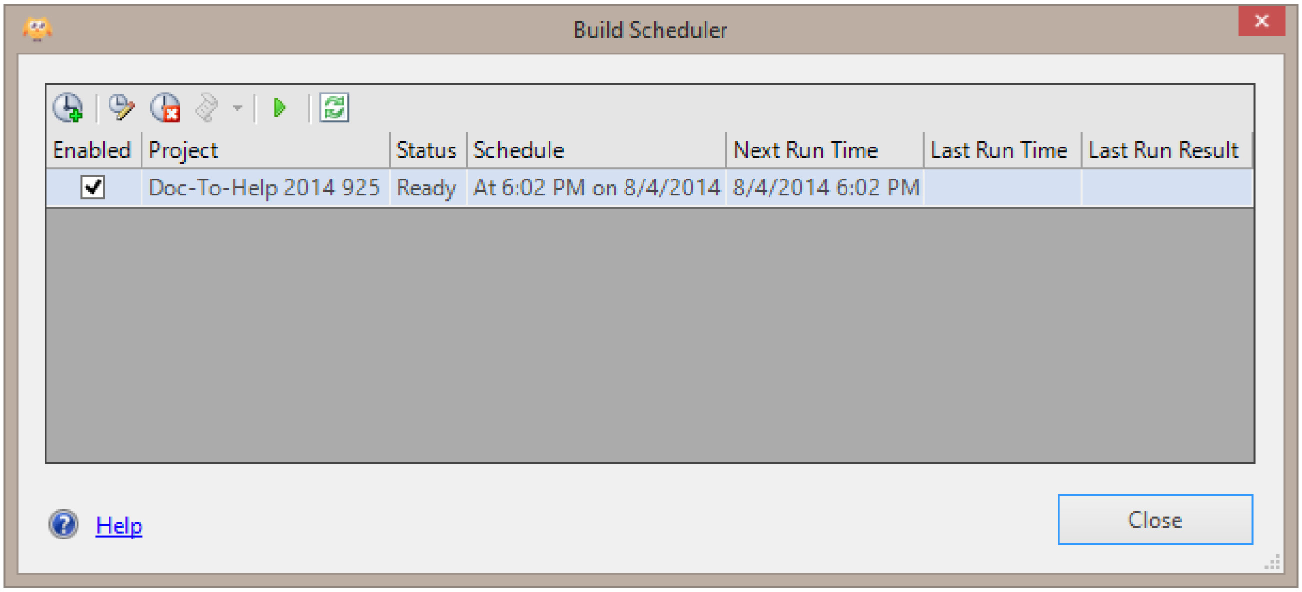
Scheduling Builds with the Build Scheduler

Using Doc-To-Help's **Build Scheduler**, you can schedule and monitor automatic builds of Targets. One-Time, Daily, and Weekly builds can be scheduled.

All of the scheduled builds created display in the **Build Scheduler** window. Only those with a check mark in the **Enabled** column will be run. The project name, status, build time, the time of the next and the last build, and the result of the last build are displayed for each scheduled build.

You can build all of the Targets for a project from a single build schedule.

The toolbar in the **Build Scheduler** allows you to create, edit, and delete builds; view build logs, run any build on the spot, and refresh the window.



Using the Build Scheduler

- Click the **Create a New Task** button to create a new build schedule. See Setting Up a Scheduled Build on page 344.
- To edit a build schedule, select it, and click the **Edit the Selected Task** button.
- To delete a task, select it and click the **Delete Selected Task** button.
- To view the build log of a task, select it and click the **View Build Log of Selected Task** button.

  After your build tasks are run, the log files are stored within your project's Temp\BuildLogs folder. Every Target will have a separate folder, and the log file names will have the "schedule_" prefix, for example: "schedule_2010-11-15_16-60-50.log."
- To run a task immediately, select it and click the **Run Selected Task** button.
- Click the **Refresh Task Status** button to refresh the task list.

To access the Build Scheduler

- Click the **File** tab and choose **Tools** > **Build Scheduler**.

Note: When the scheduled build for a project is building, you can not open that project. If you open a project and it is scheduled to build within that time frame, the scheduled build will not begin (it will fail). You can open, edit, and build any other Doc-To-Help project while the scheduled build of another project is building.

Windows Task Scheduler must be started for Doc-To-Help's Build Scheduler to run. To check the status of the Windows Build Scheduler:

- **Windows Vista and 7/8:** Click Start > Control Panel > System and Security > Administrative Tools > Scheduled Tasks

Setting Up a Scheduled Build

Using the **Schedule Build** dialog box, you can set up automatic builds of your Targets. You only need to create one build schedule for each project.

To schedule a build

1. Open the Build Scheduler (**File tab** > **Tools** > **Build Scheduler**)
2. Click the **Create a New Task** button. The **Schedule Build** dialog will open.

   You can open the **Schedule Build** dialog directly from a Doc-To-Help project. Open the project and from the **Home** tab, click the drop-down arrow at the bottom of the **Build Target** or **Rebuild Target** buttons. Choose **Schedule Build** from the options.
3. In the **Project** field, click the ellipsis button and choose the project.
4. Select the check box next to each **Target** in that project you would like to build.
5. Choose the **Build type**. For an explanation of the difference between **Build** and **Rebuild**, see Building a Target on page 341.
6. Choose the **Schedule type**. You have the choice of **One time**, **Daily**, and **Weekly**.
7. Choose the **Start Time** and date. If you have chosen a **Daily** or **Weekly** build, you can then choose when the build should recur.
8. Click **OK**. If you wish to change this schedule, open the **Build Scheduler**, choose this task from the list, and click the **Edit the Selected Task** button. To delete this task, choose it and click the **Delete Selected Task** button. See Scheduling Builds with the Build Scheduler on page 342 for more options, such as viewing build logs.

If you choose to schedule daily builds, the default is for the Targets to build every day at the chosen time (Recur every 1 days). If you would prefer to choose the exact days your builds will occur (for example, every weeknight, choose the **Schedule type** of **Daily** and select the days of the week).

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.**

If you need to start you build from Windows Services (such as the Task Scheduler or TFS Build) you should use a different executable, **C1D2HAgent.exe**. See Building Help Targets using Windows Services on page 346 for more information.

Both of these executables are located in the following default folders:

- 32-bit machine: C:\Program Files\MadCap Software\DocToHelp
- 64-bit machine: C:\Program Files (x86)\MadCap Software\DocToHelp

The commands are as follows:

Command	Description	Example
c1d2hbatch –build	Rebuilds the current Help target entirely.	c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp" -p
c1d2hbatch –make	Updates the current Help target.	c1d2hbatch -make "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual"
c1d2hbatch –build or make -lLogFile (optional)	This option will save the build log file with a specific name, rather than the default name, which is equal to the current date and time.	c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual" -lmyfilename.log Build log will be stored in the Temp\BuildLogs directory.

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 128:

  ```
  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 128:

  ```
  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 for more information.

Note: In Windows 7/8 and 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 7/8 and 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.

Building Help Targets using Windows Services

You can build Doc-To-Help Targets using Windows Services such as the Task Scheduler or TFS Build. This is done using **C1D2HBatch.exe** and **C1D2HAgent.exe**, which can be found at:

- 32-bit machine: C:\Program Files\MadCap Software\DocToHelp
- 64-bit machine: C:\Program Files (x86)\MadCap Software\DocToHelp

1. An interactive user should log on and launch **C1D2HAgent.exe**.

   C1D2HAgent.exe must be run on the same machine as the Windows Service. An interactive user is a user that has the rights to interact with the machine using the keyboard, mouse, etc. (Services are generally not interactive.)

2. Start **C1D2HBatch.exe** with the argument -buildagent.

   For example: "C:\Program Files\MadCap Software\DocToHelp\C1D2HBatch.exe" -build "C:\Users\InteractiveUser\Documents\My Doc-To-Help Projects\Samples\EmployeeHandbook\EmployeeHandbook.d2h" "Employee Handbook Manual" -buildagent

   This can be started by any Windows service.

Please note: The interactive user who started **C1D2HAgent.exe**:

- Must be logged in when C1D2HBatch.exe is running.
- Must have read and write permissions for the project folder and its subfolders

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 **File** tab.
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.

Collaborating with SharePoint

Doc-To-Help's integration with Microsoft SharePoint makes it easy to collaborate with your team, as well as across your company. There are two collaboration options:

- Upload your content to a **SharePoint Library** and take advantage of SharePoint's management and workflow features. Doc-To-Help works with SharePoint's check in/check out features and permissions, so you get a complete source control and version control system. See Uploading and Working with Documents in a SharePoint Library on page 353.
- Upload your content to a **SharePoint Translation Library** (SharePoint 2010 only) and leverage SharePoint's translation workflow. Doc-To-Help will automatically create a project for each language you need to translate to. This provides an easy way to track translations, provide content to translators who do not use Doc-To-Help, and create localized targets. See Uploading and Working with Documents in a SharePoint Translation Library on page 360.

Note: Doc-To-Help has two other SharePoint publishing options. You can publish NetHelp output to a Sharepoint Document Library, or a SharePoint Wiki Library. See Publishing to SharePoint on page 373.

SharePoint collaboration is compatible with Microsoft® SharePoint® 2007, 2010, and 2013, as well as Office 365 (SharePoint Online). SharePoint 2010 or 2013 is required when publishing Doc-To-Help Targets to a SharePoint Wiki Library. SharePoint 2010 is recommended when working with Translation Libraries (SharePoint 2013 does not include this feature). Translation Libraries cannot be managed in Windows SharePoint Services 3.0, because WSS 3.0 does not support them.

The **Options** dialog box has a new tab named **SharePoint** that you can use to set your preferences. To open this dialog box, click the **File** tab and choose the **Doc-To-Help Options** button. See Setting Doc-To-Help Options on page 34 for more information.

Uploading and Working with Documents in a SharePoint Library

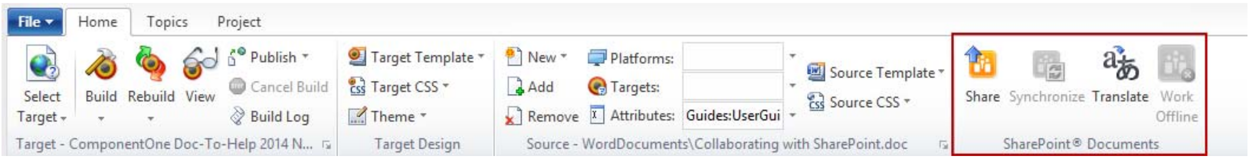
Uploading your content to a **SharePoint Library** allows you to take advantage of SharePoint's management and workflow features. Doc-To-Help works with SharePoint's check in/check out features and permissions, so you get a complete source control and version control system.

A Wizard will take you through the steps of uploading your documents to a SharePoint Library. After they are uploaded they can be checked in and out of the SharePoint Library from Doc-To-Help using right-click menus – it is not necessary to open SharePoint to access your documents.

Doc-To-Help automatically checks for document updates on your machine and those in the SharePoint Library and suggests documents that need to be synchronized. You can do this manually by clicking the **Synchronize** button. If you are using Word source documents, you will also have the option to merge documents if the local Doc-To-Help copy and the SharePoint Library copy differ.

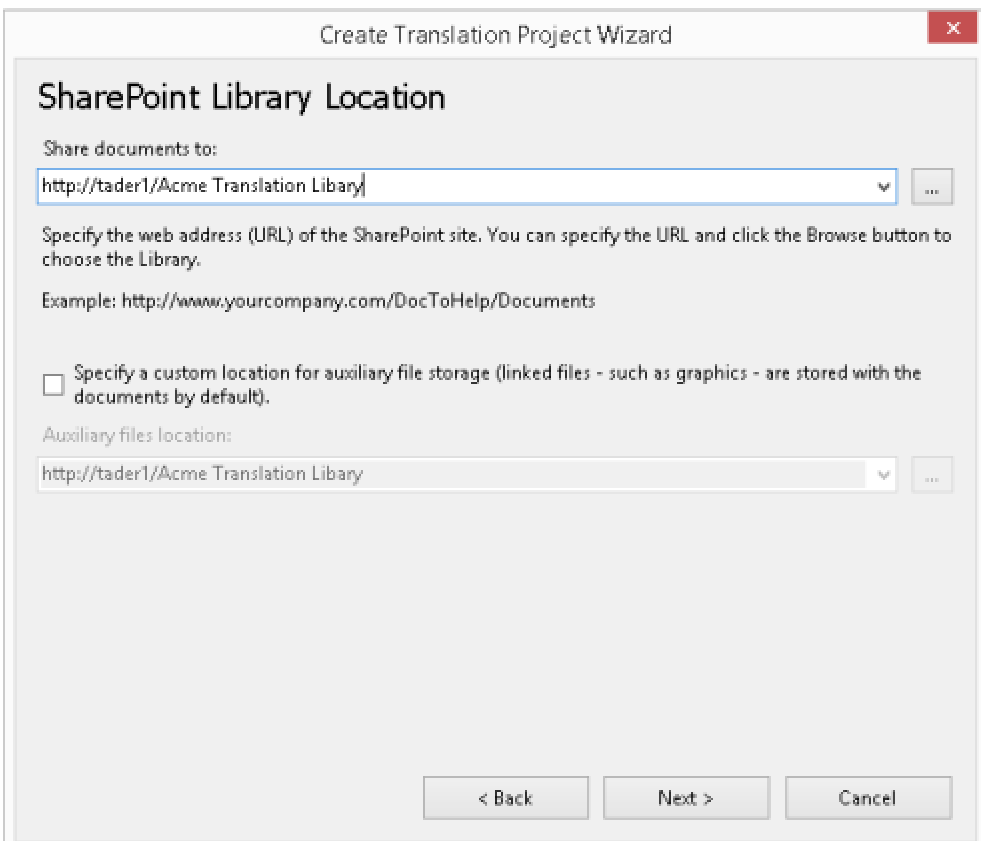
You can temporarily “turn off” synchronization using the **Work Offline** button.

See [Working with Shared Documents] on page 356 for more information.



Uploading documents to a SharePoint Library

1. Open the Doc-To-Help project that contains the documents you want to share.
2. Click the **Share** button in the Doc-To-Help **Home** tab (**SharePoint Documents** ribbon group). The **Document Share Wizard** will open. Click **Next**.
3. Choose the SharePoint Library you want to share to. Enter the SharePoint site URL in the **Share Documents to** field. You can choose a specific folder, or create a new folder for the documents by clicking the ellipsis button and using the **Select Server Folder** dialog box.



Your graphics and other auxiliary files (if any) will be stored to the same folder, unless you specify another location using the **Specify a Custom Location for Auxiliary file Storage** check box.

If you wish to upload the documents, but not have them retain any connection to SharePoint, clear the **Keep documents connected to SharePoint** check box.

4. Click **Next**. Choose the documents you would like to share. If you select the **Documents** folder check box, a subfolder named **Documents** will be created. Click **Next**.
5. Confirm your choices and click **Next**. The documents will then be uploaded.
6. Click **Finish** to close the Wizard.

   If there were graphics or other auxiliary files in the project, the **Synchronize Documents** dialog box will display. Click **OK** to upload these files to the SharePoint site. (By default, they will be stored in the **Media** subfolder.) Once uploaded, you can close the dialog box.

   In the **Documents** pane, the shared documents will now display a “hand” on the document icon.

To upload an individual document, right-click on it in the **Documents** pane and choose **Share Document to SharePoint Library**.

SharePoint Password Note: If you change your SharePoint password after uploading documents, you will no longer be able to synchronize your documents. You can fix this by clearing your Windows credentials.
1. In Windows, open the **Control Panel** > **User Accounts and Family Safety** > **Credential Manager**. (This is the path for Windows 7.)
2. Find the SharePoint Server credentials and remove them.
3. Restart Doc-To-Help.
You will be prompted to enter your new password the next time you try to connect to SharePoint.

Note: You cannot share Sandcastle plug-in docs, or .d2h files.

Adding SharePoint Library Documents to a Doc-To-Help Project

You can add any existing document stored in a SharePoint Library to your Doc-To-Help projects. The document can remain in its existing Library after import for check in/check out, or it can be downloaded with no connection to SharePoint.

Importing existing documents from SharePoint into your Doc-To-Help project

1. Open your Doc-To-Help project.
2. In the **Documents** pane, click the **Add Existing Documents** button .

    The **Document Import Wizard** will open.
3. Choose the **SharePoint** option. Click **Next**.
4. Enter the document’s SharePoint URL. Click **Next**.
5. If you wish to import the document(s) without retaining their connection to SharePoint, clear the **Keep documents connected to SharePoint** check box.
6. Choose the document(s) you would like to import using the check boxes. The documents will automatically be copied to the **Documents** folder in your project by default. You can change the folder location using the **Copy files to** ellipsis button. Click **Next**.
7. Confirm your choices and click **Next**. The document(s) will then be imported.
8. Click **Finish** to close the Wizard.

Use the **Move Up**/**Move Down**/**Move In**/**Move Out** buttons to rearrange the document order in the **Documents** pane.

Working with Shared Documents

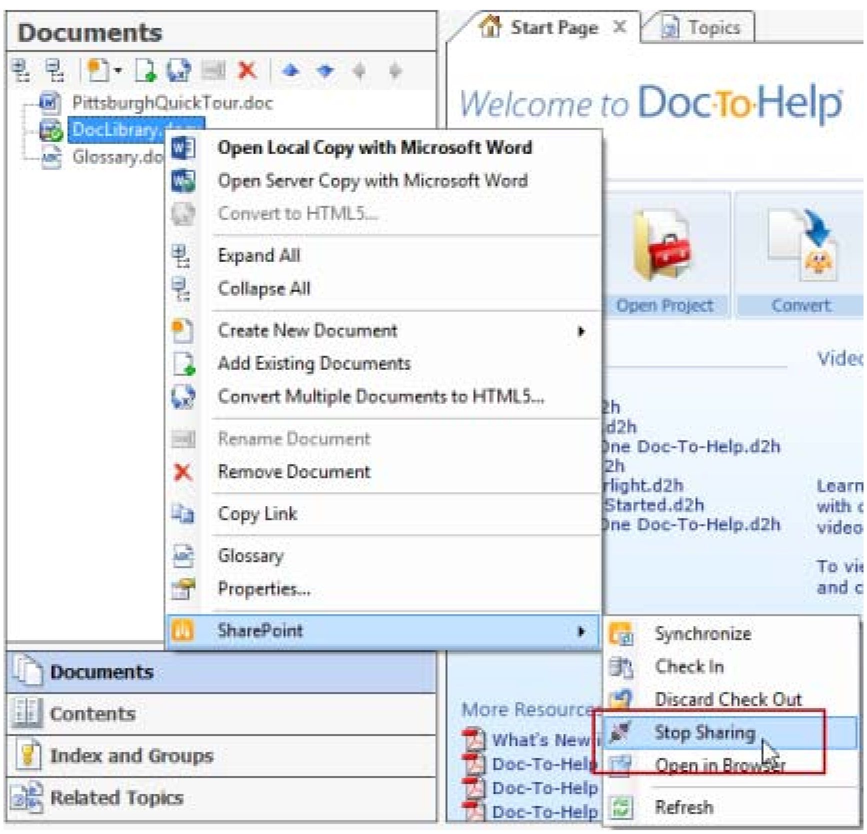
Once your documents have been shared to or downloaded from SharePoint Libraries, you can begin to work with them.

Documents can be:

- Checked in and out of the SharePoint Library
- Synchronized with the SharePoint Library
- Merged (Word only)
- Taken offline from the SharePoint Library
- No longer shared with the SharePoint Library

Once a document has been shared, it cannot be renamed or converted to HTML5.

Note: You can edit Word documents — even the source documents in your project — while Doc-To-Help builds a Target. If you choose to edit a source document that has been shared to SharePoint, it can only be edited locally and all automatic operations with the SharePoint server are disabled.



Checking documents In and Out of the SharePoint Library

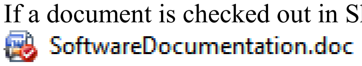
By default, Word documents are automatically checked out of the SharePoint Library when you open them in Doc-To-Help and begin editing. If you would like to change this default, clear the **Check out Word documents when editing local copy** check box in the **Options** dialog box (click the **File** tab and choose the **Doc-To-Help Options** button, **SharePoint Documents** tab.)

1. Open your Doc-To-Help project.
2. In the **Documents** pane, select the document you would like to check out.
3. Right-click and choose **SharePoint** > **Check Out** from the menu. The document icon will indicate that the document is checked out 

.

   In SharePoint, the document will be also be flagged with an icon 

   You can open the SharePoint Library by right-clicking on the document and choosing **SharePoint > Open in Browser.**
4. After making your edits, save your document and close it. Then right-click on it and choose **SharePoint** > **Check In**.

If a document is checked out in SharePoint by another user, there will be a red checked out icon next to it 

You cannot check out these documents. If you would like to know who has it checked out, right-click on it in the **Documents** pane and choose **Properties**. In the **Sharing** section, see the **Checked out by** field. You can also hover over the document name and the information will display in a tooltip.

Building Targets

You can build Targets even if source documents are checked out of the SharePoint Library by you or another user. However, if there are conflicting document versions that need to be merged, when you attempt to build the Target, the **Synchronize Documents** dialog box will open and list the conflicting documents. You can then resolve the issue before you build. If using Word source documents, Doc-To-Help will give you the option to merge the documents. See Merging Documents on page 358 for more information.

Undoing a Check Out

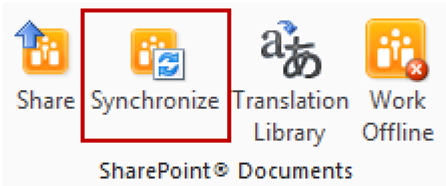
If you check out a document and make changes to it, then decide you do not want to retain the changes, right click on it in the **Documents** pane and choose **SharePoint** > **Discard Check Out**.

Synchronizing Documents

Doc-To-Help automatically synchronizes the documents changed in the SharePoint Library with those on your machine. Synchronizing uploads all local changes to the SharePoint Library and downloads SharePoint Library changes to your machine.

Synchronization should NOT be used as a substitute for checking documents in and out of the SharePoint Library. When check in/check out is used, all users know the status of documents and who has possession of them.

The automatic synchronization will occur when you close or open a document or project, and when you build a project. (You can change the synchronization defaults using the **Options** dialog box, **SharePoint Documents** tab. To open this dialog box, click the **File** tab and choose the **Doc-To-Help Options** button.)



You can synchronize manually at any time by clicking the **Synchronize** button on the **Home** tab of Doc-To-Help.

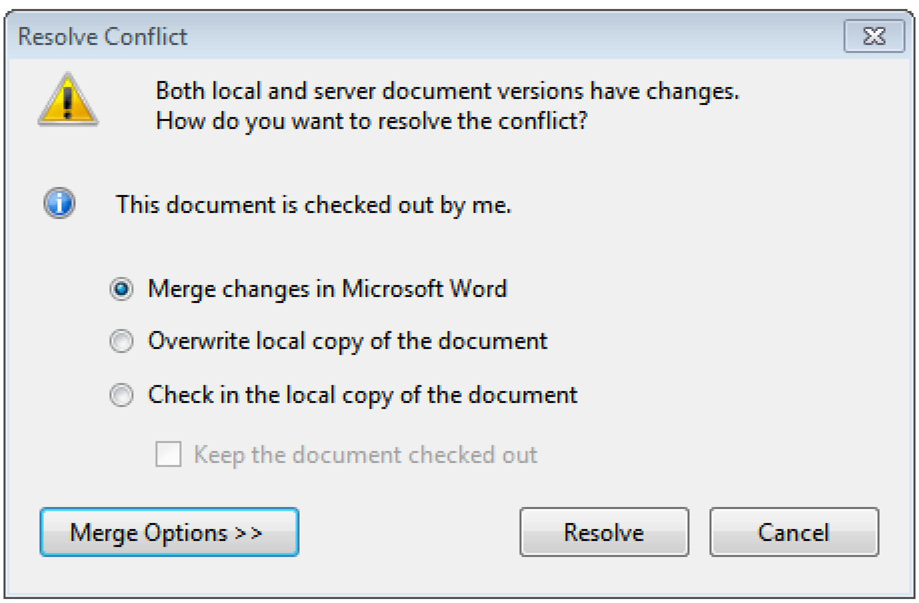
If both versions of a document (the SharePoint Library version and the version on your machine — the local copy — are different, and you trying to save, open, close or synchronize a single Word document a dialog box will give you the following options:

- Overwrite the local copy of the document
- Overwrite the copy of the document on the server (in the SharePoint Library)
- Merge the documents (Word documents only). See Merging Documents on page 358 for more information.

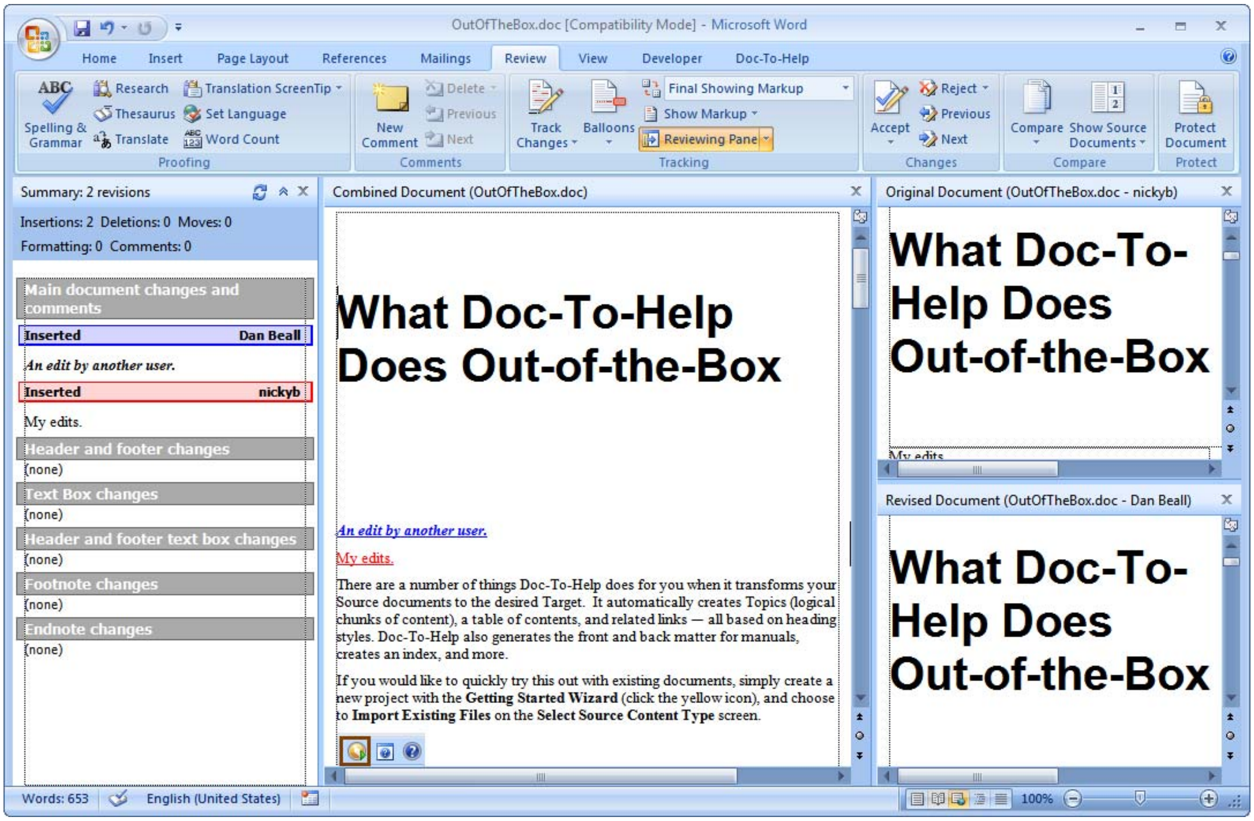
Merging Documents (Word documents only)

If both versions of a document (the SharePoint Library version and the version on your machine — the local copy— are different), and you trying to save, open, close or synchronize a single Word document, a dialog box will give you the option to overwrite one of the documents, or merge the two documents.

If you wish to merge the Word documents, choose **Merge changes in Microsoft Word** in the **Resolve Conflict** dialog box.



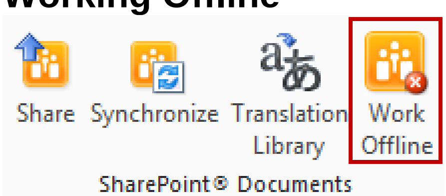
After you click the **Resolve** button, Word will display the changes to the document. Make the necessary corrections (accept all the revisions or only some of them; make additional edits) and save the document.



After you save the document, Doc-To-Help will suggest that you synchronize the changes. If you merged to the local copy, then the merged document will be uploaded to the SharePoint Library. If you merged to the SharePoint Library, the merged document will be downloaded to your local machine.

If you do not resolve conflicting document versions, when you attempt to build a Target, the **Synchronize Documents** dialog box will open and list the conflicting documents. Right-click on the document name in the **Synchronize Documents** dialog box to view the options.

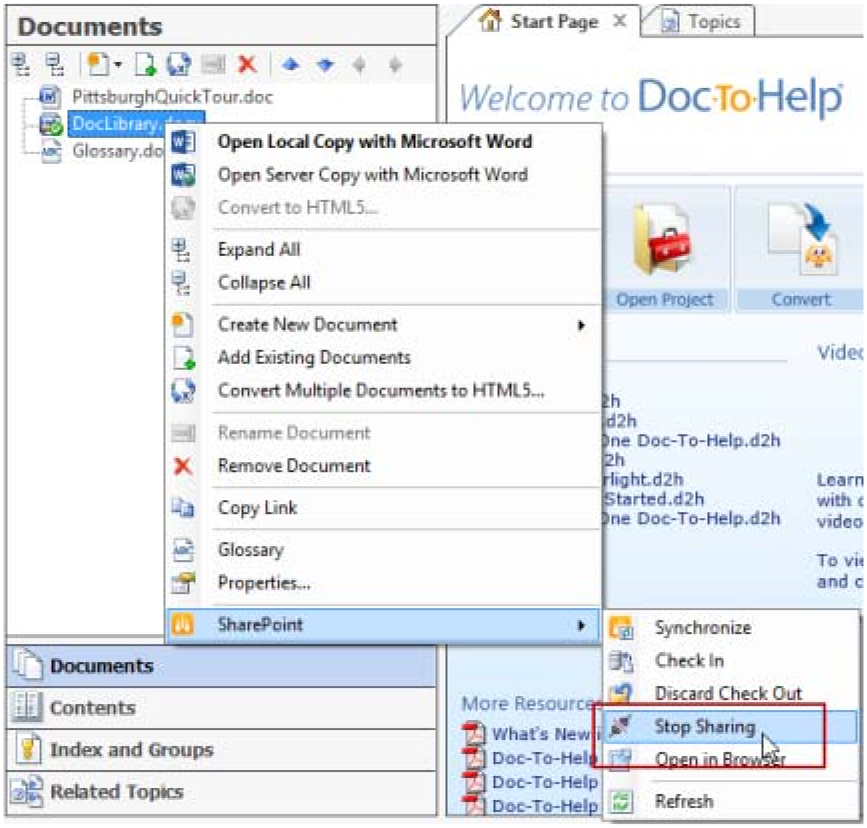
Working Offline



The **Work Offline** button on the **Home** tab turns off the synchronization with the SharePoint Library server, so there will be no synchronization checks on open/close or edit. Clicking the **Work Offline** button again will restore online mode. After you are back in online mode, click the **Synchronize** button to synchronize the local documents with the SharePoint Library.

Stop Sharing a Document

If you no longer wish to share a document with the SharePoint Library, you can stop sharing using the right-click menu. Select the document in the **Documents** pane, right-click and choose **SharePoint** > **Stop Sharing**. The document will no longer be shared with the SharePoint Library and its icon will change accordingly.



Uploading and Working with Documents in a SharePoint Translation Library

Uploading your content to a **SharePoint Translation Library** provides an easy way to track translations, provide content to translators who do not use Doc-To-Help, and create localized targets.

A Wizard will take you through the steps of uploading your documents to a SharePoint Translation Library. In addition to being uploaded, separate Doc-To-Help projects will be created for each translation language. The workflow for translation projects set in SharePoint will be used for your projects. The project translators do not need to have Doc-To-Help on their machines to translate, and as they work with the documents, status notification messages are posted in Doc-To-Help so that you know where each document is in the workflow.

SharePoint Password Note: If you change your SharePoint password after uploading documents, you will no longer be able to synchronize your documents. You can fix this by clearing your Windows credentials.
1. In Windows, open the **Control Panel** > **User Accounts and Family Safety** > **Credential Manager**. (This is the path for Windows 7.)
2. Find the SharePoint Server credentials and remove them.
3. Restart Doc-To-Help.
You will be prompted to enter your new password the next time you try to connect to SharePoint

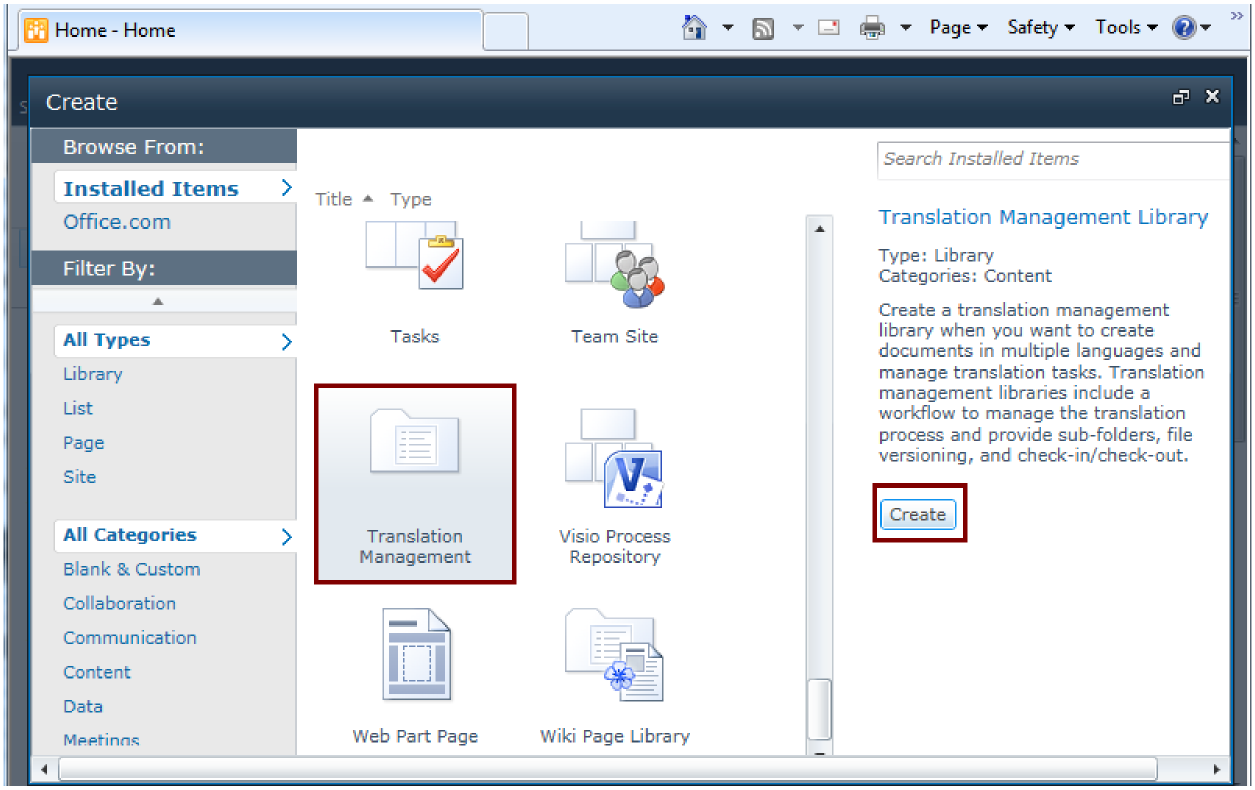
---

Creating a SharePoint Translation Library

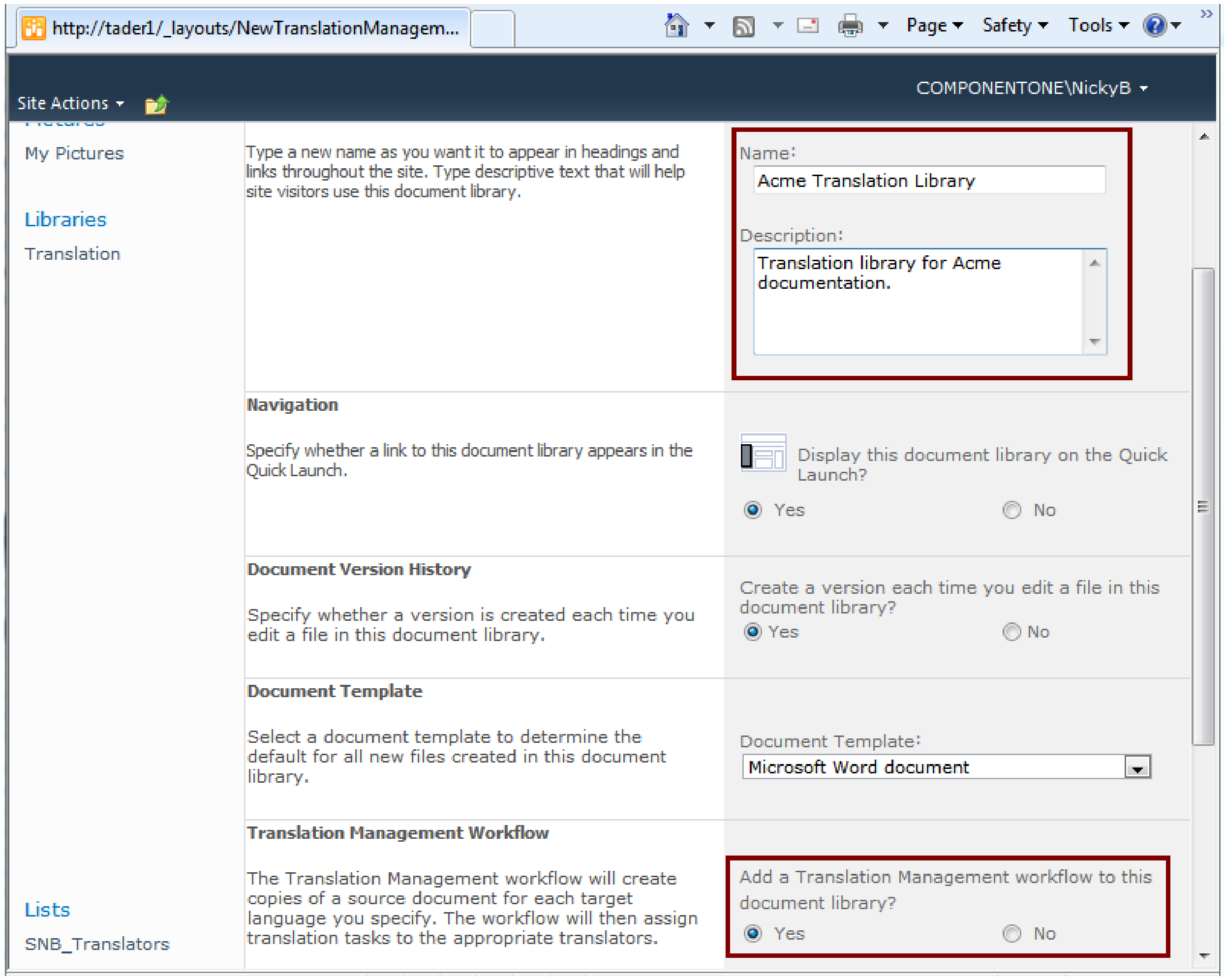
Before you upload projects from Doc-To-Help to SharePoint, you first need to create a **SharePoint Translation Library**.

To create SharePoint Translation Library

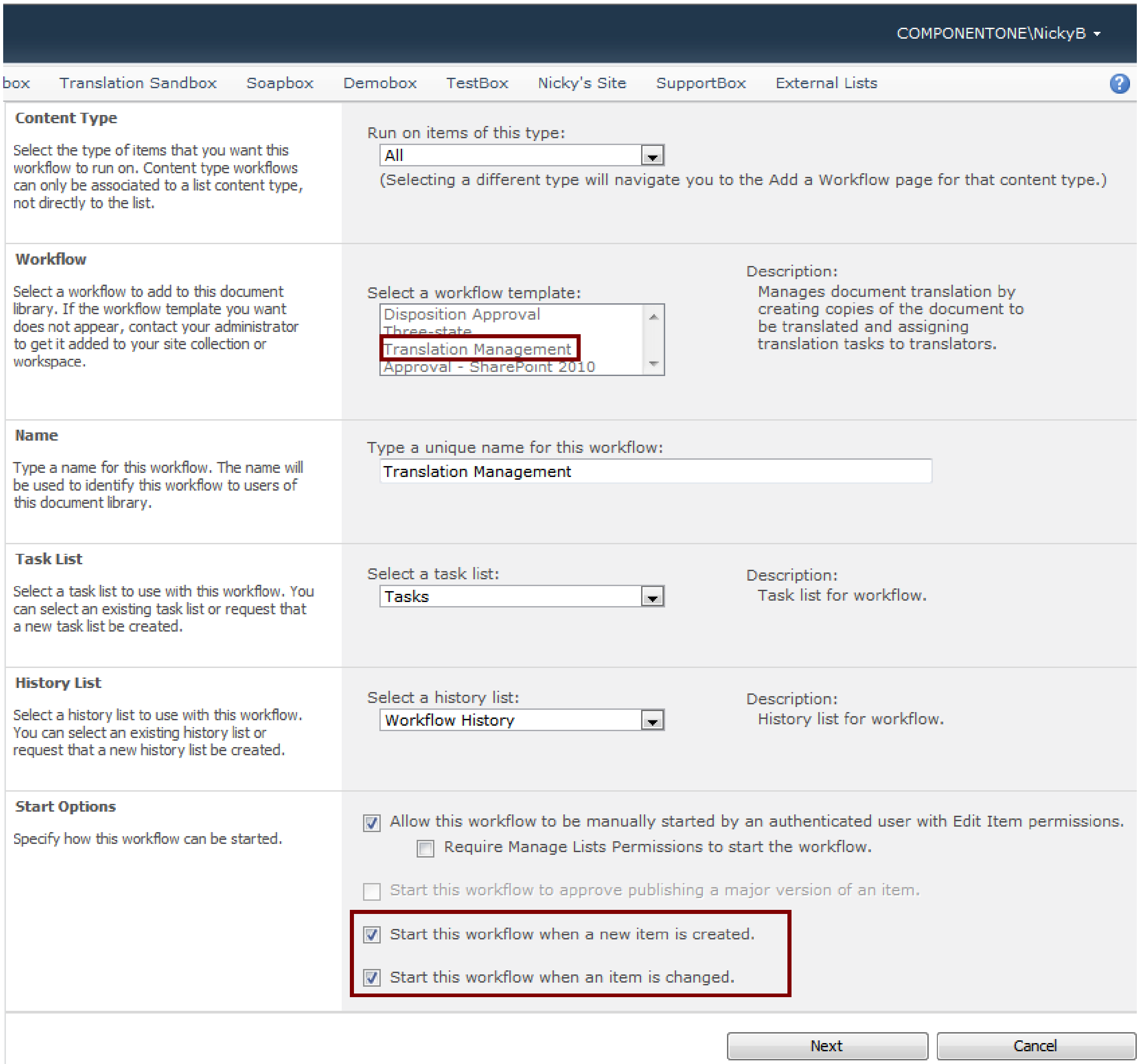
1. Open SharePoint and choose **Site Actions** > **More Options**.
2. Choose **Translation Management** and click the **Create** button.



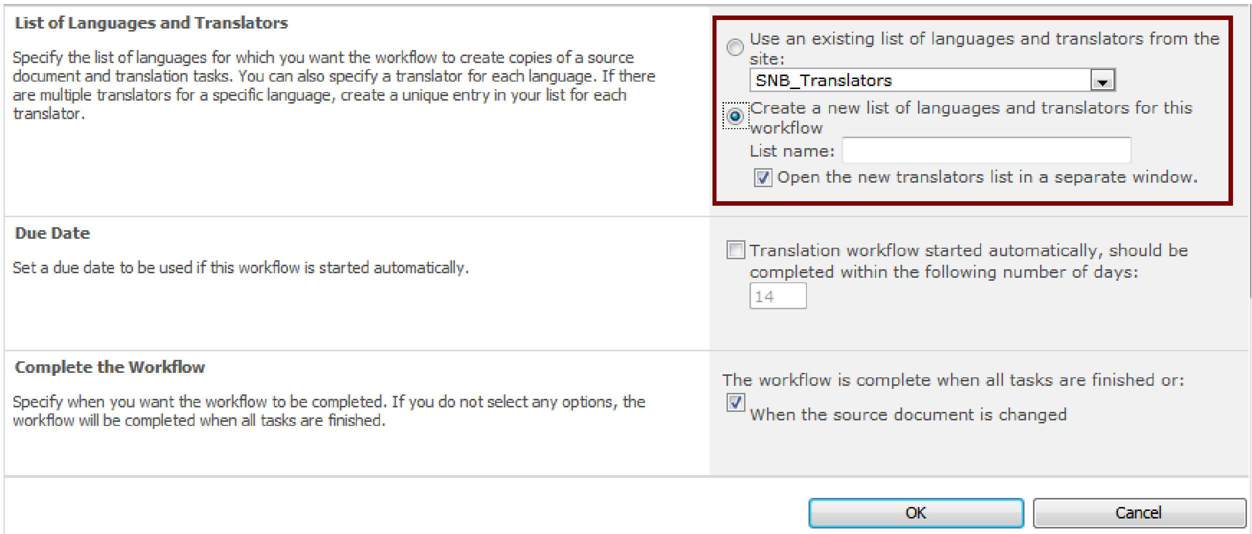
3. Name the new Library and give it a description. Note that a translation workflow will be added by default.



4. Click **Next** and select a **Translation Management Workflow**. You can also enable **Start options** that will start the workflow automatically when an item is changed or a new item is created. Please note that if the Translation Library has more than one workflow, Doc-To-Help will use the first one.



5. Click **Next** and choose a list of translators (or create a list) and click **OK**. Please note that when you upload your documents from Doc-To-Help to SharePoint, the languages available in the **Create Translation Project** wizard will be taken from translator's list associated with this workflow.



Now that your Translation Library has been created, you can upload your Doc-To-Help documents to this Library.

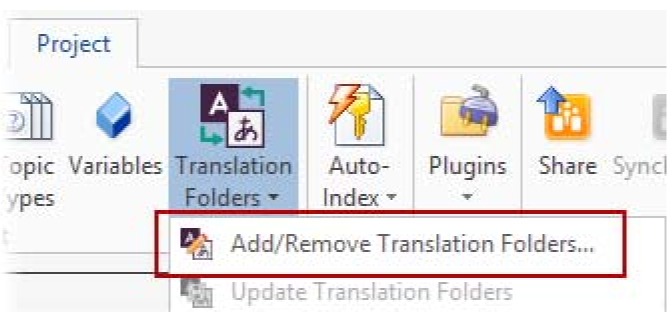
This blog post also describes this process: SharePoint's Translation Management Features

Uploading Documents to a SharePoint Translation Library

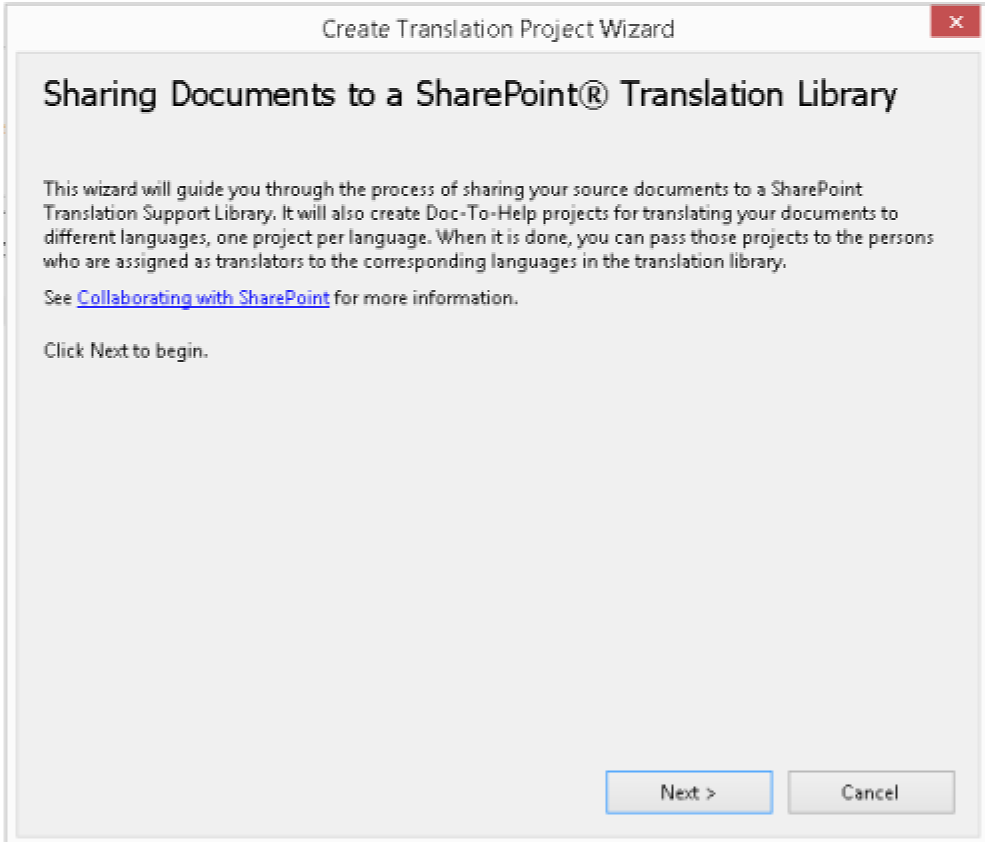
You must create a Translation Management Library in SharePoint before uploading your documents. See Creating a SharePoint Translation Management Library on page 360.

To upload documents to a SharePoint Translation Library

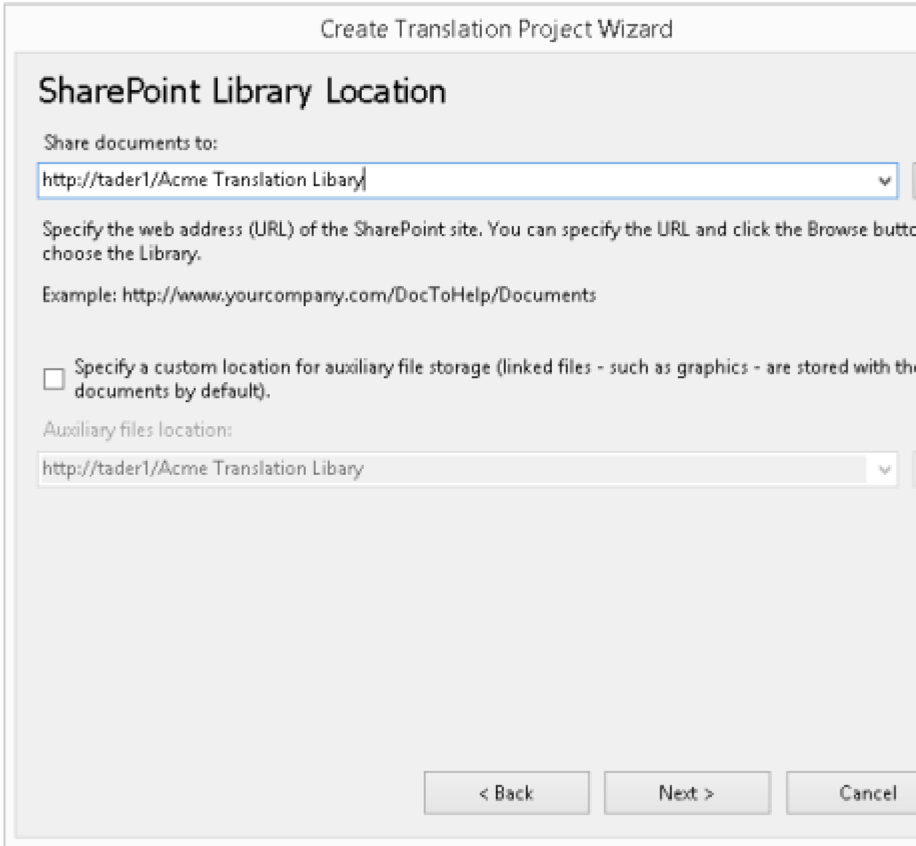
1. Open Doc-To-Help, then open the project you would like to upload to the **SharePoint Translation Library**.
2. Click the **Translate** button on the Doc-To-Help **Home** tab.



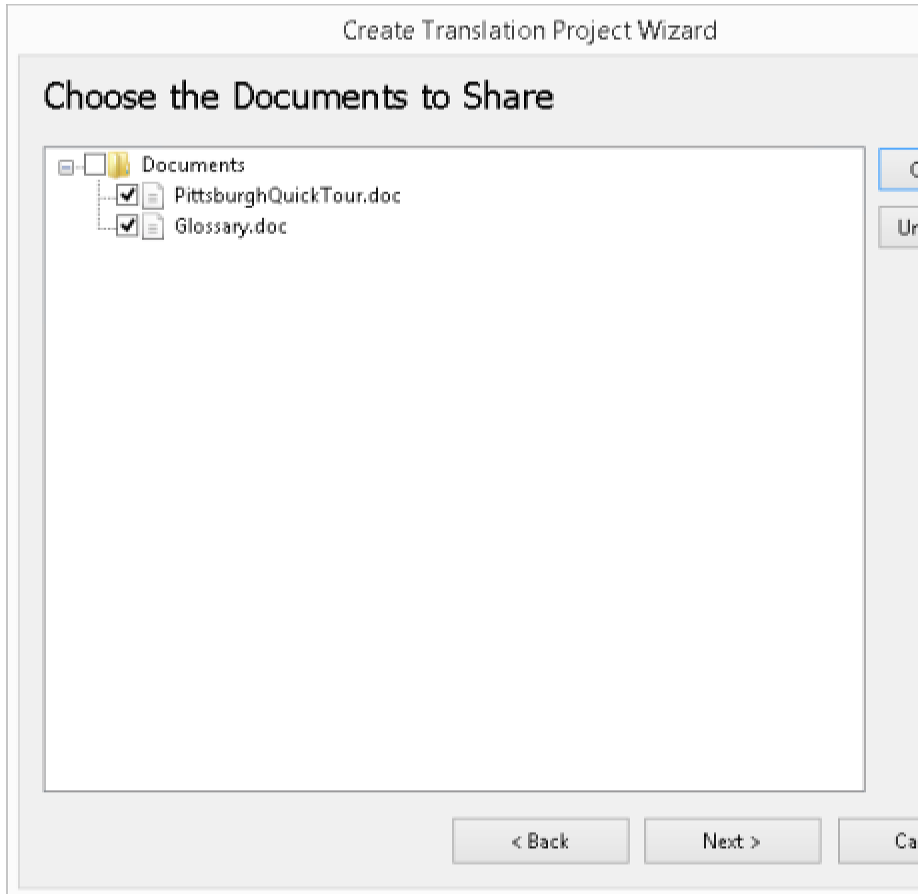
3. The **Create Translation Project Wizard** will open.



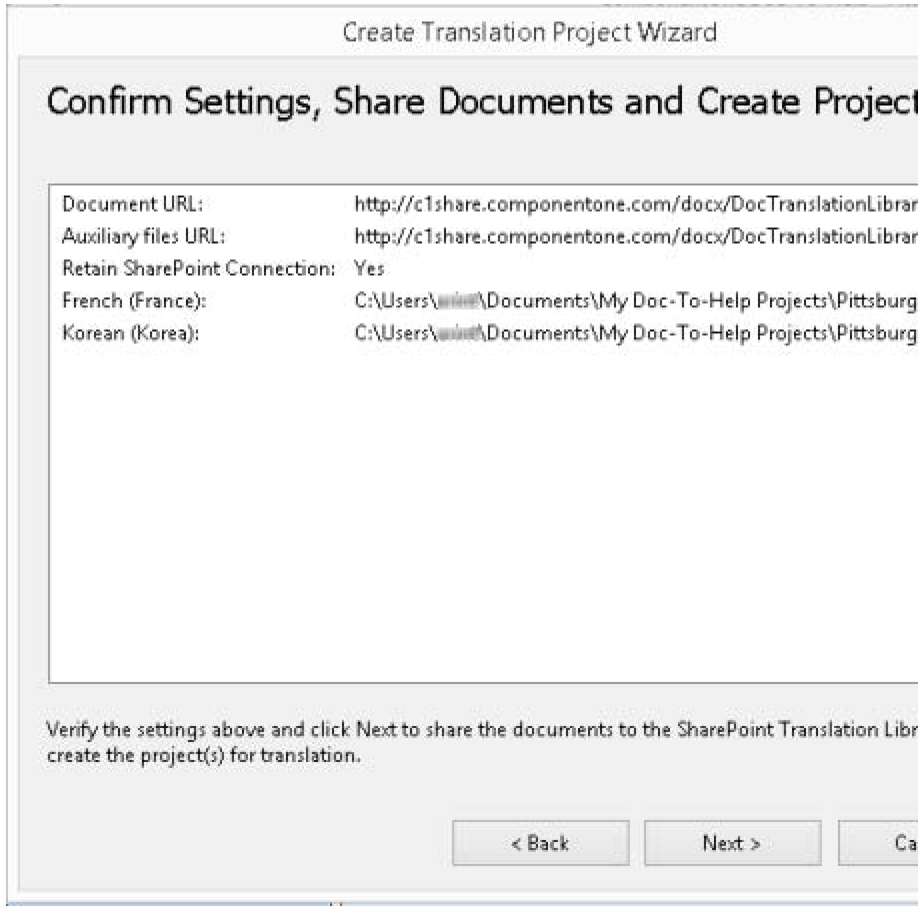
4. Click **Next**, then choose the SharePoint Translation Library.



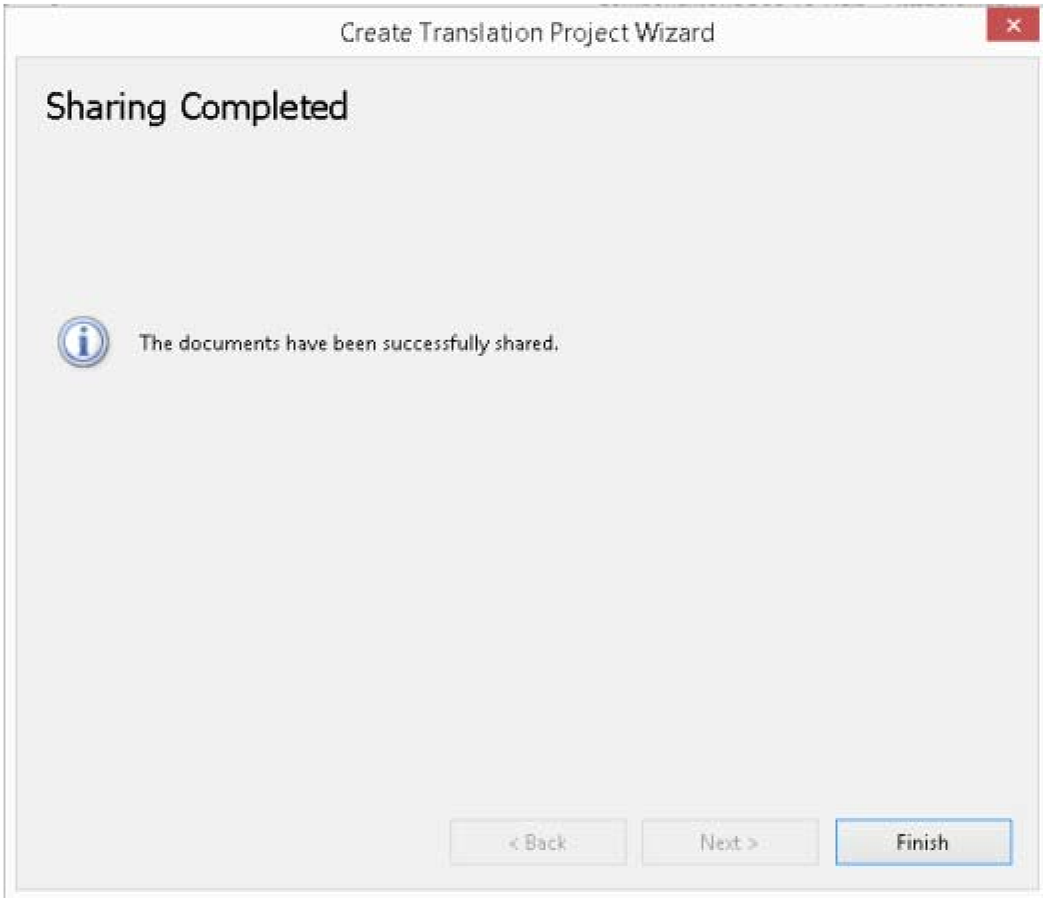
5. Confirm the documents to share and click **Next**. Note that the since the project was created with the default language of **English**, Doc-To-Help will automatically assign the documents to that language.



6. Click **Next**, confirm the settings and click **Next** again.

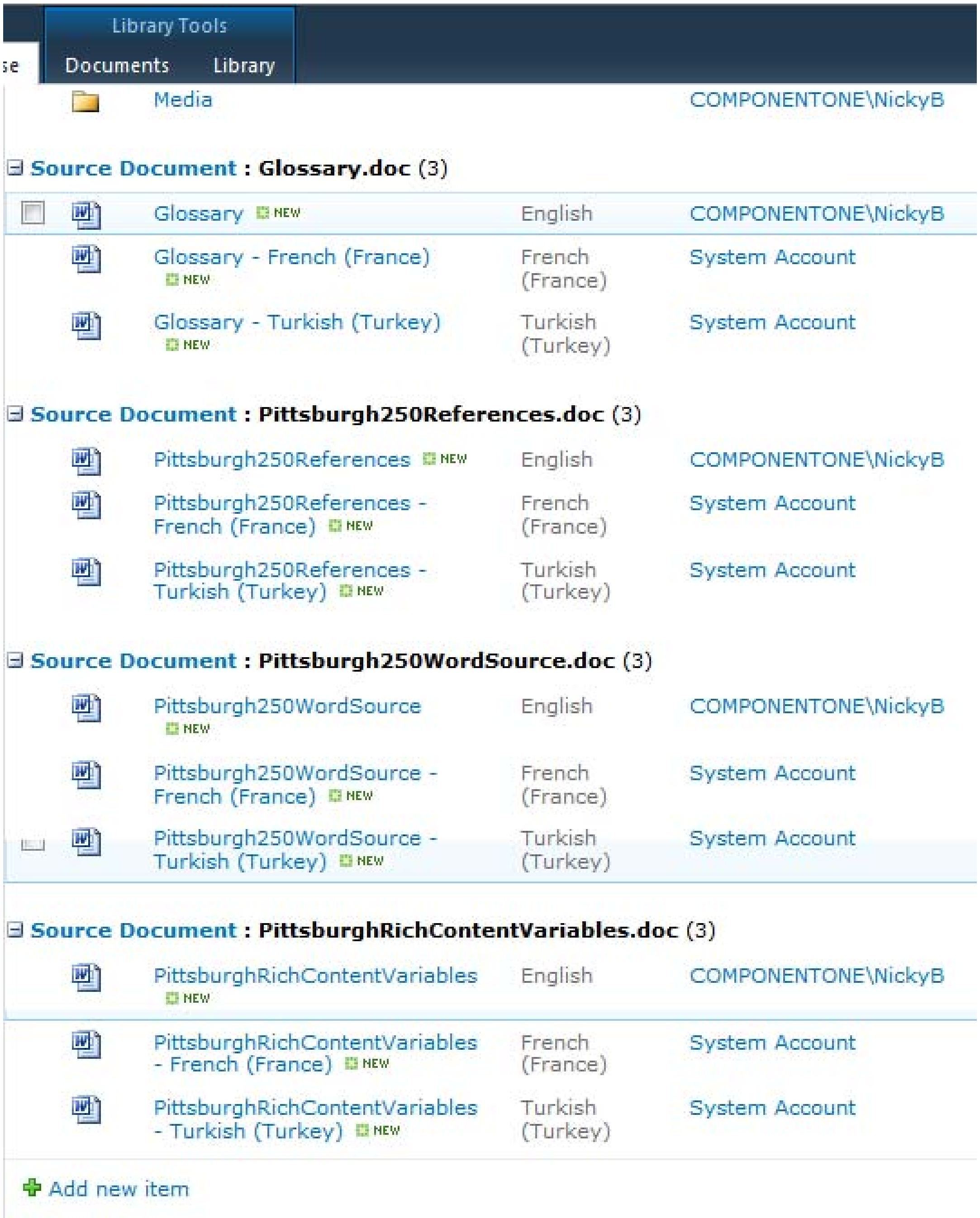


7. Doc-To-Help will then share the documents and create a project for each language represented in the list of translators chosen for the SharePoint Translation Library. Click **Finish**.

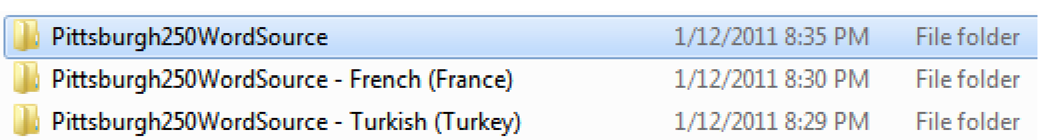


8. If the project has auxiliary files (such as graphics) Doc-To-Help will ask if you would like to synchronize those also. Click **OK**. After synchronization, click **Close**.

When you open the SharePoint Translation Library, there will now be a version of each document for each language. (The languages for this example were Turkish and French.)



In addition, a Doc-To-Help project was created for each language.



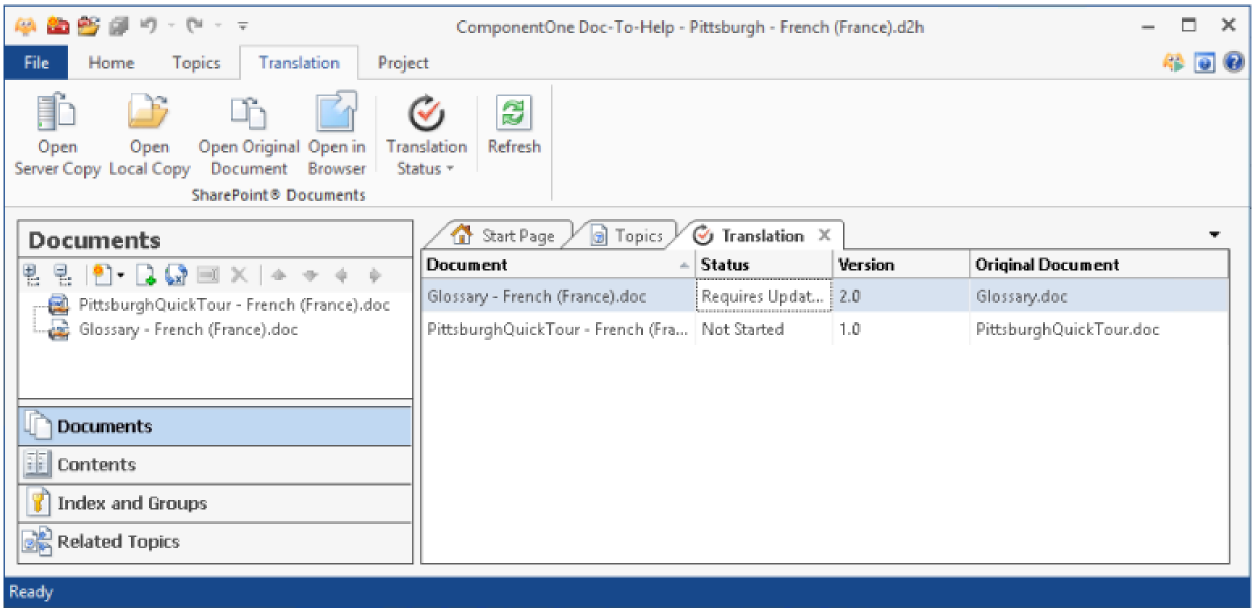
This blog post contains additional information about this process: Managing Translation with Microsoft SharePoint and Doc-To-Help.

For tips on setting up a Doc-To-Help project for translation, see this blog post: Setting Up Doc-To-Help for Languages Other Than English.

Working with a Translation Project

After you have uploaded your project to a SharePoint Translation Library, translators can check the documents out of SharePoint and begin translating them. (SharePoint 2010 only; Translation Libraries are not included in SharePoint 2013.)

When the Doc-To-Help project is opened, there will be a new Translation ribbon and window. You can use them to track the status of the documents as they go through the workflow and are translated, reviewed, and approved.



As the documents travel through the workflow, their status will be displayed under the Translation tab on page 120.

To open a document

1. Choose a document in the **Documents** pane or the **Translation** window.
2. Click the **Open Server Copy** or **Open Local Copy** button on the **Translation** tab.

   The document will open in Microsoft Word. To open the original version of the document chosen, click the **Open Original Document** button.

To open the SharePoint Translation Library

Click the **Open in Browser** button on the **Translation** tab.

To set the translation status of a document

1. Choose a document in the **Documents** pane or the **Translation** window.
2. Click the **Translation Status** button on the **Translation** tab. Options are **Not Started**, **In Progress**, and **Completed**.

To refresh the SharePoint Translation Library

Click the **Refresh** button on the **Translation** tab. Updated statuses will display under the **Translations** tab.

Notes and Best Practices:

- In this process, your project documents are living ones. If you check out documents and edit them (see Checking Documents In and Out of the Sharepoint Library on page 357) then the translators will need to compare their documents to the edited ones and adjust accordingly. See Merging Documents on page 358 to learn how to compare documents.
- You can synchronize your documents with the SharePoint server, and work offline. See Synchronizing Documents on page 357, and Working Offline on page 359.
- If you need to add a new document to any existing translation project, it must be added to the appropriate SharePoint Translation Library first. Then click the **Add New Documents** button in the **Documents** pane and choose **SharePoint** for document location. See Adding SharePoint Library Documents to a Doc-To-Help Project on page 355.
- When document translation is complete for the project, you can build it in Doc-To-Help. If you wish to localize the Help Theme, you can translate it, see Customizing Themes on page 199). You can also change other project settings, such as the name of the project and other text strings in the **Help Target**s dialog box. See Creating Help Targets on page 133 for more information.

Publishing to SharePoint

Doc-To-Help provides two ways you can publish your Doc-To-Help output to SharePoint.

One option is to publish your NetHelp to SharePoint, where anyone with permissions can access it. No need for a special setup to post your Help to the web. You can link directly to the Target in SharePoint, as well as display the Target in a SharePoint web part. All the NetHelp Target files are published to SharePoint and accessible from **index.html** (the default name for the NetHelp home page).

The other option is to publish your NetHelp output to a SharePoint wiki, where anyone with permissions can access and edit it. Publishing to a SharePoint wiki is another way to collaborate with your team – even your entire user community – if you wish. So this alternative is both a publishing and collaboration option.

Both options are quick and easy using the **Publish to SharePoint** Wizard.

The two options available from the Wizard are:

- **Upload to a SharePoint Document Library**. All NetHelp Target files will be uploaded to a SharePoint Document Library. The files will not be altered when uploaded, but once on SharePoint, they should not be edited. If changes need to be made to the content, make them in the Doc-To-Help project, build NetHelp, and upload to SharePoint. As with any NetHelp project, the default home page for the project is index.html.
- **Upload to a SharePoint Wiki Page Library**. The NetHelp Target will be converted to wiki format and uploaded to a SharePoint Wiki Page Library. Some formatting and functions (such as popups) will be lost because wikis do not support them. Once the NetHelp is uploaded to a Wiki Page Library, those with proper permissions may edit the content.

Note: Doc-To-Help also includes the option to share your Doc-To-Help source documents and graphics to a SharePoint Document Library or Translation Management Library. These options are great for content collaboration, source control, versioning, and more. See Collaborating with SharePoint on page 353.

Publishing to a SharePoint Document Library is compatible with Microsoft® SharePoint® 2007, 2010, and 2013, as well as Office 365 (SharePoint Online); publishing to a SharePoint Wiki Library is compatible with SharePoint 2010 and 2013, and Office 365.

Before publishing NetHelp to a SharePoint Library, check the following SharePoint Library settings:

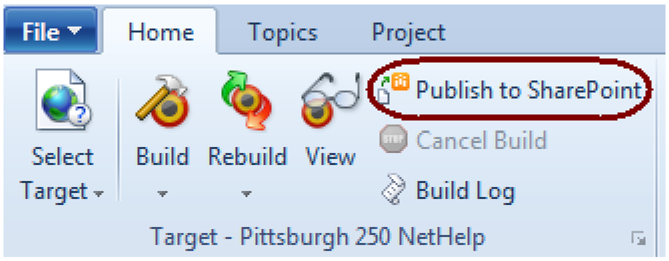
- Web Application General Settings should be set to **Permissive** (the default is **Strict**)
- In Versioning Settings, **Require Check Out** should be set to **No**.

SharePoint Password Note: If you change your SharePoint password after publishing outputs, you may receive an error message the next time you attempt to publish. You can fix this by clearing your Windows credentials.

1. In Windows, open the **Control Panel** > **User Accounts and Family Safety** > **Credential Manager**. (This is the path for Windows 7.)
2. Find the SharePoint Server credentials and remove them.
3. Restart Doc-To-Help.
You will be prompted to enter your new password the next time you try to connect to SharePoint

Publishing documents to a SharePoint Document Library or Wiki Page Library

1. Open the Doc-To-Help project. Select a NetHelp Target using the **Select Target** button and click the **Build** button.
2. Click the **Publish to SharePoint** button in the Doc-To-Help **Home** tab (**Targets** ribbon group).

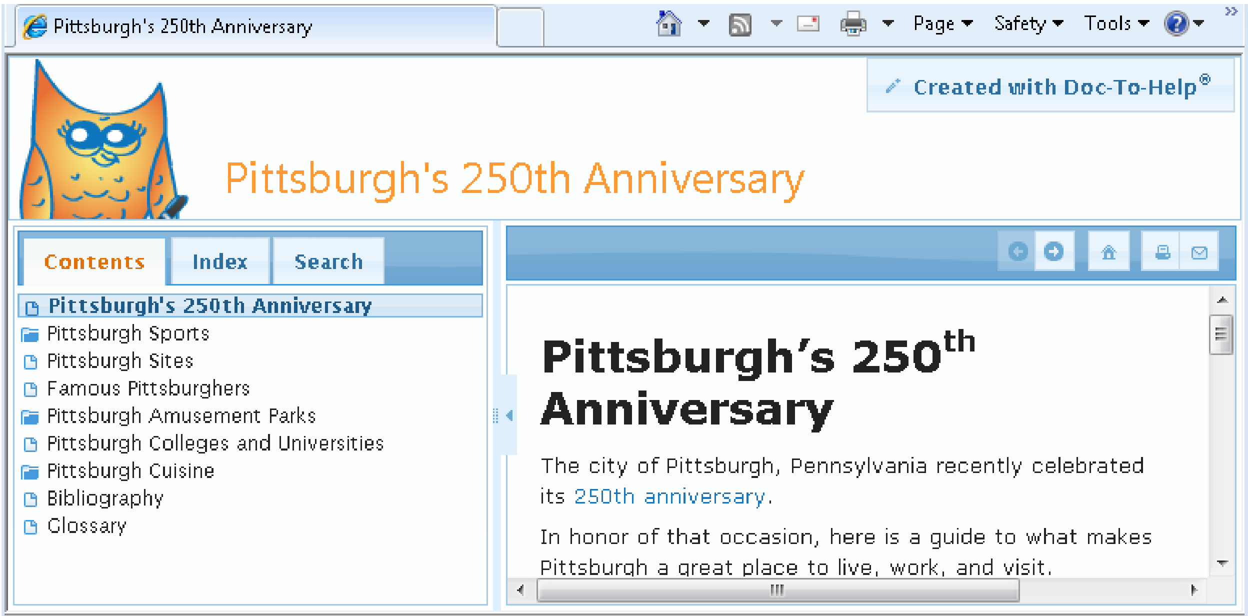
   The **Publish to SharePoint** Wizard will open. Click **Next**.
3. Choose either **Upload to a SharePoint Document Library** or **Upload to a SharePoint Wiki Library**.
4. Choose the SharePoint Library you want to upload to. Enter the SharePoint site URL in the **Upload Files to** field. You can choose a specific folder, or create a new folder for the documents by clicking the ellipsis button and using the **Select Server Folder** dialog box.

   If the folder chosen is not empty (uploading to a new Wiki Page Library will always display this message because they contain default wiki intro information), the Wizard will warn you. Click **Yes** to continue.
5. Confirm your choices and click **Next**. The documents will then be uploaded.
6. Click **Finish** to close the Wizard. After the Wizard has done its work, it will display a **View** button.

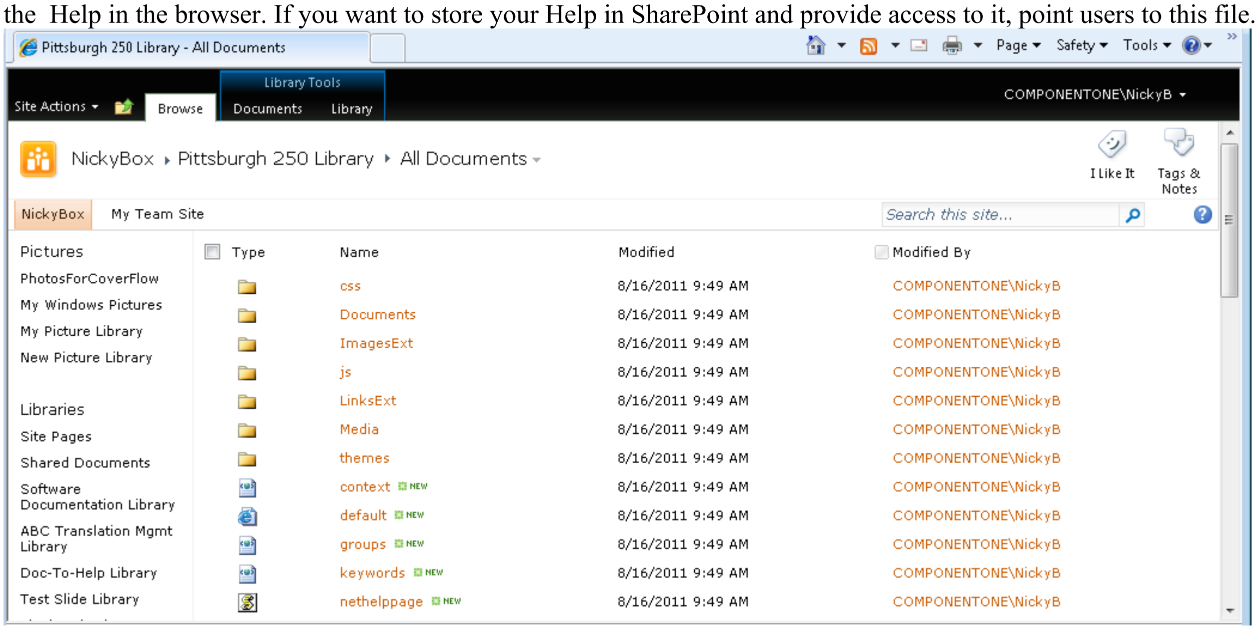
   If you uploaded to a **Document Library**, your NetHelp will display in a browser. The NetHelp output files are stored in SharePoint, and can be viewed by opening the Document Library.

   If you uploaded to a **Wiki Page Library**, SharePoint will open and the NetHelp Table of Contents will be displayed. Click on the TOC links to navigate the project.

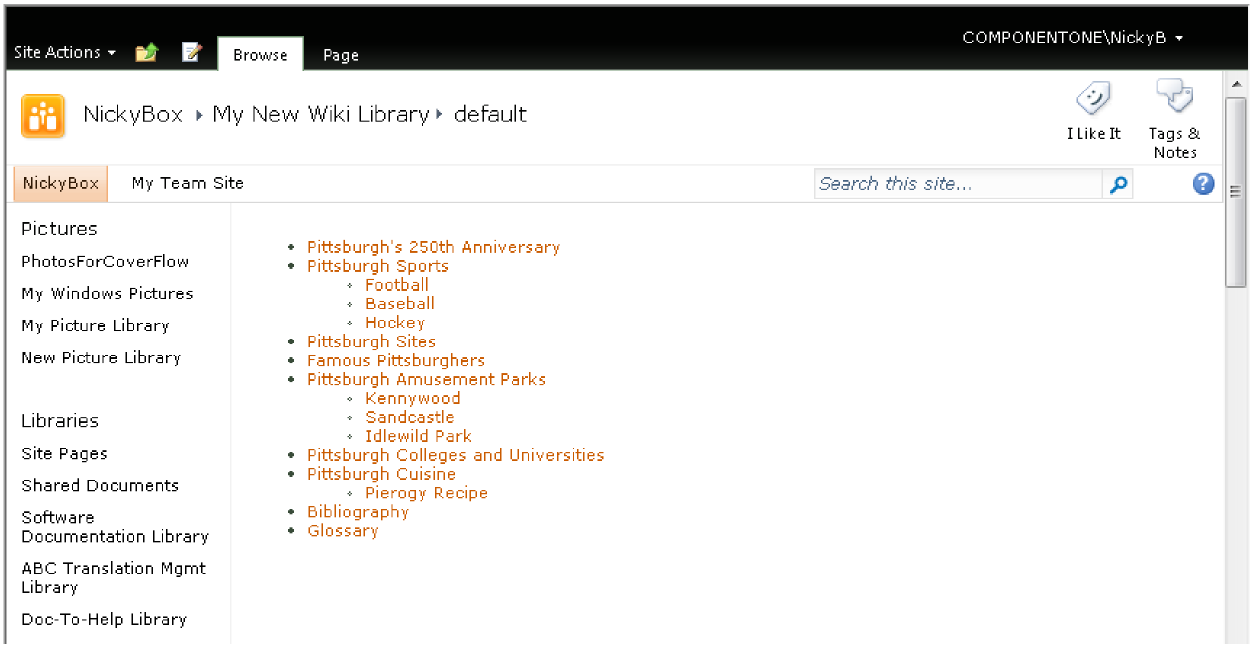
When you click the **View** button from the Wizard, the NetHelp Target uploaded to a SharePoint Document Library will open in the browser ...



... but the Target files/folders are in SharePoint. The **index.html** file (the default Home Page for NetHelp projects) opens the Help in the browser. If you want to store your Help in SharePoint and provide access to it, point users to this file.



This is a NetHelp Target uploaded to a SharePoint Wiki Library. See Editing a SharePoint Wiki on page 376 for instructions on editing a wiki.

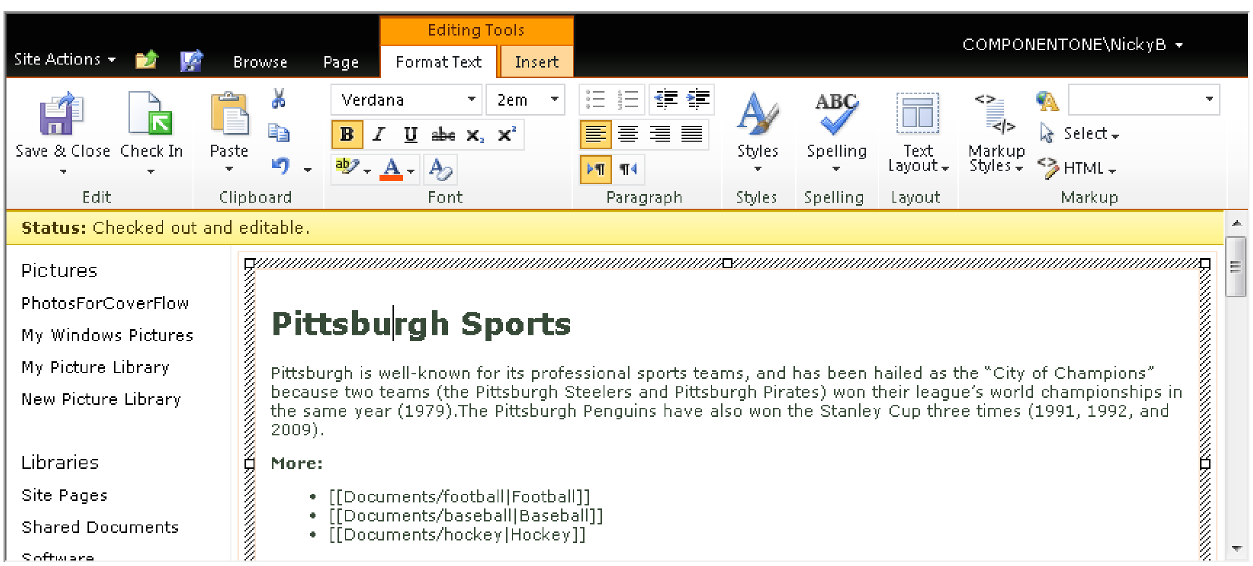


Editing a SharePoint Wiki

Once your NetHelp is in a SharePoint wiki, it can be edited by anyone with the proper permissions.

To edit a SharePoint Wiki:

1. Click the **Page** tab. Click the **Check Out** button and check the page out.
2. Click the **Edit** button. The page will switch to edit mode and editing can begin.



The **Format Text** and **Insert** ribbons displays a complete set of Word-style formatting options, as well as spell check. You can edit anything, add images, and insert links to URLs.

Note that in Edit Mode links to other topics within the project become wiki links ([[Documents/football|Football]]). This syntax is unique to wikis and makes it easy for anyone to edit existing links and create new ones. See this topic on MSDN for more information.

3. After making your edits, save and check in your changes by clicking the **Check In** button on the **Format Text** tab.

Working on a Team

Doc-To-Help Team Authoring makes it possible for multiple Help authors to work together on a single project. Projects can be made available to the entire team, and one author's changes will not be overwritten by another author's changes. Your projects can be stored on a Microsoft® Team Foundation Server, in a Web Repository, or in a File System Repository. If you would prefer to store your documents in Microsoft® SharePoint®, please see Collaborating with SharePoint on page 353.

Team authoring is a 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, the administrator* starts with a regular Doc-To-Help project and makes it available to other team members by uploading it to the central repository. Once the project is uploaded, it becomes the team project. Team members can then connect to and create working copies on their own machines.

Each team member works on their 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 the team member's working copy of the project. Likewise, changes made by other authors cannot be viewed until they are retrieved from the repository. Doc-To-Help makes it easy to check files in and out of the repository.

Team Authoring Requirements

A Doc-To-Help project must meet the following team authoring requirements:

- All of the project's documents and auxiliary files must be located within subfolders of the project folder (Media, CSS files, Documents, etc.). Doc-To-Help automatically sets your project up this way by default.
- If your project uses customized templates, style sheets or themes, all team members must have the same versions of those files in the same locations on each of their machines.
- All team members must have the same version (including build number) of Doc-To-Help installed on their machine. This information is available from the **File** tab > **Help** > **About Doc-To-Help**.
- If using a Team Foundation Server repository, all team members must install Microsoft Visual Studio Team Explorer 2008 or 2010; 2008 is available at: http://www.microsoft.com/downloads/details.aspx?familyid=0ed12659-3d41-4420-bbb0-a46e51bfca86&displaylang=en

Team Authoring options can be set in the **Options** dialog box (**File** tab > **Tools** > **Options** > **Team button**).

*N**ote**: Each team authoring project must have an administrator. The administrator uploads the project to the repository and configures the settings. In addition, the administrator handles other functions, such as: removing backup files, unlocking files from the repository, upgrading the team project, and disconnecting team authoring support from working copies of projects. See Team Authoring Administrative Functions on page 391.

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. This should be done by the project administrator.

Note: Before setting up a Team Project, make sure your project satisfies all requirements described in Team Authoring Requirements on page 381.

Doc-To-Help supports three repository types:

- **Microsoft® Team Foundation Server** — Files are stored in your company's Team Foundation Server repository.
- **File System Repository** — Files are stored at a location accessible via your company's file system, usually on a computer belonging to your organization's network. The only prerequisite for sharing a project in a file system repository is that all team members must have access to the shared folder on the network.
- **Web Repository** — Files are stored at a location on the Internet, on a Web server. They can be accessed from anywhere on the Internet, including from behind a firewall. Sharing a project with a Web repository allows you to share your team project with anyone connected to the Web. 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.

To setup IIS for WebDAV in Windows 7, 8, Vista, and Windows Server 2008:

1. Open IIS Manager and select **Default Web Site** (or the web site you want to setup) in the pane on the left.
2. In the center pane double-click on **WebDAV Authoring Rules**.
3. In the Actions pane (on the right) click on **WebDAV Settings...**
4. In the Property grid find the **Property Store** and click the **Browse** button to change this setting.
5. If there are no items in the **Namespace Collection Editor** (in the Members pane), click the **Add** button which will add a default namespace for storing WebDAV properties.
6. Return to the **WebDAV Settings**. In the **Property Behavior** section, set **Allow Property Queries with Infinite Depth** to **True.**

Sharing a Project to a Repository

To begin team project set up, the project administrator must share the project to the repository. This copy will become the master project (or the **team project**). After the team project is created, team members can connect to and download the team project to create their own working copies.

To share a project for team authoring

1. Create a new Doc-To-Help project or open an existing one.
2. Select the **File** tab > **Team Authoring> Share Project for Team Authoring**. The **Share Project** wizard will open.
3. Click **Next** to continue.
4. Under **Select Repository Type**, choose one of the following:

- **Team Foundation Server**
- **File System Repository**
- **Web Repository** (if you are need to set up an IIS virtual directory see Setting up a virtual directory on your Windows machine below)

5. Click **Next** to continue.
    - For **Team Foundation Server**, enter the **Team Project Location** URL. You can click the ellipsis button to browse to a different server.
    - For **File System Repository**, enter a network location or click the ellipsis button to browse to a folder on the network.
    - For **Web Repository**, 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 you are using Windows Authentication, leave **Use Windows Authentication** selected. If you are using another form of authentication, clear the check box and enter your login credentials.
6. For a **File System** or **Web Repository**, click the **Repository Test** button to test the connection to the repository and the files and subfolders. Click **Close**.
7. Click **Next** to continue.
8. In the **Repository Options** window, set the backup options:
    - **Limit the number of backups for each file** — Used to set the limit on the number of backups kept for each file in the repository. The default is 5, but you can change it in the **Number of backups** kept box. If you do not set a limit, every time a file is checked-in, a new backup file will be 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 check box to permit other team members to delete old backup files when they attempt to check in a newer version of a file and the backup limit 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 a team member deletes old backup files. This option is only available when the **Allow users to delete old backups in check-in** check box has been selected.

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 394 for more information.

9. Click **Next** to continue.
10. Click **Yes** to confirm your repository information and click **Next** to create the team project. The **Upload project** dialog box appears.
11. 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 you will use to perform all authoring tasks. The **team project** in the repository 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 check in the changes to the central repository.

12. Click **Finish**.

Setting up a virtual directory on your Windows machine if using IIS (Web Repositories only)

If you are using Internet Information Services (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.

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.

Connecting to a Team Project

Team members must connect to and download the team project in order to create their own working copies of the project. Once a team member has a working copy, they can check files out, edit them, and check them back in.

The project administrator (the person who originally shared the project) should provide team members with the network path or the URL (depending on the type of repository) of the team project.

Note: Once you have created a working copy of your project following the steps below, you do not need to do it again, just open the working copy of the project and begin working. See Working in a Team Project on page 387 for details.

To connect to and download a team project

1. Open Doc-To-Help.
2. Select the **File** tab > **Team Authoring** > **Connect to Team Project**.
3. Under **Select repository type**, choose one of the following:
   - **Team Foundation Server**
   - **File System Repository**
   - **Web Repository**
4. Click **Next** to continue.
   - For **Team Foundation Server**, enter the **Team Project Location** URL. You can click the ellipsis button to browse to a different server.
   - For **File System Repository**, enter a network location or click the ellipsis button to browse to a folder on the network.
   - For **Web Repository**, enter the **URL** of the virtual directory. (If necessary, enter your login credentials.)
5. Click **Next** and specify the working project folder. This will be where the project will be stored on your machine and is your working copy of the project.

6. Click **Next** and confirm the working copy information.
7. Click **Next** to create the working copy.
8. Click **Close** and then click **Finish**. Your working copy of the project will open.

The Team Authoring Environment

When you open the working copy of a team project, Doc-To-Help will include an additional tab and window, both named **Team Authoring**.

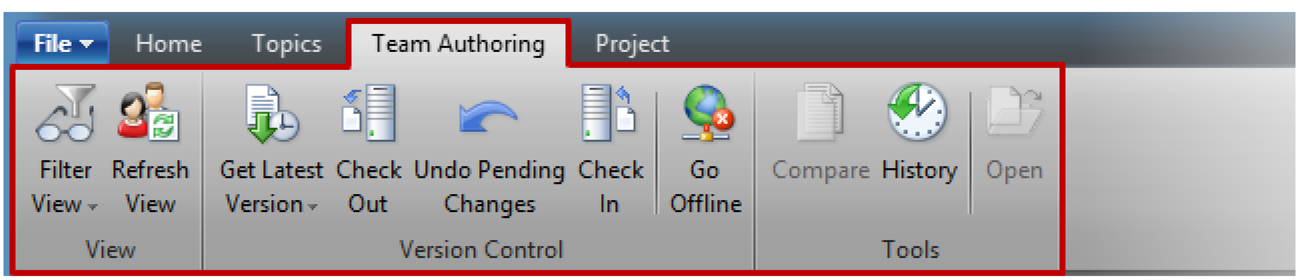
Three different file types make up a team authoring project:

- **Documents:** These are the source documents included in your Doc-To-Help project. By default, they will be located in the **Documents** folder. Any Rich Text Variable documents will be included in these folders also.
- **Auxiliary files:** These are any files that are not documents. Examples include graphics and css files. By default, these will be located in the **Media** and **CSSFiles** folders.
- **Project file (.d2h extension):** This is the project itself, which contains all project information such as keywords, attributes, windows, project properties, etc.

You can check out and work with all three file types. In projects stored in a File or Web Repository, when 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. When 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. Projects stored in Team Foundation Server (TFS) repositories have additional check out options.

Team Authoring ribbon tab

After you have opened your project, you can do the following with the **Team Authoring tab**:

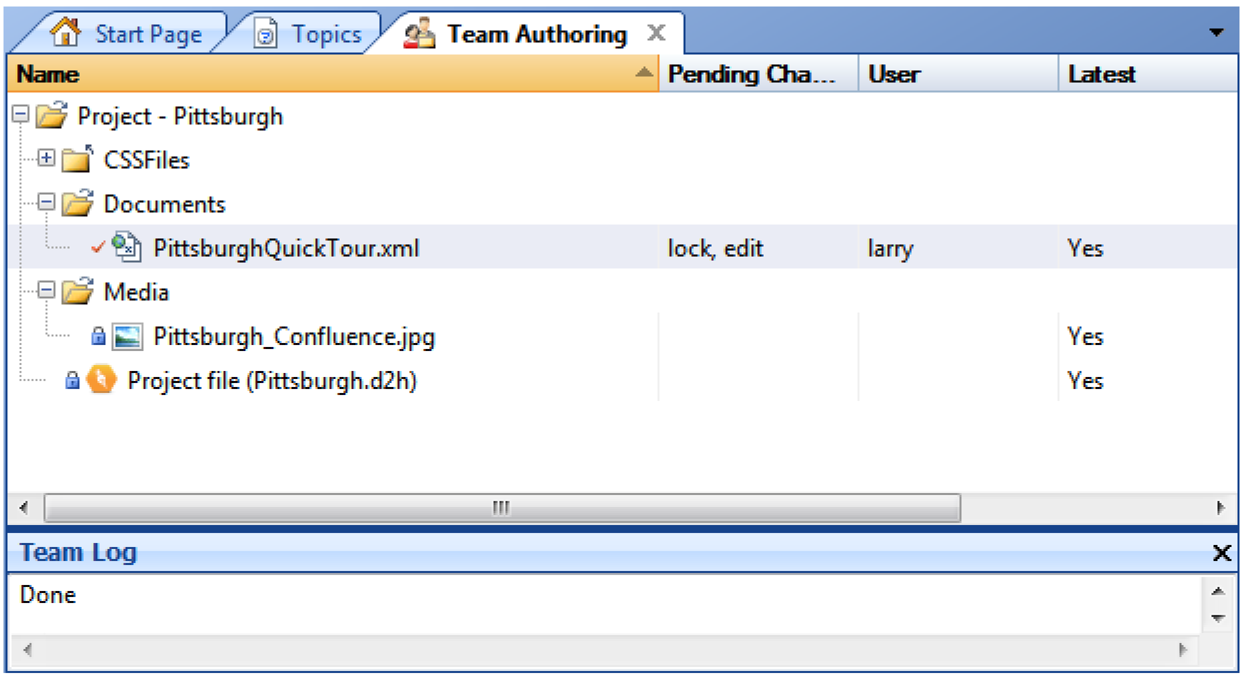


- **Filter View** — Choose the file types that will be displayed in the **Team Authoring window** (options are: **Documents**, **Auxiliary Files**, **Only Pending Changes**).
- **Refresh View** — Refresh the entire display to update the file status.
- **Get Latest Version** — Get the latest version of the selected document from the repository. (TFS projects include a drop-down that allow you to choose a specific version of a document.)
- **Check Out** — Check the selected document out of the repository.
- **Undo Pending Changes** — Undo all changes made to a document that has been checked out.
- **Check In** — Check the selected document into the repository.

- **Go Offline** — Break the connection with the server. You will be prompted to choose the project editing mode: **Restricted** or **Unrestricted**. If you choose **Restricted**, an **Unrestricted Project Editing** button will appear in the ribbon, allowing you to switch modes if you wish.
- **Compare** (TFS projects only) — Compare the selected document with the version on the server.
- **History** (TFS projects only) — View a history of the selected document's changes.
- **Open** — Open the file selected in the Team Authoring window for editing.

Team Authoring window

The **Team Authoring window** opens when you click the **Team Authoring tab**. It displays the file details. Many options can be performed via the right-click menu.



A checkmark next to file name indicates that it has been checked out. A lock icon indicates that the file is part of the team project.

To change the column display, right-click in the window and choose **Columns** from the menu. The options are:

- **Pending Change** — Displays file status. TFS projects have different status displays than File or Web Repository projects. File or Web Repositories will note the check out status and the user name, if a document has been modified and who modified it, if it is a new document, and if it is unmodified by anyone. TFS projects will display the lock and edit status. The user name will display in the **User** column.
- **User** — The name of the user working with the file.
- **Latest** — Indicates whether the file displayed is the latest version.
- **Last Check-in** — Date/time the file was last checked in.

The **Team Log** information displays for TFS projects only.

Working in a Team Project

When you work in a team project, you can edit documents, add files to the project, delete files and more. Options for working with files are available from the **Team tab**, as well as by right-clicking on a file in the **Team Authoring window**.

To open the working copy of a team project

- After you have connected to the team project (see Connecting to a Team Project on page 384) the project will automatically open.
- To reopen it, choose the **File** tab > **Open Project** and open it just as you would any other project.

Editing a Document

To edit a document, you first need to check it out. After it is checked out, no one else can edit it until it is checked back in. (If using TFS, you can allow checking out by others, depending upon the **Lock type**.)

To check out a document or file

You should click the **Refresh View** button before checking out a file to update the **Team Authoring** window.

1. Select the file in the **Team Authoring** window.
2. Click the **Check Out** button. The **Check Out** dialog box will open.

   For File and Web Repositories, after the file is checked out, click the Close button.

   For TFS, you need to confirm the **Lock type** before you check out the file. Choose **Unchanged**, **Check out** or **Check in** and click the **Check Out** button. **Unchanged** will keep the existing lock, **Check Out** will prevent other users from checking the file in or out, **Check in** allows other users to check out the file but they will not be permitted to check it back in. To set the default **Lock type**, use the Doc-To-Help **Options** dialog box (**File** tab > **Tools** > **Options** >**Team button**).
3. Click the **Open** button to open the file.
4. Make your edits and save them. Close the file.
5. Click the **Check In** button. The **Check In** dialog box will open. Confirm the check in and click the **Check In** or **OK** button.

If you decide that you DO NOT want to check in your edited document, click the **Undo Pending Changes** button to undo all changes to the document.

Adding a New File to the Team Project

To add a new document or file to the team project, you must first add it to your working copy, then check the new file into the team project. After you check the file in, other team members can check out and edit the new file.

To add a new file

1. Add the document or file to your project as you normally would. At this point it only exists in your working copy.
2. Save and close the file.
3. Select the file in the **Team Authoring** window.
4. Click the **Check In** button. The **Check In** dialog box will open. Confirm the check in and click the **Check In** or **OK** button.

When you add a file to a project, you are making a change to the project (.d2h) file itself, therefore a message box will display that informs you that the project file will be edited. Click **OK** to confirm. For TFS projects, make sure to check the project file back in after you are done. For more on project files, see Checking Out the Project File on page 388.

New files that have not yet been checked into the repository will display in a different color to alert you. If you would like to change the text and highlight color, use the Doc-To-Help **Options** dialog box (**File** tab **>** **Tools** **>** **Options** **>Team button**).

Deleting a File from the Team Project

To delete an existing file from the team project, you must first check out the project. Once you have deleted the file from the team project, it will not be available for editing by any member of the team.

If the file you are planning to delete was never checked in to the repository, it was never part of the team project and can be deleted from your working copy without checking out the project file.

To delete an existing file

1. Select the project (.d2h) file in the **Team Authoring** window.
2. Click the **Check Out** button. The **Check Out** dialog box will open.

   For File and Web Repositories, after the file is checked out, click the **Close** button.

   For TFS Repositories, you need to confirm the **Lock type** before you check out the file. Choose **Unchanged**, **Check out** or **Check in** and click the **Check Out** button. **Unchanged** will keep the existing lock, **Check Out** will prevent other users from checking the file in or out, **Check in** allows other users to check out the file but they will not be permitted to check it back in. To set the default **Lock type**, use the Doc-To-Help **Options** dialog box (**File** tab **>** **Tools** **>** **Options** **>Team button**).
3. Select the file you would like to delete in the **Documents pane**.
4. Click the **Remove Document** button in the **Documents pane** toolbar.
5. Select the project (.d2h) file in the **Team Authoring** window.
6. Click the **Check In** button. The **Check In** dialog box will open. Confirm the check in and click the **Check In** or **OK** button.

Checking out the Project File

In most cases, you do not need to check out the Doc-To-Help project (.d2h) file. However, there are some cases when this is necessary. If you need to make the following changes, check the project out, make the changes and check it back in.

- Delete existing documents
- Change existing document properties
- Change project properties
- Change Help Target properties
- Change style properties
- Change topic properties
- Edit the table of contents

Note: You do not need to check out the project file in order to build your project. When you click the **Build** button, Doc-To-Help will give you the option to temporarily suspend team authoring to perform the build. Click **OK** to accept this option. In that scenario, any changes made to the project file will be temporary. If you would like them to be permanent, check out the project file before building the project.

To check out the project file

1. Select the project (.d2h) file in the **Team Authoring** window.
2. Click the **Check Out** button. The **Check Out** dialog box will open.

   For File and Web Repositories, after the file is checked out, click the **Close** button.

   For TFS Repositories, you need to confirm the **Lock type** before you check out the file. Choose **Unchanged**, **Check out** or **Check in** and click the **Check Out** button. **Unchanged** will keep the existing lock, **Check Out** will prevent other users from checking the file in or out, **Check in** allows other users to check out the file but they will not be permitted to check it back in. To set the default **Lock type**, use the Doc-To-Help **Options** dialog box (**File** tab **>** **Tools** **>** **Options** **>Team button**).
3. After edits are made to the project, select the project (.d2h) file in the **Team Authoring** window.
4. Click the **Check In** button. The **Check In** dialog box will open. Confirm the check in and click the **Check In** or **OK** button.

Working Offline

There are occasions when you may want to take your working project offline. This is convenient if you are planning to edit your documents at home or on a trip. Please note that if you check out or lock any documents before going offline, you will prevent other members of the team from editing those files until you check them back in.

To work offline

1. Open the working copy of the project.
2. Click the **Go Offline** button. The **Go Offline** dialog box will open.
3. Choose the offline mode:
   - **Restricted Project Editing** — You can add items to the project but cannot modify items that existed on the server at the time you went offline. When you go back online, Doc-To-Help will merge the changes made by other team members while you were offline.
   - **Unrestricted Project Editing** — Unrestricted changes to the project cannot be merged with other team members while you were offline. When you go online, you will be prompted either to check out the project to send your version of the project to the server (that can cause others to lose their changes) or to undo your changes.
4. Click the **OK** button. If you chose **Restricted Project** editing, the **Unrestricted Project Editing** button will appear in the **Team Authoring** tab. It allows you to change your mode if you wish.
5. Edit the documents.
6. To return to online mode, click the **Go Online** button. In TFS projects, the **Go Online** dialog box will open and display the changes that occurred while offline. If you approve, click **Go Online**.

If you need to completely disconnect your working copy from team authoring support, the Administrator must detach the project. See Removing Team Authoring Support from a Working Copy on page 393._D2HLink_618431

Getting the Latest Version of a File

It is possible to retrieve the latest version of a file (including a project file) from the server. TFS projects have the additional option to retrieve a specific version of a file from the server. Getting the latest version does not check out the file, but synchronizes your working copy with the server.

To get the latest file version

1. Select the file in the **Team Authoring** window.
2. Click the **Get Latest Version** button.

   If you are using a File or Web Repository and you selected the project (.d2h) file, the **Get Latest Version** dialog box will open. Confirm the information and click the **OK** button.

In TFS projects you have the option of retrieving a specific version of a file from the repository. To do so, click the drop-down button on the **Get Latest Version** button and choose **Get Specific Version**. The **Get Specific Version** dialog box will allow you to choose the files to get by a variety of options, including **Date** and **Changeset**. After making your selection and choosing the appropriate overwrite option, click the **Get** button.

Comparing Documents (TFS repositories only)

In TFS repositories, you can compare two versions of a document.

To compare documents

1. Select the file in the **Team Authoring** window.
2. Click the **Compare** button. The **Compare** dialog box will open.
3. Choose the file versions and click the **OK** button.

To specify the compare and merge tools for team authoring, use the Doc-To-Help **Options** dialog box (**File** tab **> Tools > Options >Team button**).

Viewing the Document History (TFS repositories only)

In TFS repositories, you can view the history of a document.

To view document history

1. Select the file in the **Team Authoring** window.
2. Click the **History** button. The **History** dialog box will open.
3. Review the information and click the **Cancel** or **Apply** buttons.

You can enter a comment on a file using this dialog box.

Viewing File Properties (TFS repositories only)

In TFS repositories, you can view the properties of a document.

To view file properties

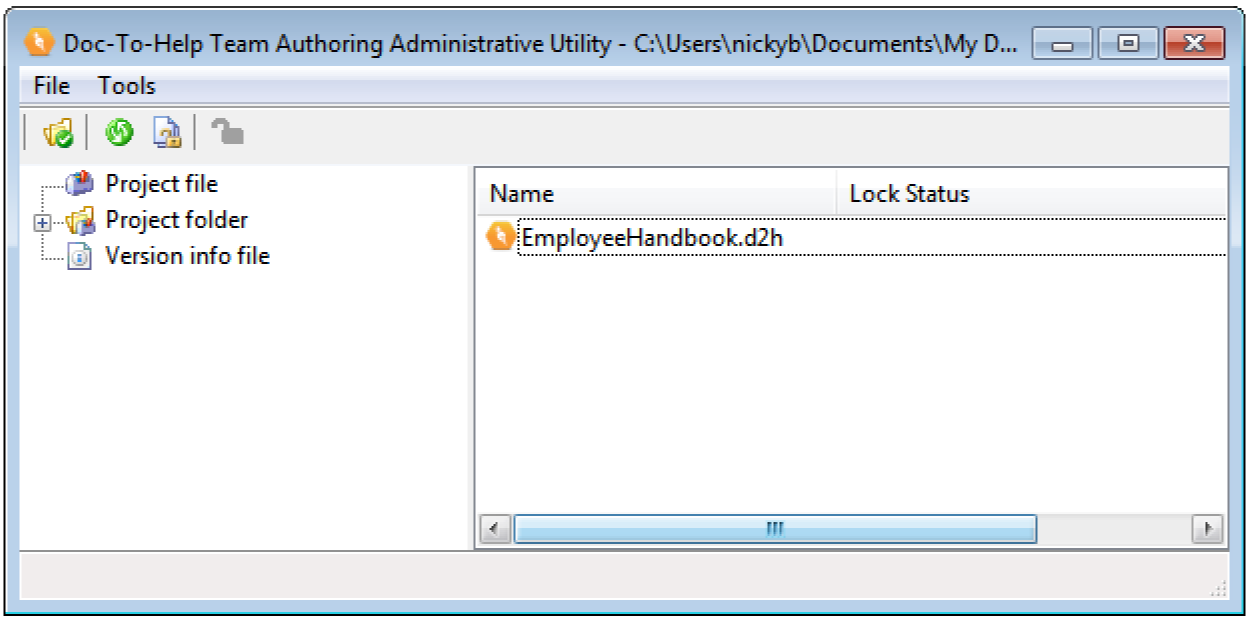
1. Select the file in the **Team Authoring** window.
2. Right-click and choose **Properties** from the menu. The **Properties** dialog box will open.
3. Review the information and click the **Close** button.

Team Authoring Administrative Functions

A special administrative utility is included when Doc-To-Help is installed. This utility, **C1D2HTeamAdmin.exe**, should be used by the Doc-To-Help administrator to perform certain actions, such as removing backup files, unlocking files from the repository, upgrading a team-authoring project, or removing team-authoring support from a working copy of a project.

Note: This utility is only used with team projects stored in a File System or Web Repository, not projects stored in Microsoft Team Foundation Server (TFS) repositories.

C1D2HTeamAdmin.exe can be found in \\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp. Double-click on it to open.



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 backups 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. If you wish, backup those folders elsewhere before deleting them.

To delete up backup files in the repository

1. Double-click C1D2HTeamAdmin.exe on page 391.

2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
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 > Clear Repository Backup**. The **Clear Repository Backup** dialog box will open.
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 394.

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 could cause a problem if an author checks out a file and forgets to (or cannot) check it back in. Then the file remains permanently locked, preventing the other team members from checking it out and modifying it. The administrator can manually remove the lock using **C1D2HTeamAdmin.exe**.

To unlock files in the repository

1. Double-click C1D2HTeamAdmin.exe on page 391.
2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
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 displays 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. Select the folder from the **Project Folder** node on the left pane, then select the 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 by 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. Normally, these locks are only temporary. However, if a team-authoring action is not normally closed due to an unexpected crash or power failure, it may leave the project permanently locked in the repository. If this happens, no team members will be able to use the repository, and Doc-To-Help will display 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 can use **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 391.
2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
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 (including build number) of Doc-To-Help installed on their machines. To upgrade the project to a newer version of Doc-To-Help, the administrator must first upgrade the team project in the repository and then each team member can install the newest version of Doc-To-Help.

To upgrade the team project to a new version of Doc-To-Help

Note: The following steps should be performed by the administrator.

1. Install the new version of Doc-To-Help.
2. Double-click C1D2HTeamAdmin.exe on page 391.
3. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
4. Select the **Repository Type**. (If necessary, enter your login credentials.)
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**.

Once the project is upgraded to the current version of Doc-To-Help, all team members should install the same version of Doc-To-Help, open their projects, and choose the **File** tab **> Team Authoring > Get Latest Version of the Project**.

Removing Team Authoring Support from a Working Copy

You can remove team-authoring support from a local, working copy of a project.

Removing team-authoring support is different from disabling team-authoring support (usually done when building a Help target) or working offline on page 389. If you want to detach your working copy of the project from the repository and remove all team-authoring functions (including 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 391.
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 using **C1D2HTeamAdmin.exe**.

To change the repository settings

1. Double-click C1D2HTeamAdmin.exe on page 391.
2. Select **File > Open Team Project** to connect to the repository. The **Load Remote Project** dialog box will open.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
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. Edit the following:
   - **Limit the number of backups for each file** — Used to set the limit on the number of backups kept for each file in the repository. The default is 5, but you can change it in the **Number of backups kept** box. If you do not set a limit, every time a file is checked-in, a new backup file will be 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 check box to permit other team members to delete old backup files when they attempt to check in a newer version of a file and the backup limit 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 a team member deletes old backup files. This option is only available when the **Allow users to delete old backups in check-in** check box has been selected.
   - 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 381 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 399 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 **index.html** file to open the project. (This file name can be renamed in the **Default name** field of the **Help Targets** dialog box.)

Eclipse Help

- The entire contents of the hub project EclipseHelp folder.

WinHelp

- Hub project .cnt file
- Hub project .hlp file
- All child project .cnt files
- All child project .hlp files

Microsoft Help Viewer

- See Creating a Modular Help Project for Microsoft Help Viewer Targets.

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 133 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).

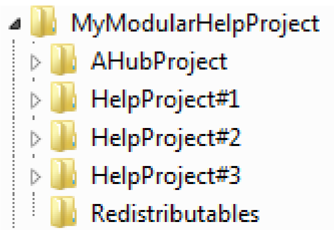
File Organization

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 399 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 182 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 118, 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 133 for more information.)

10. Choose the Help Target (can be HTML Help, NetHelp, Eclipse Help, 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 399 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 Classic** targets, you need to specify the **Module file** (.npj) field of the child project for the placeholder topic. (The .npj file can be found in the child project's NetHelp\ProjectInfo folder.)
   - For **NetHelp 2.0** and **Eclipse Help** targets, you need to specify the **Module file** field of the child project for the placeholder topic. The **Settings.xml** file is used and can be found in the child project's NetHelp or EclipseHelp 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 with the same version of Doc-To-Help. Please note that all of the projects must build the same version of NetHelp — NetHelp Classic or NetHelp 2.0.

Creating a Child Project

Child projects are tied together with the hub project (see Creating a Hub Project on page 399). 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 399 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 133 for more information.)
4. Choose the Help Target (can be HTML Help, NetHelp, Eclipse Help, 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 401.

Child projects using automatic context ID creation need to have offsets added, see Modifying Context IDs in Modular Projects on page 401.

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 402.

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 125 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 305).
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 182 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 182 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 133 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 > MadCap Software >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 399 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.)

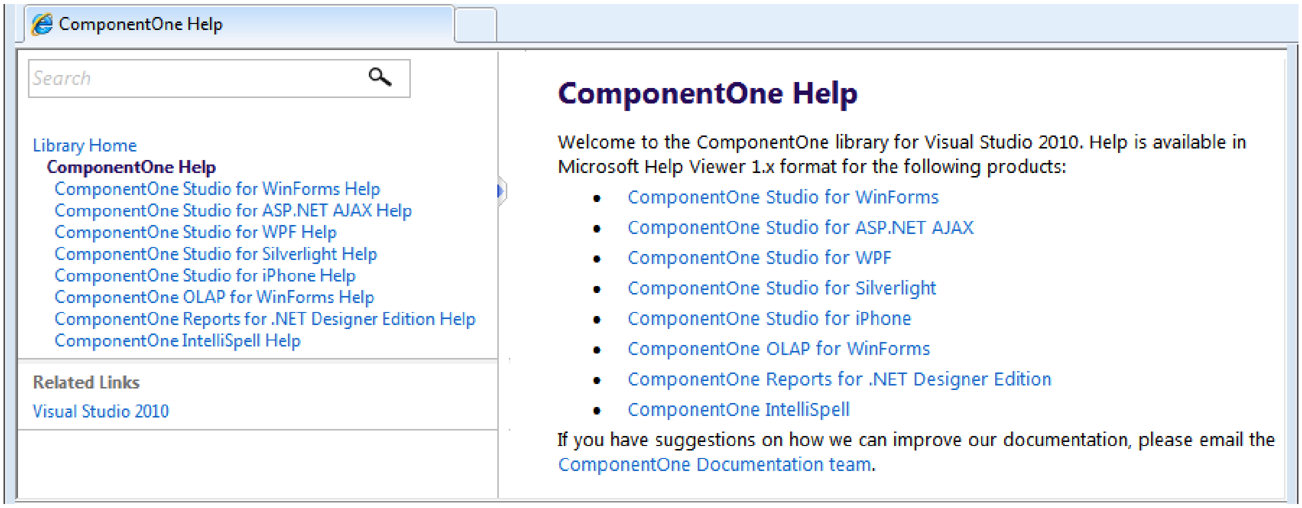
   The Modular TOC Utility will list all the child **.**hhp files in your modular Help system.
7. Choose the first child .hhp file, then click **Next**. Choose the location of the appropriate .hhp file and click **Open**. (This file is located by default in the **HTMLHelp** sub-directory of your child Help 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.

Creating a Modular Help System for Microsoft Help Viewer Targets

The Microsoft Help Viewer Target is exclusively for integration with the Microsoft Visual Studio 2010 (and above) Help system. MHV Targets usually include API reference documentation generated using the Doc-To-Help Sandcastle plug-in.

The purpose of a MHV modular project is to combine the tables of contents of multiple projects so that they appear as a single TOC when integrated with Visual Studio 2010 (and above) Help. For example, a company develops almost 300 controls for Visual Studio. These controls are grouped into several studios. In order to create a single TOC for the products within Visual Studio's Help system, we create a MHV Modular Help project. Not only is a merged TOC created, but the multiple MHV 1.x setups (HelpContentSetup.msha files) are merged into a single setup. (See Doc-To-Help Outputs and Deliverables on page 13 for information about MHV setups and deliverable files.)



To create a MHV Modular Help project

1. Set up your modular Help file structure in Windows Explorer (see File Organization on page 399). You will not need the **Redistributables** folder.

   If you have already created your child projects, simply drag them into this file structure. Make sure all of the child projects have their Microsoft Help Viewer targets built.

2. Create a new Help project that will serve as the hub or parent project. The project name should contain no spaces.
3. Open the **Project Settings** dialog box for the hub project. (See Setting Project Properties on page 182 for more information.)
4. Select the **Is modular hub project** check box to designate this project as a Modular Hub Project.
5. Close the **Project Settings** dialog box.
6. Open the hub project's source document and create placeholder topics for each of the child topics you wish to associate with the hub. The example project above includes the "ComponentOne Help" topic, as well as a topic for each of the studios listed. The "ComponentOne Help" topic uses the Heading 1 style; the studio topics use the Heading 2 style.
7. Build the Microsoft Help Viewer target.
8. Open the **Topics** window, right-click on each of the topics, and select **Properties** from the menu. In the **Topic Properties** dialog box, set the following in the **Module Link** section:
   - In the **Module file** field, choose the manifest file (.msha) that the Topic should link to.
   - If you would like the topic chosen to be the parent of all the topics in the child project, select the **Use first topic as parent** check box.
   - If you would like **Product name** listed in the **Help Targets** dialog box for your project to be inherited by all the children, select the **Inherit Product name** check box. (It is selected by default.)
   - If you would like the **Book name** property listed in the **Help Targets** dialog box for your project to be inherited by all the children, select the **Inherit Book name** check box.
9. Rebuild the Microsoft Help Viewer target.

Notes:

- You can map a Topic to ".msha" files that were created in another modular hub project. This makes it possible to create nested (multi-level) hub projects.
- All files in the MHV target folders (**MSHelpViewer** by default) are required to create a successful setup.

Limitations:

- The content of child setups must contain ".mshc" files (not ".cab" files).
- Breadcrumbs are not supported in the parent and child setups.

Manifest list file format:

If you don't want to use an ".msha" file built by Doc-To-Help, you can manually create a ".msha" file with any list of child setups and use it. See the example below.

A manifest list file is used to list all child setup manifests for a single hub topic.

This file is a standard XHTML file but must have the ".msha" extension.

Example:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>ComponentOne Studio for WinForms Help</title>
  </head>
  <body class="manifest-list">
    <div class="manifest">
     <a class="manifest-link"
href="C1BarCode/HelpContentSetup.msha">ComponentOne BarCode for .NET Help</a>
    </div>
    <div class="manifest">
       <a class="manifest-link" href="C1Zip/HelpContentSetup.msha">ComponentOne
Zip for .NET Help</a>
    </div>
  </body>
</html>
```

```
Elements used in the file:

<html xmlns="http://www.w3.org/1999/xhtml"> - required.
<head> - optional (not used).
<title> - optional (not used).
<body class="manifest-list"> - required .
<div class="manifest"> - required (one or more).
```

```
<a class="manifest-link" href="HelpContentSetup.msha">ComponentOne BarCode for
.NET Help</a> - required (only one in a manifest div). Its text is optional, not
used.
```

MHV resources:

- **MSHelpWiki.com**: http://www.mshelpwiki.com/wiki/tiki-index.php?page=Help3
- **MSDN Forum**: http://social.msdn.microsoft.com/Forums/en-US/devdocs/threads
- **Yahoo! Group**: http://tech.groups.yahoo.com/group/MSHelpViewer/

Documenting Your Class Library with Microsoft® Sandcastle

Microsoft's Sandcastle utility automatically creates MSDN style 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.

To use the Doc-To-Help Sandcastle plug-in, you must first install Sandcastle and the Sandcastle patch. If you have an older version of Sandcastle on your machine, uninstall that version first.

1. Download the **Sandcastle Help File Builder version 1.9.5.0** from http://shfb.codeplex.com/releases/view/92191, then run InstallResources\Sandcastle_2710.msi.
2. After you have completed the **Sandcastle Help File Builder** installation, go to
    - 32-bit machine: C:\Program Files\MadCap Software\DocToHelp\Plugins\Sandcastle
    - 64-bit machine: C:\Program Files (x86)\MadCap Software\DocToHelp\Plugins\Sandcastle

    and double-click on **SandcastlePatch.exe** to install it.

Note: Class libraries cannot be exported to Manual or EPUB Targets. It is suggested you exclude the Sandcastle plug-in from those outputs using a condition. See Setting Document Properties on page 297 for more information.

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 410.

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® 7/8 and 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** (**File** tab > **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 168 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 411 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. (If any of the dependent properties are changed later, Doc-To-Help will automatically regenerate the reference section when the plug-in is closed, or when building the Target.)
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 424 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 168), 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.

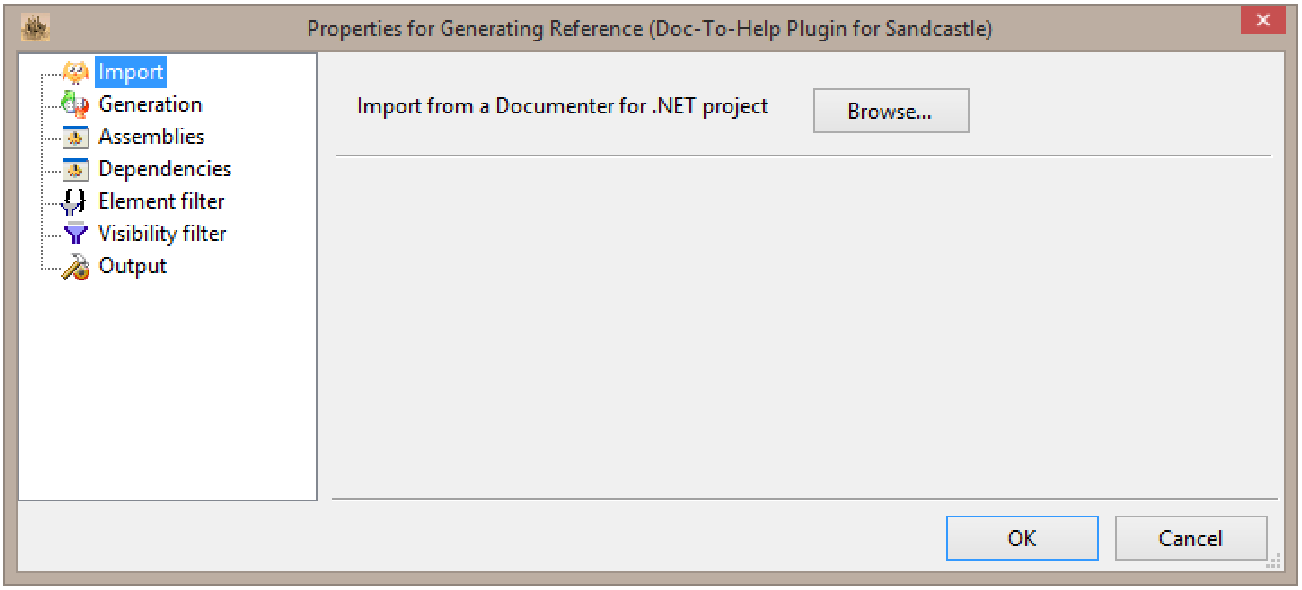
The dialog box is broken into seven nodes: **Import**, **Generation**, **Assemblies**, **Dependencies**, **Element filter**, **Visibility filter**, and **Output**.

A few notes about creating **Microsoft Help Viewer** Targets with the Doc-To-Help Plugin for Sandcastle:

- Doc-To-Help supports only self-branded targets. This option is set using the **Self-branded** check box in the **Help Targets** dialog box.
- In the Output node, Doc-To-Help supports only the **VS2005** presentation style.
- In the Output node, set the **Links to .NET Framework documentation** to **None**.

Import node

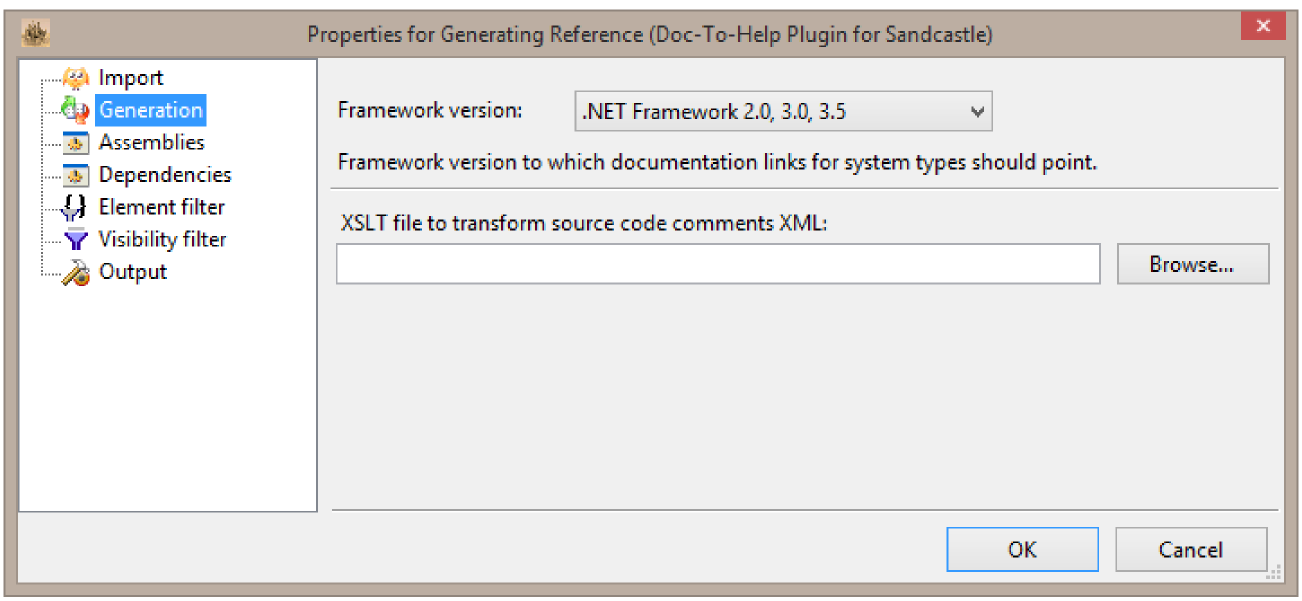
The **Import** node allows you to import a Documenter for .NET project — including all of its properties — which will overwrite the default properties for the Sandcastle plug-in document.



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.

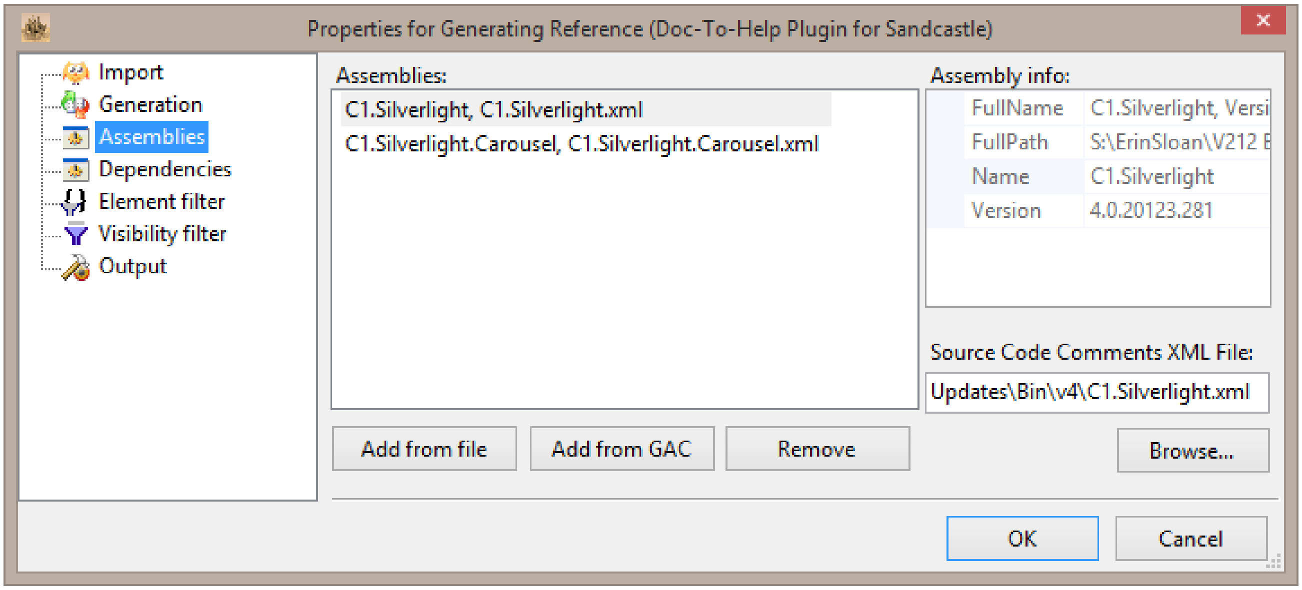


Framework version: Select the .NET Framework version you want to link to when a system type appears in your documentation. Select version 2.0, 3.0, 3.5, 1.0, 1.1, 4.0, Silverlight 3 or Silverlight 4.

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.

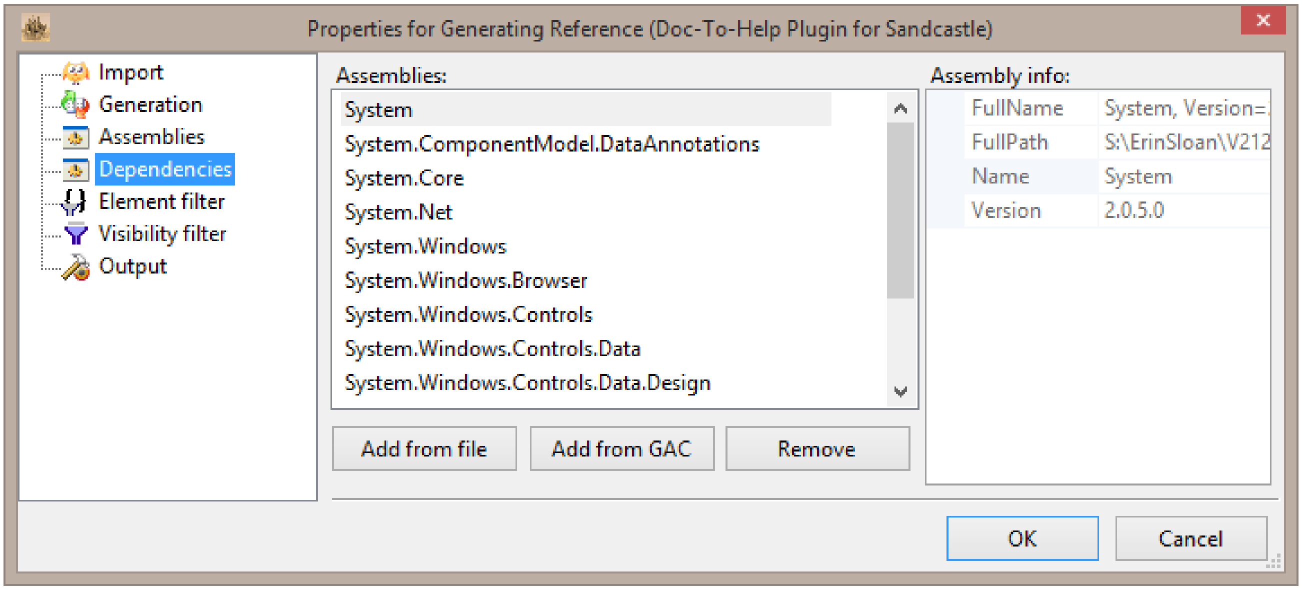
Add from GAC: Click this button to add an assembly from the Global Assemblies Cache (GAC). Strong-named assemblies must reside in the GAC in order to be documented.

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.



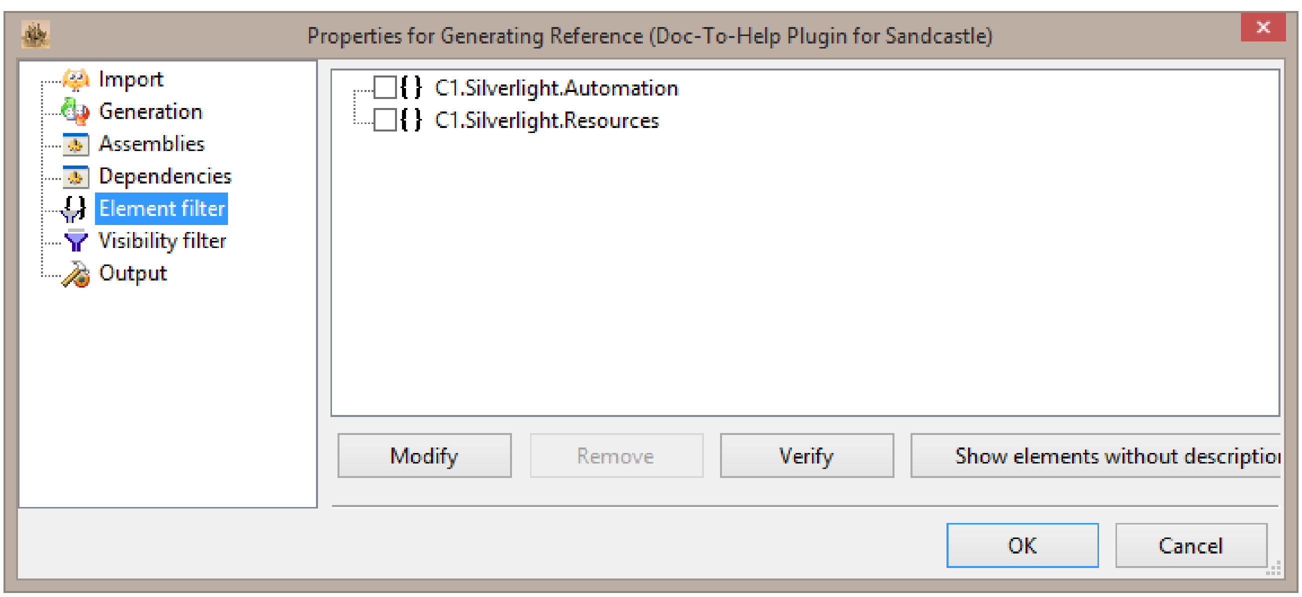
Add from file: Click this button to add an assembly from a .dll on your machine.

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.

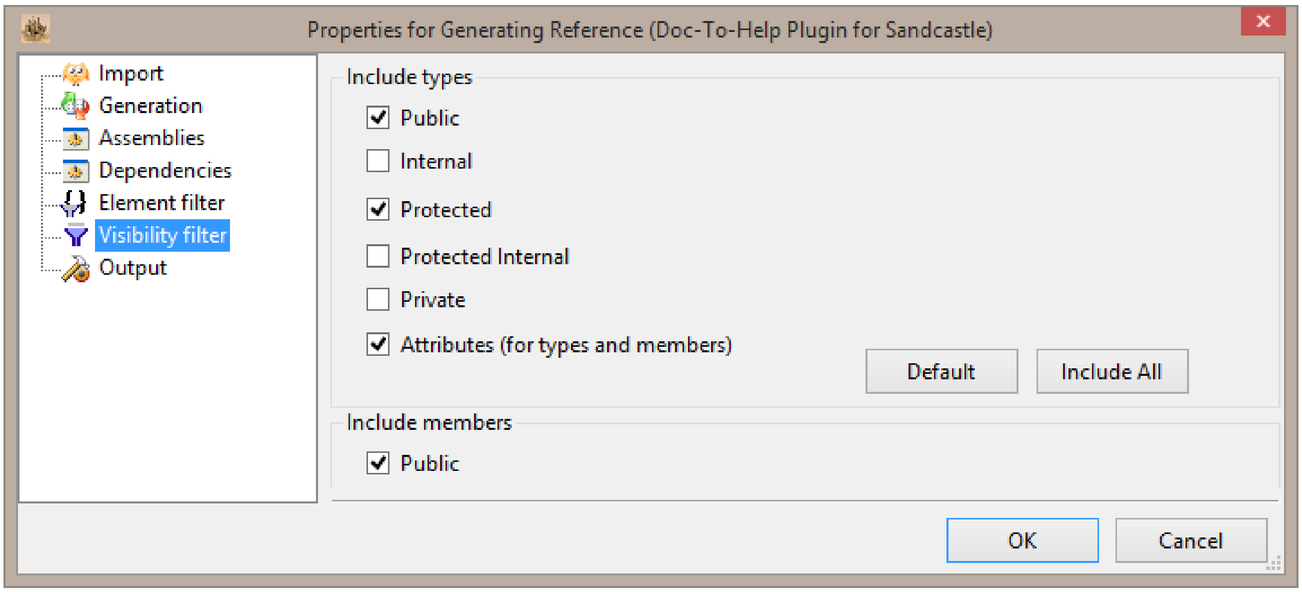


Modify: Click this button to access the **All Elements** dialog box. From the **All Elements** dialog box, you can easily add elements to the **Element filter** by clearing their check boxes. You can also click the **Exclude Elements Without Description** button to remove any elements that don't have 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

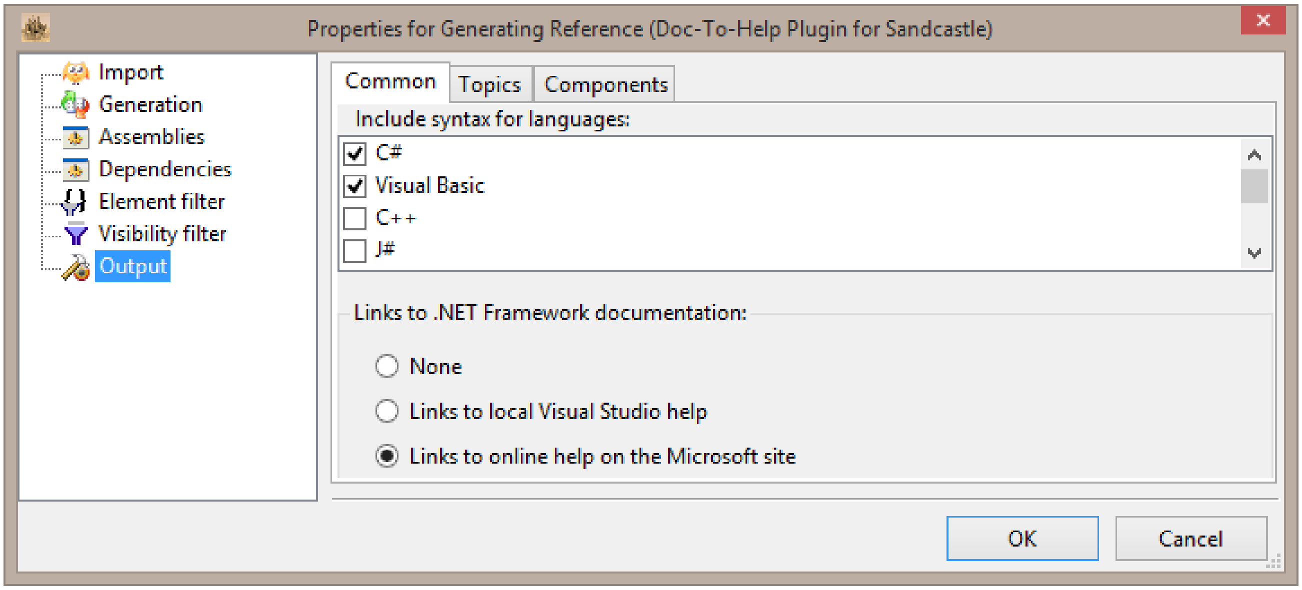
In the **Include members** section, select the check boxes for the members that you would like to appear in your reference section. You can include the following types of members in your reference section: Public, Internal, Protected, Protected Internal, Private, and Explicit Interface Implementations.

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.

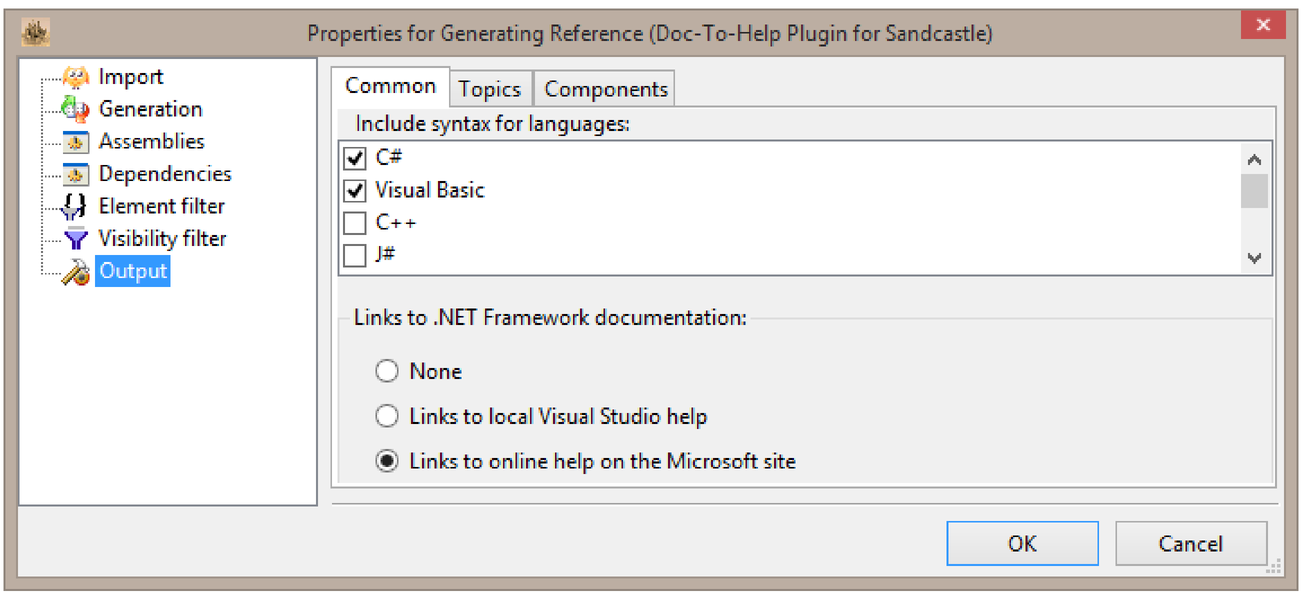
Output node

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.

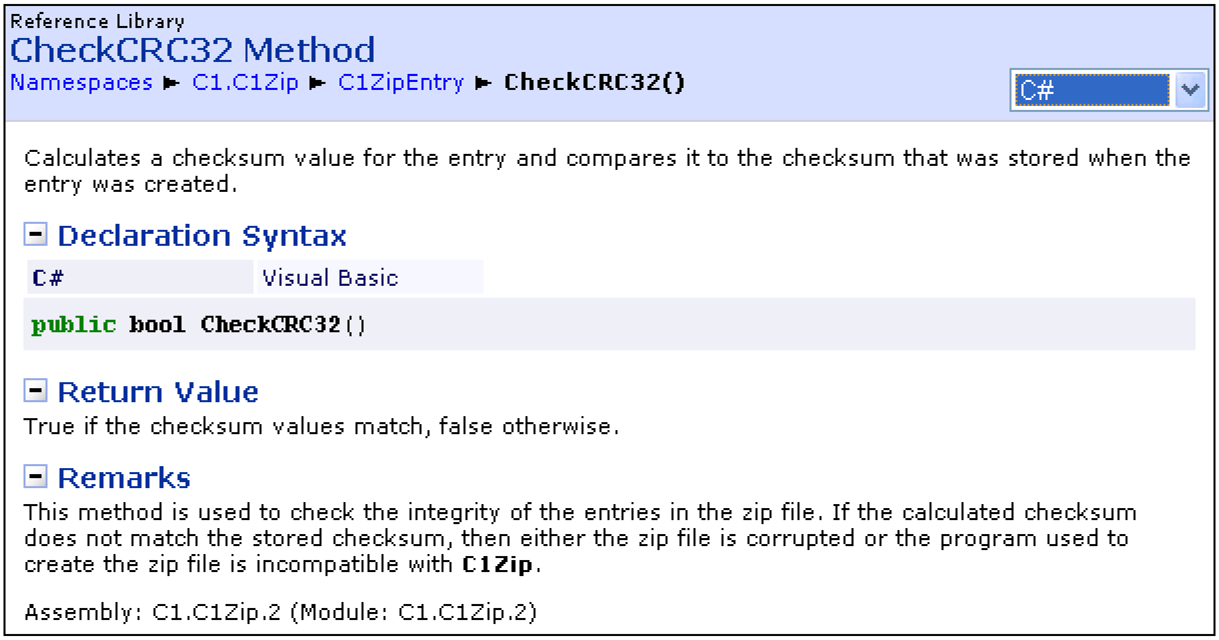
Links to .NET Framework documentation: Select the check box for the type of links Sandcastle should create when linking to the .NET Framework documentation.

- None
- Links to local Visual Studio help
- Links to online help on the Microsoft site

Presentation style for topics: Choose one of the built-in themes provided by Sandcastle.

Topic file names: Choose Friendly or GUID.

- **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

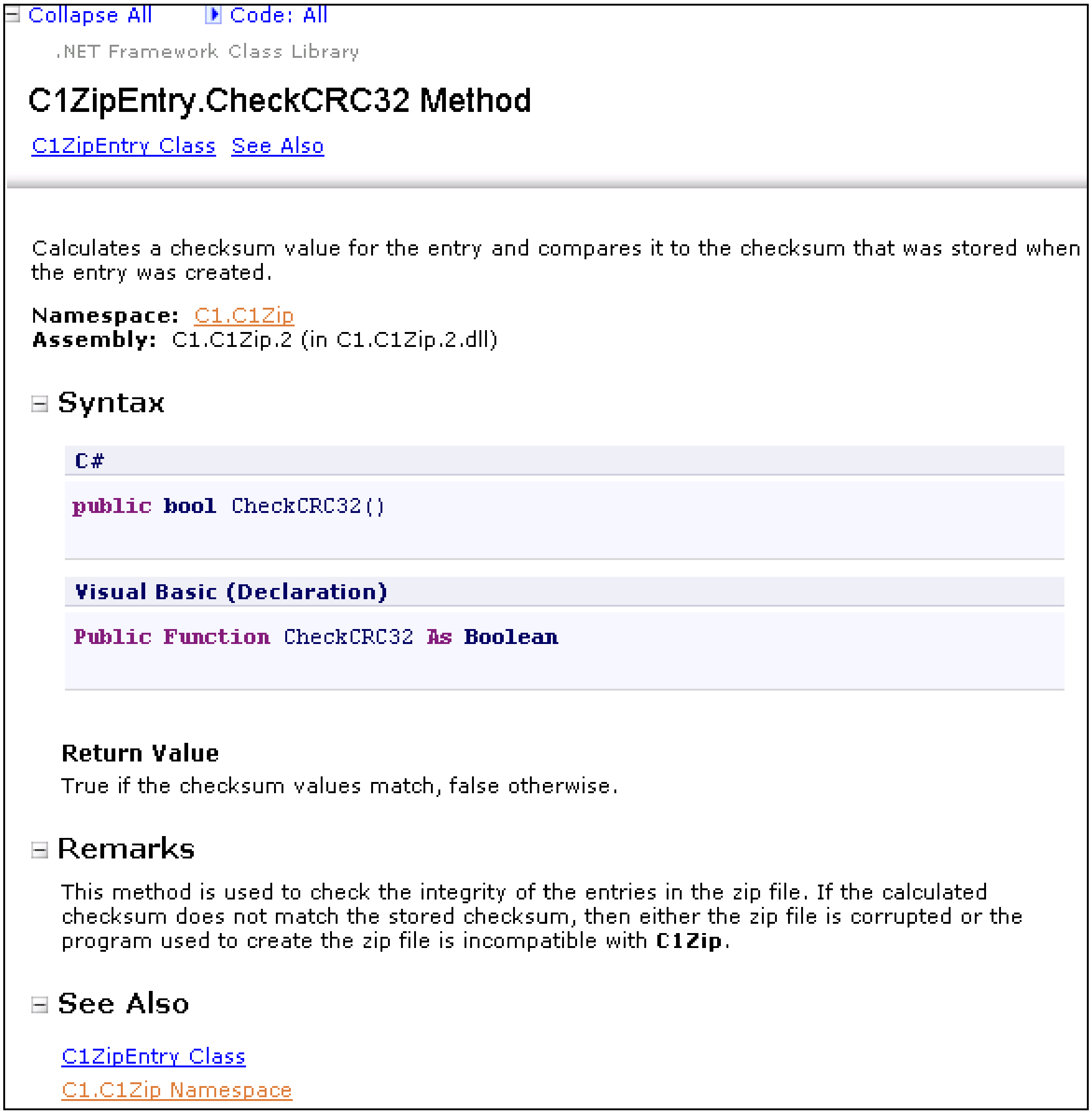
True if the checksum values match, false otherwise.

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.

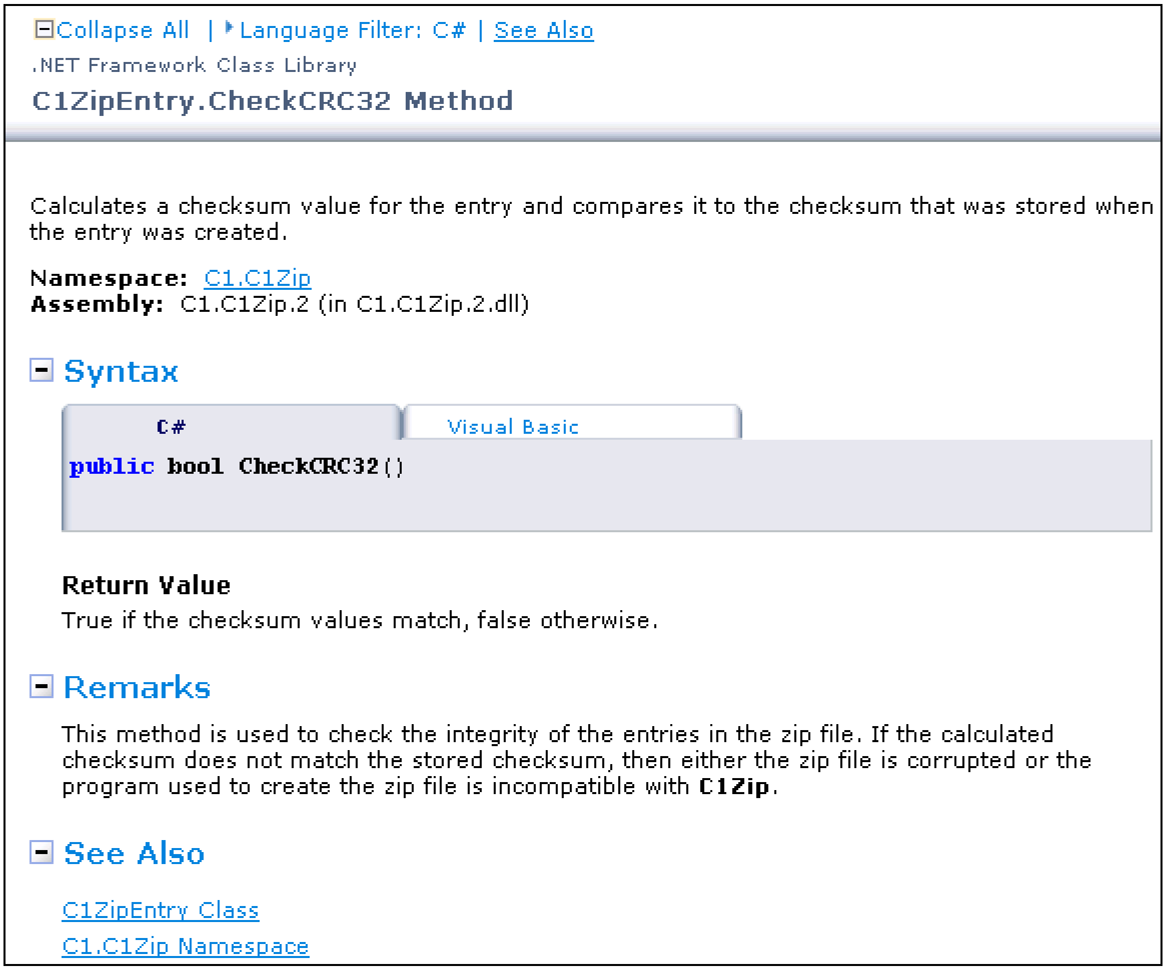
Remarks

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 "Doc-To-Help" into this field, the feedback footer statement would read "Send feedback on this topic to Doc-To-Help."
- **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 423.

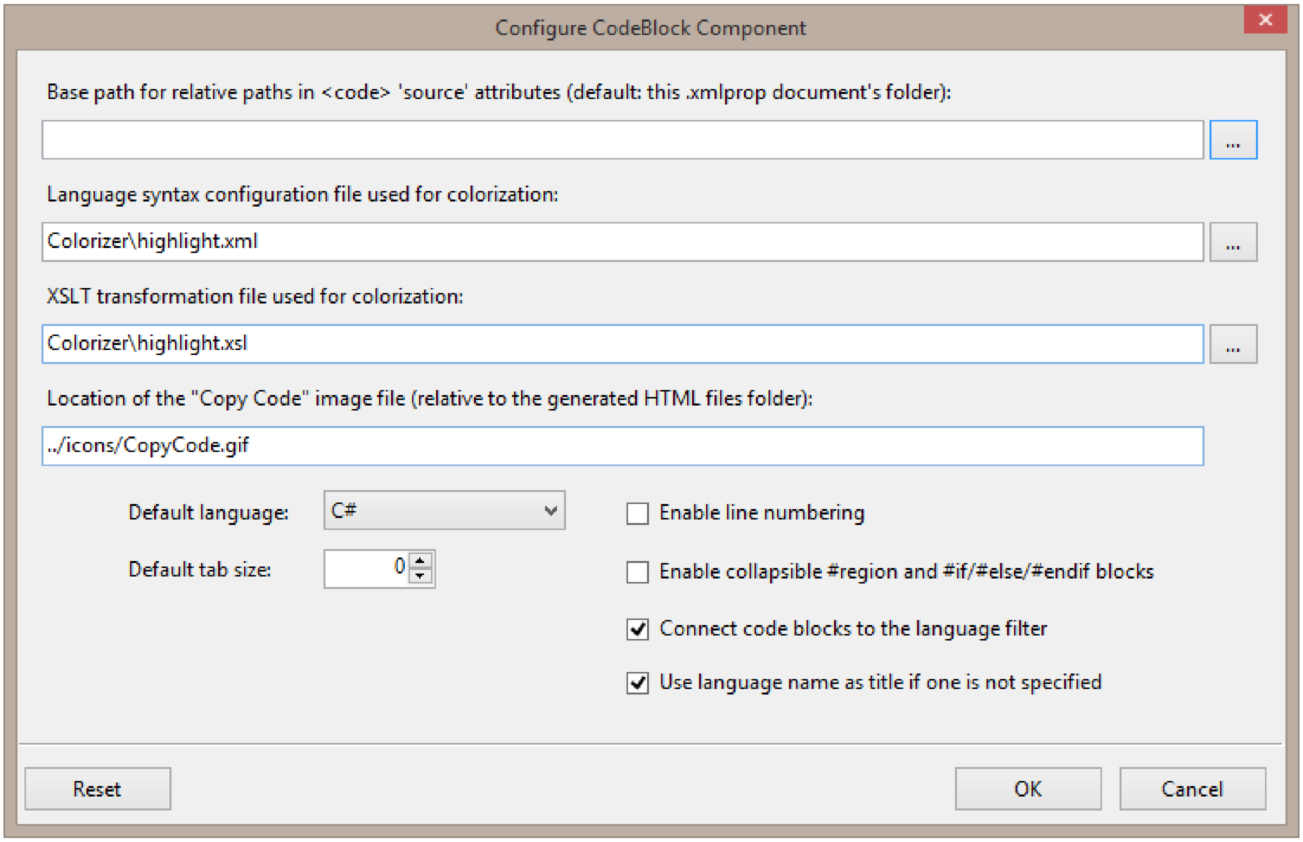
Click the **Configure** button next to its corresponding component to configure that component's settings. The **CodeBlock** component's **Configure** button opens the Configure CodeBlock Component on page 420 dialog box, while the **PostTransform** component's **Configure** button opens the Configure PostTransform Component on page 422 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 411 for more information.

Also see CodeBlock and PostTransform components on page 423.



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.

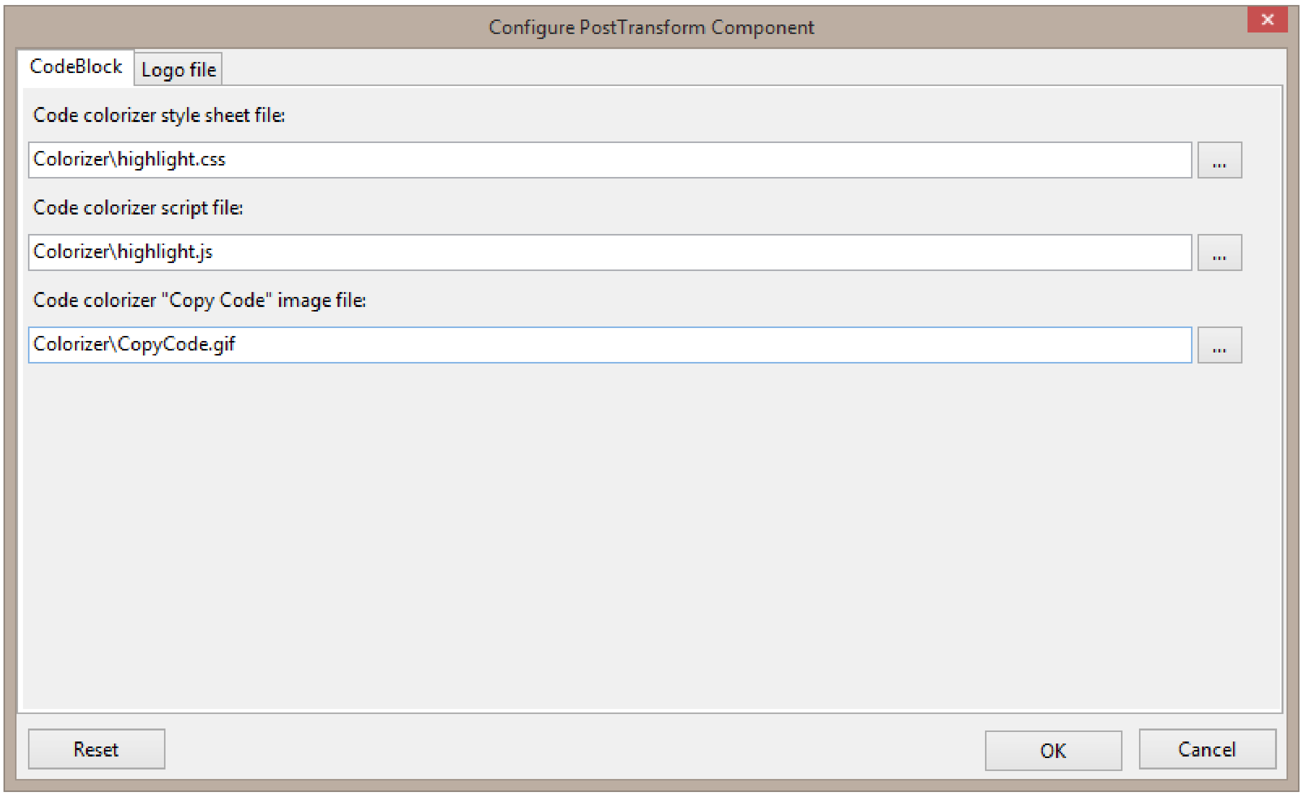
Reset: Click the **Reset** button to return to the **CodeBlock** component's default settings.

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 411 for more information.

Also see CodeBlock and PostTransform components on page 423.



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 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/Articles/16740/Creating-Custom-Build-Components-for-Sandcastle.

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

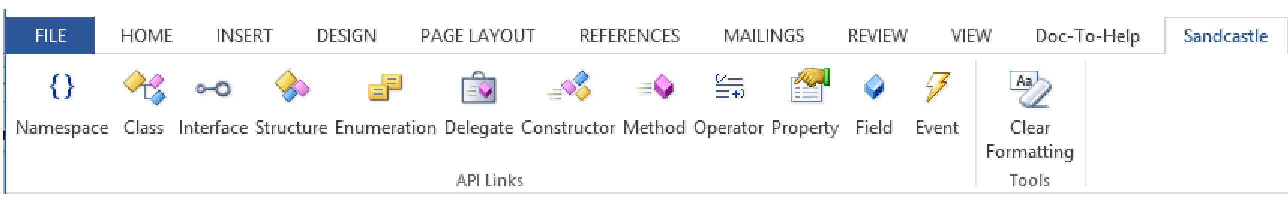
To exclude elements that do not have descriptions (to add them to the filter), go to the Element filter node of the **Properties for Generating Reference** dialog box (double-click on the plugin document to open the **Properties for Generating Reference** dialog box).

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 303 **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 431, 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 431.

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 431. 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 431. You can also use the argument list for constructor overloads, see Argument List in Links on page 432.

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 431, 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 431.

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 431, 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 431.

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 431, 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 431.

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 431, 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 431.

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 431, 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 431.

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 431, 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 431. You can also use the argument list for method overloads, see Argument List in Links on page 432.

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 431. 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 431. You can also use the argument list for operator overloads, see Argument List in Links on page 432.

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 431, 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 431. You can also use the argument list for property overloads with arguments, see Argument List in Links on page 432.

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 431, 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 431.

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 431, 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 Windows® 7/8 and Vista** at C:\Users\(user name)\AppData\Roaming\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:

- 32-bit machine: C:\Program Files\MadCap Software\DocToHelp\Backup\Templates
- 64-bit machine: C:\Program Files (x86)\MadCap Software\DocToHelp\Backup\Templates

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, NetHelp, Microsoft Help Viewer, 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 168.)

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 303 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>
```

Doc-To-Help Frequently Asked Questions

Doc-To-Help's FAQs cover a number of topics.

FAQs: General Questions

What Outputs (Deliverables) can I create with Doc-To-Help?

- NetHelp (uncompiled HTML Help that opens in a browser)
- HTML Help (compiled HTML Help, aka .chm or "chum")
- EPUB
- Eclipse Help
- JavaHelp
- WinHelp
- Two types of Manuals, Word and PDF

- Microsoft Help Viewer
- Microsoft Help 2.0

See Doc-To-Help Outputs and Deliverables on page 13 for more information.

What browsers can users view NetHelp output in?

NetHelp 2.0 supported browsers:

- Internet Explorer 8 or higher
- Firefox 3 or higher
- Opera 9.6 or higher
- Apple Safari 3.1 or higher
- Google Chrome (all versions)

NetHelp 2.0 Responsive Theme supported browsers:

- Desktop: Internet Explorer 8 or higher
- Mobile platforms: Android 4.0 or higher, iOS

NetHelp Classic supported browsers:

- Internet Explorer 8 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

See NetHelp Installation Requirements.

I need to create an output that can be hosted on my web server, but still be accessed through the application.

You should create NetHelp, which is an HTML output that is browser-based and can be hosted locally or on any web server.

Where do I begin when I open Doc-To-Help?

You can get started quickly in Doc-To-Help using the **Getting Started Wizard**. Just click the button on the upper right to open it . In just a few clicks, you can create a project. This Wizard also leads you to sample projects, and to many Doc-To-Help resources (videos, webcasts, training, and more).

Are there any sample Doc-To-Help projects I can work with?

Doc-To-Help includes 11 sample projects that demonstrate the editors, plus are examples of software documentation, an employee handbook, training materials, Responsive Help, and API/SDK documentation with Microsoft Sandcastle. There is even a sample in German.

These projects can be found here, and can also be opened using the **Getting Started Wizard**:
Windows® 7/8 and Vista: \\Users\<username>\Documents\My Doc-To-Help Projects\Samples

These path names may be different depending on the language version of your operating system.

FAQ: File Size

I built my Target and the file size is significantly larger than the last build, even though I only changed one sentence. Why?

This is because there are too many files in the **Temp** folder of your Doc-To-Help project. Open Windows Explorer, delete the **Temp** folder, and rebuild the project. The file size should be reduced.

FAQs: Styles

Do I need to have Styles applied in my documents to use Doc-To-Help?

Yes, but not too many to start off with. Doc-To-Help uses Heading Styles (Heading 1, 2, 3, 4, 5, 6) to divide your document into Topics in online Help outputs. Doc-To-Help also uses these Styles to create your Table of Contents. So, you need to use at least the Heading 1 Style in your documents to create online Help and Manuals. You can then use Heading 2, 3, 4, 5, and 6 if you need to. However, if you have older documents that use other heading styles (for example, you named Heading 1 "Main Heading"), you don't need to change all of your documents, simply open the **Project Styles** dialog box (click the button on the **Project** ribbon in Doc-To-Help) and add that style to the list of Paragraph Styles, and assign it the **Topic Type** of "Contents."

I keep getting an error message that says my topic titles are too long.

You most likely applied a Heading Style (Heading 1, 2, 3, 4) to an entire sentence or paragraph. Check the topic and if this is the case, change the style for the sentence or paragraph to Body Text or another non-Heading Style.

Some of my Help Topics open in a separate window. How do I stop that?

Topics that begin with certain heading styles (for example: Heading 3, 4, and 5) may open in a secondary window called "proc" if their Topic Type is set to do so. To change this, open the **Topic Types** dialog box (click the button in the **Project** ribbon in Doc-To-Help) and for the Procedural **Topic Type**, change the Window to "Main."

FAQs: Templates and CSSs

Doc-To-Help uses Templates and CSSs to transform your documents into Help Outputs and Manuals. Doc-To-Help includes these Templates and CSSs with the product. You should always start your project with our included templates/CSSs and customize them as you like.

What is my project using — a Template or a CSS?

- Doc-To-Help projects authored in Microsoft® Word (.doc or .docx) files are transformed using Word templates (.dot or .dotx files).
- Doc-To-Help projects authored in HTML are transformed into online outputs using cascading style sheets (.css files). They are transformed into manuals using Word templates (.dot or .dotx files).

Is there a difference between Source and Target Templates?

Yes! You should always use Source Templates when authoring your Word documents. You use Target Templates to transform those documents into Outputs. Never use a Target Template as a Source Template and vice versa.

What template or CSS should I edit to change the way my Outputs look?

You should only edit the appropriate Target Template or CSS. There is no need to edit your Source Templates or CSSs unless you want to, because you are the only one who sees them.

For those authoring in Word, here is the list of Word Templates included with Doc-To-Help and how they work together:

Predefined Source Templates	Predefined Target Templates (these transform your output)	
Available Source Templates	**Printed Manual Target Templates**	**Online Help Target Templates**
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_NOMARGIN_A4.DOT: This is the source template for A4 sized paper. In the New Project Wizard called “Normal A4 Template (no left indent)”	**C1H_PRNOMARGIN_A4.DOT:** This is the target template used to format the printed manual target (A4 sized paper).	**C1H_HTML.DOT**: : This is the target template used to format NetHelp, HTML Help, EPUB, Microsoft Help Viewer, Microsoft Help 2.0, JavaHelp, and Eclipse Help Targets
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 has a two inch wide left margin.	
C1H_NORM_A4.DOT: This is the source template for A4 sized paper with a two-inch left margin. In the New Project Wizard called “Normal A4 Template”	**C1H_PRNORM_A4.DOT**: This is a target template used to format the printed manual target (A4 size paper).	
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	**C1H_PRSMAL.DOT**: This is the target template used to	

small-sized manuals. In the New Project Wizard called "Small 7" x 9" Crop-Marked Template"	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).	

For those authoring in HTML5 or HTML, here is the list of CSSs and how they work together:

Predefined Source Style Sheets	Predefined Target Style Sheets and templates	
Pick One	**Printed Manuals**	**Online Help** **Pick One for HTML5 or HTML/One for WinHelp**
C1H_Source_full.css: Source style sheet with the full set of styles. In the New Project Wizard called "Full Set of Styles"	Choose a **Target Word template** (see Word File Templates on page 8) In most cases, you would choose **C1H_PRNOMARGIN.DOT**	**C1H_HTML_full.css**: Target style sheet for all HTML-based targets with the full set of styles.
C1H_Source_short.css: Source style sheet with the minimum set of styles. In the New Project Wizard called "Minimal Set of Styles"		**C1H_HTML_short.css**: Target style sheet for all HTML-based targets with the minimum set of styles.
		NetHelp.css: Target style sheet for NetHelp Targets.
		C1H_Help_full.css: Target style sheet for WinHelp target with the full set of styles.
		C1H_Help_short.css: Target style sheet for WinHelp target with the minimum set of styles.

How do I set my Source Template or Target Template?

To set a Source Template, open your Word document and in:

- Word 2007/2010/2013: **Office** button > **Word Options** button > click **Add-Ins** in the left pane > from the **Manage** drop-down list, choose **Templates** > click **Go**.

To set a Target Template:

In Doc-To-Help, open the **Home** tab, select the Target using the **Select Target** button, and choose the Target template from the **Target Template** drop-down.

Where are Word Templates stored on my machine?

In Windows® 7/8 and Vista at C:\Users\ (user name)\AppData\Roaming\Microsoft\Templates.

Path names may be different depending on the language version of your operating system.

FAQs: Images

What is the best way to insert images into documents?

In Word, click the **Picture** button in the **Insert** ribbon tab to open the **Insert Picture** dialog box. Choose your graphic from the list and click the **Insert** drop-down. Choose **Insert and Link**. This is the recommended option for Doc-To-Help projects. Cutting and pasting images into your documents is not a good best practice, because it increases file size and makes your graphics hard to edit and manage.

All graphics should be stored in the **Media** folder that is created automatically for Doc-To-Help projects.

Please note that if your source documents are **.docx** files, the **Media** folder should be a subfolder of the **Documents** folder (the folder Doc-To-Help stores all of your source documents in). This change can be made quickly using Windows Explorer.

My graphics are not displaying in my source documents. Why is this happening?

In Word, if your graphics do not display in your source documents, but the message "The linked image cannot be displayed" is displayed instead, there is a way to fix this. If the image does display in your Targets, it is there, there is a just an issue with the path to the folder the files are stored in. First, go to the **Project Settings** dialog box (**Project** tab > **Project** ribbon group dialog box launcher) and set the proper folder in the **Media folder(s)** field (in the **Advanced** section). You can also verify and change the way links are set up for graphics in your Word document. In Word 2010/2013, click the **File** tab, and choose the **Edit Links to Files** link on the far right (under **Related Documents**).

If the image does not display in the Target, it is not part of the project. You should check the path to the image by turning on **Field Codes** in Word. Sometimes the path to the graphic will point to a folder on another machine. Or the graphic may not be in that folder.

If an image was cut and pasted into a Word document, then that document is copied into a Doc-To-Help project on another machine, those images may not appear in both the source document and the Targets. You must find the original image and move it to the correct machine. If you can find the original image file, you should **Insert and Link** the image using the **Insert Picture** dialog box instead of using cut/paste.

FAQs: System Requirements

What versions of Office Does Doc-To-Help support?

Microsoft Office 2007 through 2013. Word 95, Word 97, Word 2000, and Word 2003 are not supported.

What versions of Windows Does Doc-To-Help Support?

Doc-To-Help supports Windows 7 and 8, including 32-bit and 64-bit operating systems. Vista is also supported. Windows 95, 97, NT, and XP are not supported.

What versions of SharePoint are supported?

Doc-To-Help is compatible with Microsoft® SharePoint® 2007, 2010, and 2013, as well as Office 365 (SharePoint Online). SharePoint 2010 or 2013 is required when publishing Doc-To-Help Targets to a SharePoint Wiki Library. SharePoint 2010 is recommended when working with Translation Libraries (SharePoint 2013 does not include this feature).

Are the free versions of SharePoint – WSS and Foundation – supported?

Yes. Doc-To-Help's SharePoint integration works with the free versions of SharePoint that install with some versions of Microsoft Office. Translation Libraries cannot be managed in Windows SharePoint Services 3.0, because WSS 3.0 does not support them.

What versions of Microsoft Team Foundation Server are supported?

Microsoft Team Foundation Server 2008, 2010, and 2012.

Does Doc-To-Help Support Linux?

No

Does Doc-To-Help Support Open Office?

No, but you can import Open Office documents. See the Open Office Files and Doc-To-Help blog post.

Does Doc-To-Help Work on a Mac?

No. However, there are some Macs that have a partitioned operating system -- one partition runs a Mac OS and one runs a Windows OS. In this situation, you could install Doc-To-Help on the Windows partition.

Do you have functionality similar to ClearCase or SourceSafe?

We have Team Authoring Support, which is built into Doc-To-Help. With TAS, you share your project to a file repository and anyone with Doc-To-Help installed on their PC can access the project and the documents inside the project. Once they check a document out, it is locked. As soon as they check a document back in, it is assigned a version number. So, you can track changes from version to version, but Doc-To-Help does not do this for you.

Does Doc-To-Help Integrate with ClearCase?

Doc-To-Help does not directly integrate with ClearCase, but it will not interfere with ClearCase, so you can use it with Doc-To-Help

Does Doc-To-Help Integrate with SourceSafe?

Doc-To-Help does not directly integrate with SourceSafe, but it will not interfere with SourceSafe, so you can use it with Doc-To-Help.

Does Doc-To-Help Integrate with any Versioning Software Packages?

Doc-To-Help integrates directly with Team Foundation Server and Microsoft SharePoint.

FAQs: Help Functionality

I need my customers to be able to access my Help through the application, either by clicking a help button, and icon, or by pressing F1.

You can create context-sensitive Help with Doc-To-Help. This type of Help opens a specific Help topic when the end user clicks a Help button or icon. To create context-sensitive help, the software developer embeds a Context ID (or MapID) number in the code. That number corresponds to a Context ID (or MapID) number in the topic in Doc-To-Help. You can use the Topics window to associate Context ID numbers with your topics, or you can have Doc-To-Help automatically create Context ID numbers for all your topics. See Implementing Context Sensitive Help on page 186 for more information.

Can I Create “What’s This?” Help with Doc-To-Help?

Yes. For HTML Help, apply the WhatsThis style to the topics and map them like any-other context-sensitive Help topic using Context IDs. In the **Document Properties** dialog box of the source documents containing these topics, set **Plain Text Popups** to "yes." For WinHelp outputs, apply the WhatsThis style to the topics and map them like any-other context-sensitive Help topic using Context IDs.

I want to Create Embedded Help, can I do that with Doc-To-Help?

If you are developing a desktop application in .NET, you can use Dynamic Help. This is a Visual Studio control included with Doc-To-Help that software developers can include in your software application to create a Help pane. Then you can map any HTML Help (.chm) or NetHelp file to the software interface visually. You do not need to use Context IDs for Dynamic Help. Doc-To-Help's interface includes this control on the right.

You can also embed NetHelp 2.0 Targets into web pages; see Embedding NetHelp 2.0 Targets into Web Pages on page 337.

How Do I Change the Caption in my HTML Help Projects?

The caption in HTML Help projects can be changed in the **Windows** dialog box (Click the **Windows** button on the **Project** ribbon in Doc-To-Help).

FAQs: Manuals

The first numbered list in my manual starts with the number 1, then continues from there throughout my manual. How can I fix this?

In the **Help Targets** dialog box (click the dialog box launcher on the **Home** tab, **Targets** ribbon group of Doc-To-Help) choose the **Manual** output and in the "Fix Numbered Lists" field, choose "Always."

I want my PDF to generate at the same time I build my Manual Target, how can I do this?

In the **Help Targets** dialog box (click the dialog box launcher on the **Home** tab, **Targets** ribbon group of Doc-To-Help) choose the **Manual** output and select the check box next to "Generate PDF Target."

FAQs: Authoring in Microsoft Word

My Word documents and templates have a lot of custom styles. Will Doc-To-Help be able to use these?

Yes. You can copy all of the styles from your templates over to Doc-To-Help's custom templates using Word's **Organizer**. It is fast, easy, and well documented. And you can use the **Project Styles** dialog box in Doc-To-Help to map your existing styles to Doc-To-Help Styles.

Can I convert my Word documents to HTML5 so that I can work with them in Doc-To-Help's built-in editor?

Yes. There is an option to convert existing Word documents to HTML5. Either choose the **Convert to HTML5** option when you import the documents or click the **Convert Multiple Documents to HTML5** button in the **Documents** pane in Doc-To-Help's interface.

I create mathematical equations using Word's Equation Editor and they are appearing in the wrong places (or are bolded) in Help outputs. How can I fix this?

It is best to create screen captures of each equation and insert them into your document just like any other graphic. You then have control of the appearance and placement of your equations.

I see text in my document that I don't want to. How can I turn it off?

In Word, you might have Hidden Text and/or Field Codes turned on. In pre-Word 2007 versions, choose **Tools > Options** to turn them off. In Word 2007 and after, click the **Word Options** button.

FAQs: Troubleshooting

When you build your Help Targets, any error messages will display in the **Output** window, in the tab named **Error(s) and Unresolved Links**.

The error messages will include an explanation that will help you troubleshoot the issue.

A project must have a table of contents, so if you see the error "There are no valid topics in the table of contents" you should click the **Contents** accordion button and check the Table of Contents. If there are no items in it, you should add topics by dragging them from the Topics window. If no Topics are in the Topics window, you need to check and see if your source documents contain Heading styles.

- One of the most common errors is a broken link. These will be indicated by an icon that looks like a broken chain link. Double-clicking on the description of the broken link will open the topic containing that link so that you can fix it.

- A message with a red X icon next to it must be fixed.
- A message with a yellow shield is a warning and does not need to be fixed, but you may want to investigate it.
- A message with a blue circle next to it is informational.

When working in Doc-To-Help, a message told me I could not use a feature because "This operation requires administrative rights." Why am I getting this message?

If your operating system is Windows Vista or 7/8, User Access Control (UAC) prevents some features from working unless you are logged in to Doc-To-Help as an administrator. The features include activating/deactivating Doc-To-Help, if you are not logged in as an administrator, Doc-To-Help will display the message "This operation requires administrative rights" with an explanation. To log in as an administrator, right-click on any Doc-To-Help shortcut (from the **Start menu** or on your desktop) and choose **Run as administrator** from the menu. Doc-To-Help will open.

My WinHelp Target won't build on my new machine. What is the issue?

Due to limitations with WinHelp, it will not build on 64-bit machines. If your new machine is 64-bit, you cannot build WinHelp on it.

My Eclipse Target won't build on my machine. What is the issue?

You must have 32-bit Eclipse installed to build your Doc-To-Help project; you cannot build if you have 64-bit Eclipse installed. End users can view Eclipse Help with either version, plus the Java Runtime Environment (JRE) installed. Eclipse is available for download from http://eclipse.org/downloads/.

I uploaded documents to SharePoint using the Share button in Doc-To-Help, and the next time I opened the project and attempted to synchronize the documents, I received the message "The user name or password is incorrect." Why did this happen and how can I fix this?

If you change your SharePoint password after uploading documents, you will no longer be able to synchronize your documents. You can fix this by clearing your Windows credentials.

1. In Windows, open the **Control Panel** > **User Accounts and Family Safety** > **Credential Manager**. (This is the path for Windows 7.)
2. Find the SharePoint Server credentials and remove them.
3. Restart Doc-To-Help.

You will be prompted to enter your new password the next time you try to connect to SharePoint.

I received a "Word Not Installed" error message, but I have Word installed. How do I fix this?

See Known Issues and Workarounds in the Doc-To-Help Forum.

Doc-To-Help is suddenly forcing me to save a modified version of my Word templates. Why is this happening?

It has been discovered that a specific Word Addin can cause this behavior. Check to see if you have the Word Com-Addin from Intel "Send to Bluetooth" installed. If so, deactivate it in Word.

Glossary

ALinks

See Groups.

API Documentation

Documentation for an Application Program Interface. See Sandcastle.

Associative Links

See Groups.

Automatic Subtopic Links

See Subtopic Links.

Build Target

See Target.

Cascading Style Sheet

See CSS.

Character Styles

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 source documents to customize your project. Character styles allow you to create these types of hot spots and more using Doc-To-Help Markup Language (D2HML) on page 303. Also see Defining Character/Paragraph Styles and Topic Types on page 168.

Child Project

See Modular Help System.

Compiling

Building your Target (output). See Building a Target on page 341.

Conditional Text

See Conditions.

Conditions

Marking text, documents, and/or topics as Conditional means that those items will display only in specific Targets (for example, only online Help targets, or only in specific manuals). This gives you a great deal of control for single sourcing. Doc-To-Help has three types of conditions: platform, target, and attribute. See Utilizing Conditions on page 160 for more information.

Context ID Number

See Context Sensitive Help.

Context Sensitive Help

Context Sensitive Help is Help that is displayed based on the user's location in the interface. It is accessed by clicking on a Help button or icon in a dialog box, or by pressing the **F1** button The appropriate Help topic is linked to the interface using a Context ID (or Map Number). See Implementing Context Sensitive Help on page 186.

CSS

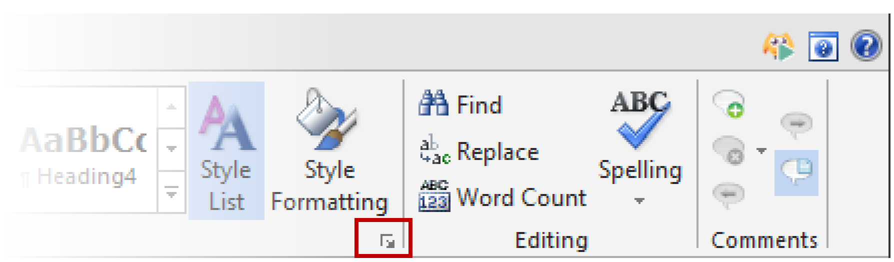
A Cascading Style Sheet. If your source documents are being authored in HTML5 or HTML, the **Source CSS** is used to apply styles to your documents. The **Target CSS** is used to transform your source documents into online Help and Manual outputs, and controls the look and behavior of those Targets. The **Source** and **Target CSS** for your project are chosen from the **Home tab** of Doc-To-Help. See HTML File Style Sheets for the list of CSS files included with Doc-To-Help. You should always start your projects with these files and edit them as you wish to customize them.

Deliverables

See Target.

Dialog Box Launchers

A dialog box launcher is the small arrow in a ribbon group that will open a dialog box.



Direct Formatting

See Local Formatting.

Eclipse Help

Eclipse Help is a Help Target that integrates with the Eclipse development environment from IBM. See Doc-To-Help Outputs and Deliverables on page 13.

EPUB

EPUB stands for "electronic publication" and is standard of the International Digital Publishing Forum (IDPF). See Doc-To-Help Outputs and Deliverables on page 13.

F1 Help

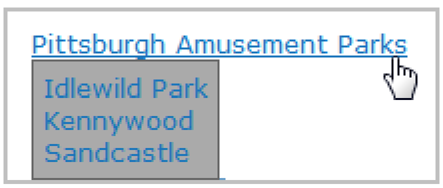
See Context Sensitive Help.

Glossary

A glossary is a list of words with their definitions, often placed at the end of a book or as the last topic in a Help Target. When you create a new project in Doc-To-Help, a Glossary document is added automatically. By default, Doc-To-Help creates links from Help topics to glossary terms automatically. See Creating a Glossary on page 299.

Groups

Groups (also referred to associative topics or ALinks) are topics that have been related to each other under a common name. After these topics are related, a link can be created in your source document that will display a popup window containing links to each topic in that Group. The name of the Group is not displayed in the list. See Creating an Index or Groups on page 327.



Help 2.0

See Microsoft Help 2.0.

HTML Help

HTML Help is a compiled HTML Target with a tri-pane format. The deliverable is a .chm ("chum") file. See Doc-To-Help Outputs and Deliverables on page 13.

Hub Project

See Modular Help System.

Index Entry

Index entries (also referred to as Keywords or KLinks) display in your online Help or manual Targets in an Index. In the final Help Target, these entries will link to (or list the page number of) the Topic(s) containing the index entry. See Creating an Index or Groups on page 327. (If you index using Microsoft Word's index features, index entries will be tagged with the XE field.)

JavaHelp

JavaHelp is a Help Target for pure Java applications. See Doc-To-Help Outputs and Deliverables on page 13.

Keyword

See Index Entry.

KLinks

See Index Entry.

Link

There are two types of Links in Help Targets; links to other topics within the Help Target, or links to website URLs. In online Help Targets, both types of links wil have the standard link style (blue, underlined text) and clicking on them will open the appropriate Topic or URL. In Manual Targets, links will have the same appearance and behavior in PDFs, but links to Topics will also include the page number of the Topic. For information on creating links to Topics in Doc-To-Help, see Creating Links on page 305. If you are authoring in Microsoft Word, you can create links to URLs using the **Insert Hyperlink** feature; if working in the Content Editor, see Creating Hyperlinks in the Content Editor.

Link Tag

A Link Tag is a unique identifier for a topic, and makes it possible to link to a topic. Occasionally 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. See Adding a Link Tag on page 311.

Live Links

When converting a Manual target (.doc) to PDF, Doc-To-Help can make links to website URLs live in the PDF. To do so, in the Help Targets dialog box on page 133, 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.

Local Formatting

Local formatting is applying formatting to a document (for example, bold, italics, font, font size, etc.) without using a style. Using local formatting makes your document less standards-compliant, harder to maintain, and may introduce inconsistencies.

Manual Formatting

See Local Formatting.

Map Number

See Context Sensitive Help.

Master Document

When you build a Manual Target, Doc-To-Help creates a master Word document that contains all of the source documents in the project, plus Front Matter, a Table of Contents, and the Index. This document is called the Master

Document, and can be found by default in the **Manual** folder of your project. It will be named "project name-master.doc(x)" by default. If you built a PDF version of the Manual, this is the document that is converted.

Microsoft Help 2.0

Microsoft Help 2.0 is a Help Target that integrates with Microsoft Visual Studio 2002-2008 only. See Doc-To-Help Outputs and Deliverables on page 13.

Microsoft Help Viewer

Microsoft Help Viewer is a Help Target that integrates with Microsoft Visual Studio 2010 only. See Doc-To-Help Outputs and Deliverables on page 13.

Modular Help System

A Modular Help System makes several Help Targets appear to be a single Target, even if one or more of the Targets is omitted from a software installation. For example, if your software application has several optional templates that are sold separately, you can create a Help project for the application, plus Help projects for each template. When the software application is installed, the Help will look like a single project, even if one of the optional templates (and its Help Target) is not installed. A Modular Help System consists of a Hub (or Parent) project and its Child projects. See Creating a Modular Help System on page 397.

NetHelp

NetHelp is a browser-based, uncompiled HTML Target. See Doc-To-Help Outputs and Deliverables on page 13.

Outputs

See Target.

Paragraph Styles

Paragraph Styles are used to assign formatting and 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. See Defining Character/Paragraph Styles and Topic Types on page 168.

Responsive Design

Responsive web designs work on both desktop and mobile platforms. They adapt automatically to device window size and are touch-enabled. Doc-To-Help includes a Responsive theme for NetHelp.

Related Topics

See Subtopic Links.

Sandcastle

Microsoft's Sandcastle utility automatically creates MSDN style reference documentation from .NET assemblies and XML comment files. See Documenting Your Class Library with Microsoft Sandcastle on page 409.

SDK Documentation

Documentation for a Software Development Kit. See Sandcastle.

Single Sourcing

Doc-To-Help is a tool for single sourcing. Single sourcing is a using one set of source documents to create several types of documentation. The most basic single sourcing project will use one set of source documents to create both an online Help and a Manual Targets. Using single sourcing techniques such as conditions and variables, you can also use one set of documents and create documentation for multiple audiences (for example: Administrator version, Manager version), and multiple deliverables (for example: user documentation, training, etc.) as well. See Introduction to Single Sourcing on page 3.

Skin

See Theme.

Source CSS

See CSS.

Source Documents

Source documents are the Word, HTML5, or HTML documents that make up your Doc-To-Help project. These are the documents that you edit, apply styles to, create links in, etc. — Doc-To-Help transforms them to online Help and Manual Targets for you. All the source documents in a Doc-To-Help project are listed in the **Documents** pane. You can open a source document by double-clicking on it. Word documents will open in Microsoft Word, HTML5 documents will open in Doc-To-Help's built-in Content Editor, and HTML documents will open in the HTML Editor installed on your machine, such as Microsoft FrontPage or Adobe Dreamweaver. See Working with Source Documents on page 263.

Source Template

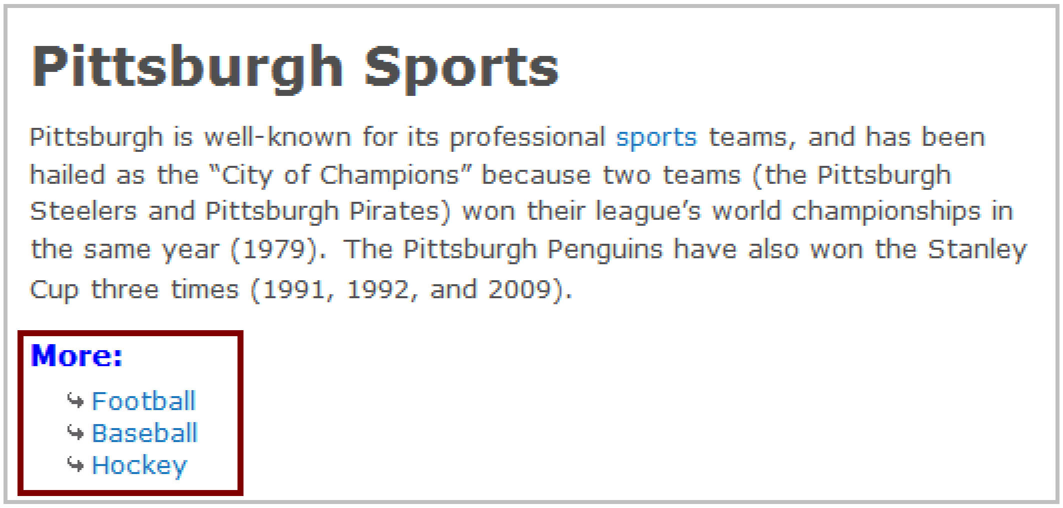
See Template.

Styles

Styles (such as Heading 1, Body Text, etc.) are contained in templates and CSSs. They should be applied to text and graphics in your source documents because they control the way that your final Targets both look and behave. Also see Paragraph Styles, and Character Styles.

Subtopic Links

Subtopic links are the "blue button" links that are displayed at the bottom of Help topics. By default, subtopic links are automatically created based on hierarchy — if authoring in Word, Heading 1 topics will display all of the Heading 2 topics under it (its "children") as subtopic links; Heading 2 topics will display the Heading 3s under it as subtopic links. If authoring in HTML5/HTML, the automatic subtopic links will be based on the Table of Contents structure. See Managing Related Topics on page 325 for more information. You may add custom related subtopic links using the Related Topics pane on page 126.



Target

Doc-To-Help can create nine different types of Targets (outputs): Help 2.0, HTML Help, JavaHelp, EPUB, Manual, Eclipse Help, Microsoft Help Viewer, NetHelp, and WinHelp. See Doc-To-Help Outputs and Deliverables on page 13 for a complete guide to targets and deliverables.

Target CSS

See CSS.

Target Template

See Template.

Template

In Microsoft Word, a template (or .dot) file is used to apply styles to your documents. In a Doc-To-Help project, a **Source Template** is used to apply styles to your documents. The **Target Template** is used to transform your source documents into online Help and Manual outputs, and controls the look and behavior of those Targets. The **Source** and **Target Templates** for your project are chosen from the **Home tab** of Doc-To-Help. (If you have existing Word files you are adding to a Doc-To-Help project, apply a Doc-To-Help **Source Template** to those files using the **Templates and Add-Ins** dialog box in Word.) See Word File Templates on page 8 for the list of template files included with Doc-To-Help. You should always start your projects with these files and edit them as you wish to customize them.

Theme

The "skin" that surrounds your online Help is called the Theme. The theme for your Help Target is chosen from the **Theme** drop-down on Doc-To-Help's **Home** tab. See Customizing Themes with the Theme Designer on page 199.

Theme Designer

See Theme.

Theme Editor

See Theme.

Topic Types

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. See Defining Character/Paragraph Styles and Topic Types on page 168.

Variable

Variables are used to manage content in one place for reuse across your project. Doc-To-Help has two different types of Variables — text or rich content. 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 document. When you build your Target, the variable content is automatically inserted. If you edit a variable later, those changes will updated throughout the project the next time you build the Target. See Creating Variables for more information on creating and using variables.

WinHelp

WinHelp is a compiled Help Target. It has two deliverables; an .hlp and .cnt file. See Doc-To-Help Outputs and Deliverables on page 13.

XE field

See Index Entry.

Index