



MADCAP DOC-TO-HELP 6

Project Management

Copyright 2019 MadCap Software. All rights reserved.

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

MadCap Software
9191 Towne Center Drive, Suite 150
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Introduction 5

CHAPTER 2

Migrating a Project 6

 Transfer the Project 7

 Transfer the Custom Templates and Themes 8

 Get Started 9

CHAPTER 3

Modular Projects 10

 Common Scenarios 11

 Modular Output Deliverables 12

 Standardizing Modular Output 13

 File Organization 14

 Creating a Hub Project 15

 Creating a Child Project 19

 Creating Modular Output for Microsoft Help Viewer Targets 26

CHAPTER 4

Project Properties and Settings 30

Compacting a Project File	31
Importing and Exporting Project Settings	32
Setting Project Properties	35
 CHAPTER 5	
Reports	39
 CHAPTER 6	
SharePoint Integration	41
SharePoint Library	43
Publishing to SharePoint	53
 CHAPTER 7	
Team Authoring	57
How It Works	58
Team Authoring Requirements	59
Steps Associated With Team Authoring	60
Team Authoring Environment	61
Setting Up a Team Project	62
Connecting to a Team Project	68
Working in a Team Project	70
Team Authoring Administrative Functions	82
 APPENDIX	
PDFs	91

Introduction

There are several features in Doc-To-Help that can help you better manage your projects, especially if you have a team of content developers. This includes setting up modular projects, providing various project settings, viewing reports, implementing team authoring, and more.

Here are the primary ways to manage your project:

- **Migrating Projects** 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 the entire project folder, but here are a few tips to make the transition easier. See "Migrating a Project" on page 6.
- **Modular Projects** A modular project is necessary when you have a collection of several different output files (and would like to keep them that way) but would like them to appear to the end user as a single output system. See "Modular Projects" on page 10.
- **Project Properties and Settings** You can provide your project with various settings. See "Project Properties and Settings" on page 30.
- **Reports** Doc-To-Help has several 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. See "Reports" on page 39.
- **SharePoint** Doc-To-Help's integration with Microsoft SharePoint makes it easy to collaborate with your team, as well as across your company. See "SharePoint Integration" on page 41.
- **Team Authoring** Team authoring is a source control feature that makes it possible for multiple authors to work together on a single project. See "Team Authoring" on page 57.

Migrating a Project

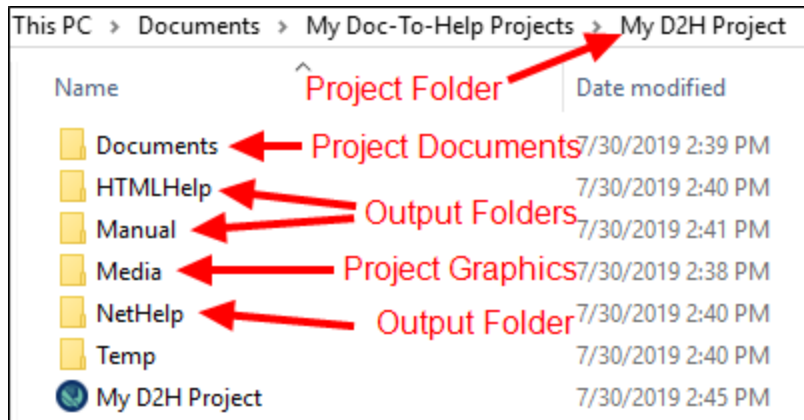
Many of us collaborate on Doc-To-Help projects using Team Authoring, TFS, or SharePoint, but sometimes we simply need to transfer a project to a coworker or a client. This is done by sending the entire project folder, but here are a few tips to make the transition easier.

This chapter discusses the following:

- Transfer the Project 7
- Transfer the Custom Templates and Themes 8
- Get Started 9

Transfer the Project

Here is a typical project folder:



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

C:\Users\[user name]\Documents\My Doc-To-Help Projects

So when you give colleagues 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 > Common Tasks > Options**.)

✓ **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 .dotx):

Windows® 7 and later at: \\Users\<user name>\AppData\Roaming\Microsoft\Templates

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

C:\Users\[user name]\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 to your colleague. Then your colleague can drop that folder into the My Doc-To-Help Projects folder (C:\Users\[user name]\Documents\My Doc-To-Help Projects).

Get Started

Once transferred, open the project by choosing **File > Doc-To-Help > Open Project**. The file extension of all Doc-To-Help projects is .d2h.

Double-click on any document in the Documents window pane to begin editing.

Modular Projects

A modular project is necessary when you have a collection of several different output files (and would like to keep them that way) but would like them to appear to the end user as a single output system.

This chapter discusses the following:

- Common Scenarios 11
- Modular Output Deliverables 12
- Standardizing Modular Output 13
- File Organization 14
- Creating a Hub Project 15
- Creating a Child Project 19
- Creating Modular Output for Microsoft Help Viewer Targets 26

Common Scenarios

There are four common scenarios necessitating a modular project:

- **Modular Software** Your software application is sold as separate modules with the user purchasing one or more modules at a time. Modular output will contain all the output files, but only the appropriate output files will accompany the purchased modules.
- **Large Documentation Sets** If you wish, you can chunk information into several small online outputs and create a modular project rather than deploy one large online output system. Smaller projects enable streamlined updating and easier distribution.
- **Online Outputs with Future Projects Planned** If you plan to release your online output in stages, you can pre-position placeholders for future online projects before they are released. Instead of distributing the entire online output collection each time you add to the system, you only need to distribute additional online output files. If you didn't plan for an addition, you can distribute a new hub output file along with the new output files.
- **Documentation Teams Sharing Responsibilities Across One Large Project** Modular output can be a solution for a large project with many authors. Each author is assigned one or more projects. The team leader is usually charged with the responsibility of maintaining the hub project and assembling the entire project.

You may want to check out Doc-To-Help's Team Authoring capabilities and use those instead. See "Team Authoring" on page 57.

Modular Output Deliverables

Modular outputs are composed of one hub (or master) project and any number of child projects. The modular system appears to the user as one integrated collection of online output. The contents and index are merged at run time, when the output 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 project. See "File Organization" on page 14.

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 output. (This file name can be renamed in the "Default name" field of the Help Targets dialog.)

ECLIPSE HELP

- The entire contents of the hub project EclipseHelp folder.

MICROSOFT HELP VIEWER

- See "Creating Modular Output for Microsoft Help Viewer Targets" on page 26.



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 output. 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. See the online Help or the *Targets Guide*.

Standardizing Modular Output

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
- 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).

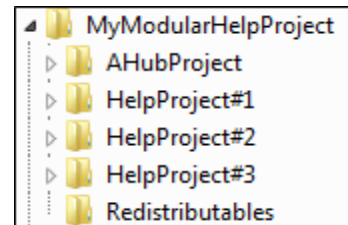
File Organization

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.



NOTE: Modular projects can reference files that are not installed (e.g., 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 output will be added to the 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 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.

Creating a Hub Project

In modular output, 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 (e.g., welcome message, company contact information) if you wish.

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 output and child output to the end user.



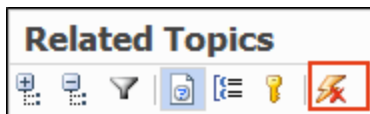
NOTE: Before you create a hub project, be sure you have created and built the output for your child projects. You should also set up your file system for the entire modular system. See "Modular Projects" on page 10.

HOW TO SET UP A HUB PROJECT

1. Create your hub project. The project name should contain no spaces.
2. Select the **Project** ribbon.
3. Click the **Project Settings** button on the **Project** ribbon. The Project Settings dialog opens.
4. In the **Output options** section, select **Is modular hub project**.
5. Click **OK** to close the Project Settings dialog.
6. 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 choose only one. (The placeholder topics will later be mapped to the child topics. This is done by editing the topic properties.)

Here are the document structure options:

- Heading 1 placeholder topics only
 - One Heading 1, followed by placeholder topics (Heading 2 paragraphs)
 - One Heading 1, followed by content in Body Text, then the placeholder topics (Heading 2 paragraphs)
 - One Heading 1, followed by one Heading 2 and Body Text content, then the placeholder topics (Heading 2 paragraphs)
7. Click the **Related Topics** view in the Doc-To-Help Project panel.
 8. At the top of the **Related Topics** view, click the **Disable Subtopic Relations** button. This disables any automatic related links that may be displayed in the hub project.



9. Build the output for the hub project.
10. Add at least two index entries (one must be a keyword) to the project.
11. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
12. On the left, choose the target (can be HTML Help, NetHelp, or Eclipse Help). The hub project, and all of the child projects, must use the same target.

13. If you are using HTML Help, in the **Advanced** section, clear the **Binary index** and **Binary table of contents** check boxes.
14. Build the output.

HOW TO MAP PLACEHOLDER TOPICS TO CHILD TOPICS

1. Open your hub project.
2. In the Topics Panel window, right-click a placeholder topic and choose **Properties**. The Topic Properties dialog opens.
3. Edit the topic properties, 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 files can be found in the child project's HTMLHelp folder.)
 - **For NetHelp 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.
4. Repeat these steps to map all additional placeholder topics.
5. Open the **Target** ribbon and click the **Rebuild** button. When it is finished, click **View** to see the output.

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 projects, you will need to place copies of the child project files (CHM for HTML Help) in the same folder as the hub project files to view the complete modular project. Whenever you do a full rebuild of the target (as opposed to simply building the 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 output must be created with the same version of Doc-To-Help.

Creating a Child Project

Child projects are tied together with the hub project (see "Creating a Hub Project" on page 15).

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 "Modular Projects" on page 10.


HOW TO SET UP A CHILD PROJECT


1. Create your child project. The project name should contain no spaces.
2. Build the output for the child project. This will generate the project's topics and allow you to create cross-project links.
3. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
4. On the left, choose the target (can be HTML Help, NetHelp, or Eclipse Help).

 **NOTE:** The hub project, and all of the child projects, must use the same target.

5. If you are using HTML Help, in the **Advanced** section, clear the **Binary index** and **Binary table of contents** check boxes.
6. Build the output.

 **NOTE:** Cross-project links may be created between child projects. See "Cross-Project Linking" on page 21.

 **NOTE:** Child projects using automatic context ID creation need to have offsets added. See "Modifying Context IDs in Modular Projects" on page 22.

 **NOTE:** Glossaries may be used in Modular hub projects, but HTML Help targets require a few adjustments to avoid duplicate items. See "Modular Output Glossaries" on page 23.



NOTE: One issue with modular HTML Help is that the child projects do not contain the entire table of contents (TOC). The Modular TOC Utility removes the TOC limitation by automatically adding the full TOC to each Help file in the modular project. See "Modular TOC Utility" on page 24.

Cross-Project Linking

It is possible to create links from one set of output files to another in modular output. Regular topic links should be used for links within each set of output files.

HOW TO CREATE A CROSS-MODULE LINK

The following shows how to create a link from one project called "HelpProject#2" to another called "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.
3. Build the target, then 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.
7. Build the target, then close the project.
8. Open the hub project.
9. Build the target.



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.

HOW TO CUSTOMIZE CONTEXT IDS

1. Open a child project.
2. Select the **Project** ribbon.
3. Click the **Project Settings** button. The Project Settings dialog opens.
4. In the **Context IDs** section, 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.
5. Build the target, then close the project.
6. Open another child project.
7. Open the Project Settings dialog, and enter a different number in the ID Offset field (e.g., 2000).
8. Build the target, and close the project.
9. Repeat these steps for all child projects.

Modular Output Glossaries

Glossaries may be used in modular hub projects, but HTML Help 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.

HOW TO WORK WITH GLOSSARIES IN HTML HELP FILES

1. Verify all child projects, and the hub project, contain the same Glossary document file.
2. Open a child project.
3. Click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
4. In the **Advanced** section, select the **Skip glossary** check box.
5. Build the output.
6. Repeat this 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 (TOC). 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 TOC limitation by automatically adding the full TOC 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 TOC is incorporated into each modular Help file.

HOW 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 opens.
2. Select **Create a new Modular TOC file** and click **Next**.
3. After reading the dialog 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 "Modular Projects" on page 10.)



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.)
7. The Modular TOC Utility lists all the child .hhp files in your modular Help system. Choose the first child .hhp file, then click **Next**.
8. 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.)
9. Repeat until each of the child .hhp files has been located.
10. Click **Next**. The Modular TOC Utility prompts you to verify that the path for the build project.

11. Click **Next**.
12. Browse to select a directory to store your new modular Help files.
13. Click **Next**.
14. Verify the .hub file and build paths.
15. Click **Finish** to build the new modular Help files.
16. After the files are built, the Modular TOC Utility Editor opens.
17. 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 TOC.

Creating Modular Output for Microsoft Help Viewer Targets

The Microsoft Help Viewer (MHV) target is exclusively for integration with the Microsoft Visual Studio 2010 (and above) output.

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) output.

☆ EXAMPLE

Let's say your 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, you create an MHV modular project. Not only is a merged TOC created, but the multiple MHV 1.x setups (HelpContentSetup.msha files) are merged into a single setup.

HOW TO CREATE AN MHV MODULAR PROJECT

1. Set up your modular file structure in Windows Explorer (see "Modular Projects" on page 10). You will not need the Redistributables folder.
2. 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.
3. Create a new project that will serve as the hub or parent project. The project name should contain no spaces.
4. Select the **Project** ribbon.
5. Click the **Project Settings** button. The Project Settings dialog opens.
6. In the **Output options** section, select **Is modular hub project**.
7. Click **OK** to close the Project Settings dialog.
8. Open the hub project's source document and create placeholder topics for each of the child topics you wish to associate with the hub.
9. Build the Microsoft Help Viewer target.

10. Click the **Topics Panel** button on the **Project** ribbon. Right-click on each of the topics, and select **Properties** from the menu. In the Topic Properties dialog, 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 the product name listed in the Help Targets dialog 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 for your project to be inherited by all the children, select the **Inherit Book name** check box.
11. Rebuild the Microsoft Help Viewer target.



NOTE: 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.



NOTE: All files in the MHV target folders (MSHelpViewer by default) are required to create a successful setup.

LIMITATIONS

Following are limitations of this kind of output:

- 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.

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>MadCap Software</title>
  </head>
  <body class="manifest-list">
    <div class="manifest">
      <a class="manifest-link" href-
f="C1Barcode/HelpContentSetup.msha">MadCap BarCode for
.NET Help</a>
    </div>
    <div class="manifest">
      <a class="manifest-link" href-
f="C1Zip/HelpContentSetup.msha">MadCap 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).
- `MadCap BarCode for .NET Help` Required (only one in a manifest div). Its text is optional, not used.

MHV RESOURCES

- MSDN Forum <http://social.msdn.microsoft.com/Forums/en-US/devdocs/threads>

CHAPTER 4

Project Properties and Settings

You can work with project properties and settings in multiple ways.

This chapter discusses the following:

- Compacting a Project File 31
- Importing and Exporting Project Settings 32
- Setting Project Properties 35

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.

HOW TO COMPACT A DOC-TO-HELP PROJECT

1. Make sure you close all Doc-To-Help projects before using the compact utility.
2. Select **File > Doc-To-Help > Common Tasks > Compact**. The Open Doc-To-Help Project dialog opens.
3. Choose the project (.d2h file) and click **Open**. The project is compacted. A message box informs 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 Project Settings Import or Export Wizard. This can save significant time if you'd like to use the same settings for multiple projects.

PROJECT OBJECTS


Project objects can be copied from another Doc-To-Help project, or from an XML file. Following are the objects that are available for import/export:

- Keywords
- Groups
- Variables
- Attributes
- Windows
- Styles
- Scripts
- Templates
- Style Sheets
- Help Targets
- Project Properties

HOW TO IMPORT PROJECT SETTINGS

1. Choose **File > Doc-To-Help > Common Tasks > Import Project Settings**. The Project Settings Import Wizard opens.
2. Choose to import from an **XML** file or an existing **Doc-To-Help** project.
3. Click **Next**.
4. Click , browse to the XML file or Doc-To-Help project file (.d2h) location, select it, and click **Open**.
5. Click **Next**.
6. Select the project objects you'd like to import. By default, all are selected.
7. (Optional) If you'd like to overwrite the existing settings in the current project (rather than add to them), select **Overwrite objects existing in the current project**.
8. Click the **Import** button. Any issues with the import will be displayed.

HOW TO EXPORT PROJECT SETTINGS

1. Choose **File > Doc-To-Help > Common Tasks > Export Project Settings**. The Project Settings Export Wizard opens.
2. Choose to export from an **XML** file or an existing **Doc-To-Help** project.
3. Click **Next**.
4. Click , browse to the XML file or Doc-To-Help project file (.d2h) location, select it, and click **Open**.
5. Click **Next**.
6. Select the project objects you'd like to export. By default, all are selected.
7. (Optional) If you'd like to overwrite the existing settings in the current project (rather than add to them), select **Overwrite objects existing in the current project**.
8. 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 (x86)\MadCap Software\MadCap DocToHelp 6\ directory. Storing project settings in an XML file makes it possible to automatically modify Doc-To-Help projects.



NOTE: You can also import settings from another Doc-To-Help project using the New Project Wizard.

Setting Project Properties

The Project Settings dialog is used to set the universal properties for the entire project, regardless of target.

HOW TO SET PROJECT PROPERTIES

1. Select the **Project** ribbon.
2. Click the **Project Settings** button. The Project Settings dialog opens.
3. Set any of the following:

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 target and window properties. For the HTML Help window, Help 2.0, JavaHelp, and NetHelp, it is the default caption. For Manual, it is the default title. For the MSHelp Viewer, it is the default product vendor, product name, and product book.

DOCUMENTS

- **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 source documents will reside. The default folder name is Documents. It is strongly recommended that all source documents be stored inside this folder, because it helps to keep the project files organized and will ensure that links to files of any kind are preserved.

OUTPUT OPTIONS

- **Keep outline numbers** When selected, outline numbers are included as part of topic titles and are present in 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 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 output. 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 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 (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 (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 file can dynamically load the contents of other output files, if present. Only HTML Help, and NetHelp platforms can support modular hub projects. For HTML Help targets, to specify a component output 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) to the component **filenames**. When testing your project, you will need to copy the component files into the output folder of each modular target. For NetHelp targets, to specify a component output 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.
- **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 (e.g., 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 ribbon (PlugIns 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.
- **Update customized table of contents in build** If selected, any customized tables of contents (TOCs) 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.

4. Click **OK**.

Choosing a Source Folder

You can set the default folder that your source documents are stored in. This can be done for Word and media files.


DEFAULT FOLDER FOR WORD SOURCE FILES

The default folder for Word source files is Documents. This folder is automatically created when you create a project. It is strongly recommended that all source documents be stored inside this folder, because it helps to keep the project files organized and will ensure that links to files of any kind are preserved.

DEFAULT FOLDER FOR MEDIA FILES


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 be stored inside this folder.

HOW TO ADD A SOURCE DOCUMENT FOLDER

1. Select the **Project** ribbon.
2. Click the **Project Settings** button. The Project Settings dialog opens.
3. Depending on the type of files, click  next to the **Document folder(s)**, **Plugin folder(s)**, or **Media folder(s)** field. The Choose Source Folder dialog opens.
4. Click **Add Folder**. A Browse dialog opens.
5. Choose or create a folder and click **OK**.

HOW TO DELETE A FOLDER

To delete a folder, choose the folder in the window and click Remove Folder.

1. Select the **Project** ribbon.
2. Click the **Project Settings** button. The Project Settings dialog opens.
3. Depending on the type of files, click  next to the **Document folder(s)**, **Plugin folder(s)**, or **Media folder(s)** field. The Choose Source Folder dialog opens.
4. Select the folder you want to delete and click **Remove Folder**.
5. Click **OK**.

Reports

Doc-To-Help has several 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.

Following are the available reports:

- **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** Displays a list of associated topics, if any. Groups are listed in alphabetical order.
- **Index Report by Keyword** Shows a list of associated topics, if any. Keywords are listed by full path name in alphabetical order.
- **Index Report by Topic** Shows a list of associated keywords and groups, if any. Topics are listed in alphabetical order by title.
- **Script Listings** Lists the code for each script defined in the project. Scripts are listed in alphabetical order.
- **Style Definitions** Lists the properties of each style defined in the project. Styles are listed in alphabetical order.
- **Topic Detail 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** 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.

- **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 path name 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.

HOW TO RUN A REPORT

1. Select **File > Doc-To-Help**.
2. Click **Reports**, then choose the report name. The Reports window opens.
3. 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.


SharePoint Integration


Doc-To-Help’s integration with Microsoft SharePoint makes it easy to collaborate with your team, as well as across your company.

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 "SharePoint Library" on page 43.

This chapter discusses the following:

SharePoint Library	43
Publishing to SharePoint	53

 **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 53.

 **NOTE:** SharePoint collaboration is compatible with Microsoft® SharePoint® 2007 and above, as well as Office 365 (SharePoint Online). SharePoint 2010 or 2013 is required when publishing Doc-To-Help targets to a SharePoint Wiki Library.



NOTE: Doc-To-Help's SharePoint integration works with the free versions of SharePoint that install with some versions of Microsoft Office.



NOTE: The Options dialog has a tab named SharePoint that you can use to set your preferences. To open this dialog in Doc-To-Help, select **File > Doc-To-Help > Common Tasks > Options**.



NOTE: If you open a document from the SharePoint server (not the local copy), the Word will not automatically find the Doc-To-Help project. You must open the document separately for Doc-To-Help to find the project.

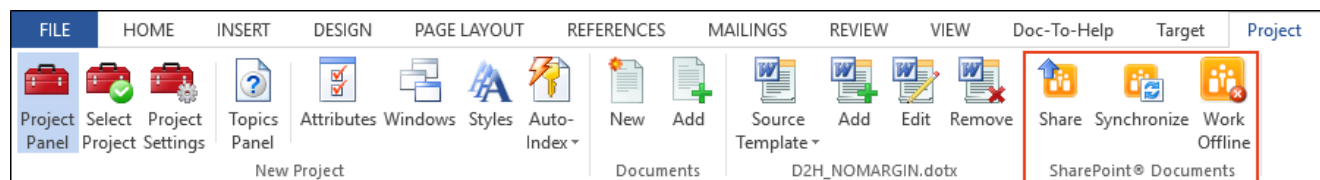
SharePoint Library

Uploading your content to a SharePoint Library lets you 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. You 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.




TASKS ASSOCIATED WITH THE SHAREPOINT LIBRARY


Following are the primary tasks for using the SharePoint Library:


- **Add SharePoint Documents to 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. See "Adding SharePoint Library Documents to a Doc-To-Help Project" on page 45.
- **Check Documents In and Out** You can check documents in and out of the SharePoint Library to work on them. See "Checking Documents In and Out of the SharePoint Library" on page 46.
- **Merge Documents** 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 will give you the option to overwrite one of the documents, or merge the two documents. See "Merging Word Documents" on page 48.
- **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. See "Stop Sharing a Document " on page 49.

- **Synchronize Documents** Doc-To-Help automatically synchronizes the documents changed in the SharePoint Library with those on your machine. However, you can also synchronize documents manually. See "Synchronizing Documents " on page 49.
- **Upload Project Documents to SharePoint** You can upload Doc-To-Help documents to a SharePoint Library. See "Uploading Documents to a SharePoint Library" on page 50.
- **Work Offline** You can turn off the synchronization with the SharePoint Library server and work offline. See "Working Offline " on page 52.

 **NOTE:** 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.

 **NOTE:** You cannot share D2H files.

 **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 Word.

You will be prompted to enter your new password the next time you try to connect to SharePoint.

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.

HOW TO ADD SHAREPOINT LIBRARY DOCUMENTS TO DOC-TO-HELP


1. In the Doc-To-Help Project panel, open the Documents window pane and click the **Add Existing Documents** button . The Document Import Wizard opens.
2. Choose the **SharePoint** option and click **Next**.
3. Enter the document's **SharePoint URL**.
4. If you wish to import the document(s) without retaining their connection to SharePoint, clear the **Retain SharePoint Connection** check box and click **Next**.
5. 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**.
6. Confirm your choices and click **Next**. The document(s) are imported.
7. Click **Finish**.


Use the arrow buttons in the toolbar of the Documents pane to rearrange the document order.

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 and begin editing. If you would like to change this default, open the Options dialog (**File > Doc-To-Help>Common Tasks>Options**), select **SharePoint**, and clear the **Check out Word documents when editing local copy** check box.



HOW TO CHECK DOCUMENTS IN AND OUT OF THE SHAREPOINT LIBRARY

1. In Word, open the Documents pane, and select the document you would like to check out.
2. Right-click and from the menu, choose **SharePoint > Check Out**. The document icon indicates that the document is checked out  **OutOfTheBox.doc**.

In SharePoint, the document is also flagged with an icon  **OutOfTheBox**

You can open the SharePoint Library by right-clicking on the document and choosing **SharePoint > Open in Browser**.

3. After making edits, save your document and close it. Then right-click on it and choose **SharePoint > Check In**.

 **NOTE:** If a document is checked out in SharePoint by another user, there will be a red checked out icon next to it  **SoftwareDocumentation.doc**

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 will open and list the conflicting documents. You can then resolve the issue before you build. Doc-To-Help will give you the option to merge the documents. See "Merging Word Documents" on page 48.

UNDOING A CHECK OUT

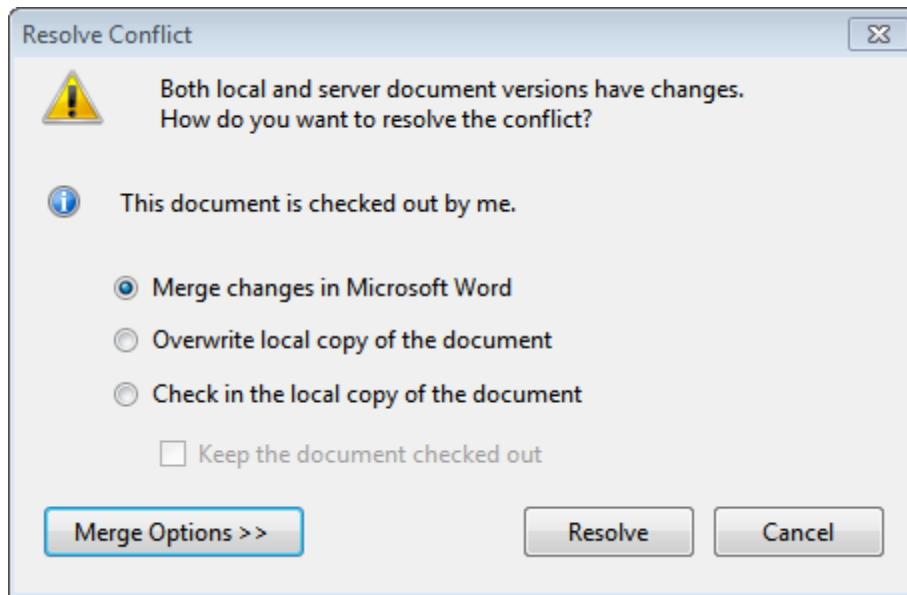
If you check out a document and make changes to it, then decide you do not want to retain the changes, right click on it in the Documents pane and choose **SharePoint > Discard Check Out**.

Merging Word Documents

A conflict will arise if both versions of a document (the SharePoint Library version and the local version on your machine) are different, and you try to save, open, close, or synchronize a single Word document. A dialog will give you the option to overwrite one of the documents, or merge the two documents.

HOW TO MERGE DOCUMENTS

1. In the Resolve Conflict dialog choose **Merge changes in Microsoft Word**.



2. After you click the **Resolve** button, Word displays the changes to the document.
3. 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, Word will suggest that you synchronize the changes. If you merged to the local copy, 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 will open and list the conflicting documents. Right-click on the document name in the Synchronize Documents dialog to view the options.

Stop Sharing a Document

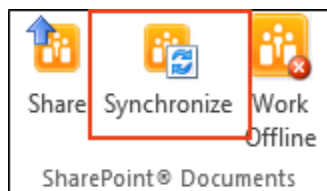
If you no longer wish to share a document with the SharePoint Library, you can stop sharing using the right-click menu. In the Documents pane, right-click the document and choose **SharePoint > Stop Sharing**. The document will no longer be shared with the SharePoint Library and its icon will change accordingly.

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, SharePoint Documents tab.)



You can synchronize manually at any time by opening the **Project** ribbon in Doc-To-Help and clicking the **Synchronize** button.

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 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. See "Merging Word Documents" on the previous page.

Uploading Documents to a SharePoint Library

You can upload Doc-To-Help documents to a SharePoint Library.

HOW TO UPLOAD DOCUMENTS TO A SHAREPOINT LIBRARY

1. Open the Doc-To-Help project that contains the documents you want to share.
2. Select the **Project** ribbon, and in the **SharePoint Documents** group click the **Share** button. The Document Share Wizard opens.
3. Click **Next**.
4. Choose the SharePoint Library.
5. In the **Share Documents** to field, enter the SharePoint site URL. 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.
6. (Optional) 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.
7. (Optional) If you wish to upload the documents, but not have them retain any connection to SharePoint, clear the **Retain SharePoint Connection** check box.
8. Click **Next**.
9. Choose the documents you would like to share. If you select the **Documents** folder check box, a subfolder named "Documents" will be created.
10. Click **Next**.
11. Confirm your choices and click **Next**. The documents are uploaded.
12. Click **Finish**.

If there were graphics or other auxiliary files in the project, the Synchronize Documents dialog 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.

In the Documents pane, the shared documents display a special document icon .

HOW TO UPLOAD AN INDIVIDUAL DOCUMENT

- In the Documents window pane in Word, right-click on the document and choose **Share Document to SharePoint Library**.



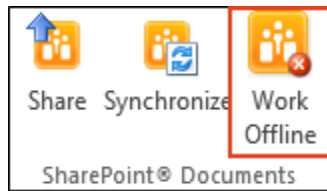
NOTE: If you change your SharePoint password after uploading documents, you will no longer be able to synchronize your documents. You can fix this by clearing your Windows credentials.

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 Word.

You will be prompted to enter your new password the next time you try to connect to SharePoint.

Working Offline

To turn off the synchronization with the SharePoint Library server, open the **Project** ribbon and click the **Work Offline** button. 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.

Publishing to SharePoint

After generating NetHelp output, you can publish it to SharePoint.

SHAREPOINT PUBLISH OPTIONS

Doc-To-Help provides two ways you can publish your output to SharePoint.

- **Option 1** One option is to publish your NetHelp to SharePoint, where anyone with permissions can access it. There is no need for a special setup to post your output 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 output files are published to SharePoint and accessible from index.html (the default name for the NetHelp home page).
- **Option 2** 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 output 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 output 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 Word source documents and graphics to a SharePoint Document Library. These options are useful for content collaboration, source control, versioning, and more. See "Merging Word Documents" on page 48.



NOTE: Publishing to a SharePoint Document Library is compatible with Microsoft® SharePoint® 2007 and above, as well as Office 365 (SharePoint Online); publishing to a SharePoint Wiki Library is compatible with SharePoint 2010 and above, as well as Office 365.

BEFORE YOU PUBLISH

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**.



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 Word.

You will be prompted to enter your new password the next time you try to connect to SharePoint

HOW TO PUBLISH DOCUMENTS TO A SHAREPOINT DOCUMENT LIBRARY OR WIKI PAGE LIBRARY

1. In the Doc-To-Help project, choose the **Target** ribbon, click the **Active Target** button, and choose a target.
2. Click the **Build** button.
3. Click the **Publish to SharePoint** button. The Publish to SharePoint Wizard opens.
4. Click **Next**.
5. Choose either **Upload to a SharePoint Document Library** or **Upload to a SharePoint Wiki Library**.
6. Choose the SharePoint Library to which you want to upload.
7. In the **Upload Files to** field, enter the SharePoint site URL. 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.
8. 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.
9. Confirm your choices and click **Next**. The documents will be uploaded.
10. 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 (TOC) will be displayed. Click on the TOC links to navigate the project.

When you click the **View** button from the Wizard, the NetHelp output uploaded to a SharePoint Document Library will open in the browser.

However, the output files/folders are in SharePoint. The index.html file (the default home page for NetHelp projects) opens the output in the browser. If you want to store your output in SharePoint and provide access to it, point users to this file.

For more information on working in a SharePoint Wiki Library, see "Editing a SharePoint Wiki" on the next page.

Editing a SharePoint Wiki

Once your NetHelp is in a SharePoint wiki, it can be edited by anyone with the proper permissions.

HOW TO EDIT A SHAREPOINT WIKI

1. Click the **Page** tab.
2. Click the **Check Out** button.
3. Click the **Edit** button. The page switches to edit mode and editing can begin.

The **Format Text** and **Insert** ribbons display 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: 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.

4. Make your edits and save your work.
5. On the **Format Text** tab, click the **Check In** button.

Team Authoring

Team authoring is a source control feature that makes it possible for multiple authors to work together on a single project. 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. Although the same project is made available to the entire team, one author's changes will not be overwritten by another author's changes.

With team authoring, your projects can be stored in the following kinds of repositories:

- Microsoft® Team Foundation Server
- Web Repository
- File System Repository

If you would prefer to store your documents in Microsoft® SharePoint®, see "Merging Word Documents" on page 48.

This chapter discusses the following:

How It Works	58
Team Authoring Requirements	59
Steps Associated With Team Authoring	60
Team Authoring Environment	61
Setting Up a Team Project	62
Connecting to a Team Project	68
Working in a Team Project	70
Team Authoring Administrative Functions	82

How It Works

Team members work on their own local copy of the Doc-To-Help project. 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 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 **About** button on the **Doc-To-Help** ribbon.
- If using a Team Foundation Server repository, all team members must install Microsoft Visual Studio Team Explorer 2008 or later.

Steps Associated With Team Authoring

Following are the primary steps associated with team authoring:

1. **Set Up Project** Each team authoring project must have an administrator (usually the documentation manager or lead writer). The administrator makes the project available to other team members by uploading it to the central repository and configuring the settings. See "Setting Up a Team Project" on page 62.

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 82.

2. **Connect to Project** After a team authoring project is set up, team members must connect to and download the team project in order to create their own working copies of the project. Once team members have a working copy, they can check files out, edit them, and check them back in. See "Connecting to a Team Project" on page 68.
3. **Work in Team Project** After connecting to a team project, team members can work in their own working copy of the project. See "Working in a Team Project" on page 70.

Additional team authoring options can be set in the Options dialog (**File > Doc-To-Help > Common Tasks > Options > Team**).

Team Authoring Environment

When you open the working copy of a team project, Doc-To-Help will include an additional ribbon 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 content variable documents will be included in these folders also.
- **Auxiliary Files** These are any files that are not documents. Examples include image 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, and project properties.

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.

You can manage components of your Doc-To-Help project in the following ways if you intend to bind it with a shared repository:

Once your Team Authoring project is open, you can access the Team Authoring panel for more functions to help manage this project.



NOTE: Doc-To-Help integrates directly with Team Foundation Server and Microsoft SharePoint. It does not directly integrate with third-party tools such as ClearCase. However, Doc-To-Help will not interfere with those tools, so you can use them with your Doc-To-Help projects.

Setting Up a Team Project

Each team authoring project must have an administrator (usually the documentation manager or lead writer). The administrator makes the project available to other team members by uploading it to the central repository and configuring the settings. For more information on team authoring, including requirements, see "Team Authoring" on page 57.

REPOSITORY TYPES

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 lets you 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.

HOW TO SET UP IIS FOR WEBDAV USING IIS MANAGER 6.5 OR HIGHER

1. Open IIS Manager and in the pane on the left, select **Default Web Site** (or the website you want to set up).
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**.



NOTE: If WebDAV is not already enabled on your computer, you will need to enable it before you can set up IIS. To do this, do the following:




1. Open the Windows Control Panel.
2. Click **Programs and Features > Turn Windows features on or off**. The Windows Features dialog opens.
3. Expand **Internet Information Services > World Wide Web Services > Common HTTP Features**.
4. Select **WebDAV Publishing**.
5. Click **OK**.



NOTE: If you are using an older version of IIS, check with your network administrator to verify the settings needed to use WebDAV.

HOW TO SHARE A PROJECT FOR TEAM AUTHORING

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.

1. Create a new Doc-To-Help project or open an existing one.
2. Select **File > Doc-To-Help > Team Authoring > Share Project for Team Authoring**. The Share Project Wizard opens.
3. Click **Next**.
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 "How to Set Up a Virtual Directory on Your Windows Machine" on page 66.
5. Click **Next**.
 - **For Team Foundation Server** Enter the Team Project Location URL. You can click  to browse to a different server.
 - **For File System Repository** Enter a network location or click  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  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. If you selected 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**.

8. In the Repository Options window, set the backup options:

- **Limit the number of backups for each file** This is 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 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 84.

9. Click **Next**.

10. Click **Yes** to confirm your repository information and click **Next** to create the team project. The Upload Project dialog opens.

11. Once the upload process is complete, click **Close**. At this point, your project has been uploaded to a 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**.

HOW TO SET UP A VIRTUAL DIRECTORY ON YOUR WINDOWS MACHINE

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.

The following steps may be slightly different depending on your operating system.

1. 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 **Add Virtual Directory**. The Add Virtual Directory dialog opens.
7. In the **Alias** text box, enter a name for your virtual directory. This name will be part of the URL for your web repository.
8. Next to the **Physical path** field, click . In the dialog that opens, browse to the physical directory on your machine that contains the content you want to share and click **OK**.
9. In the Add Virtual Directory dialog, click **OK**.
10. In the **Connections** pane, select your new directory.
11. In the **Features View** pane, double-click **WebDAV Authoring Rules**. The WebDAV Authoring Rules dialog opens.
12. In the **Actions** pane, click **Add Authoring Rule**. The Add Authoring Rule dialog opens.
13. In the **Allow access to this content to** section, select **All users**.
14. In the **Permissions** section, select **Read**, **Source**, and **Write**.
15. Click **OK**. The virtual directory is now set up. You can view the rules and permissions for the directory in the Features View pane. Make sure the directory has Read, Source, and Write privileges.





NOTE: 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

After a team authoring project is set up (see "Setting Up a Team Project" on page 62), team members must connect to and download the team project in order to create their own working copies of the project. Once team members have 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.

HOW TO CONNECT TO AND DOWNLOAD A TEAM PROJECT

1. In Word, select **File > Doc-To-Help > Team Authoring > Connect to Team Project**.
2. Under **Select repository type**, choose one of the following:
 - **Team Foundation Server**
 - **File System Repository**
 - **Web Repository**
3. Click **Next**.
 - **For Team Foundation Server** Enter the team project location URL. You can click  to browse to a different server.
 - **For File System Repository** Enter a network location or click  to browse to a folder on the network.
 - **For Web Repository** Enter the URL of the virtual directory. (If necessary, enter your login credentials.)
4. Click **Next** and specify the working project folder. This is where the project will be stored on your machine and is your working copy of the project.
5. Click **Next** and confirm the working copy information.
6. Click **Next** to create the working copy.
7. Click **Close** and then click **Finish**. Your working copy of the project opens.

DISCONNECTING FROM A TEAM PROJECT

1. In Word, select **File > Doc-To-Help > Team Authoring > Disconnect Project from Team Authoring**.
2. A popup window will ask you to confirm if you wish to proceed with disconnecting from the project. Click **Yes**.

After you have connected to a team project, creating a working copy of your project, 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 the next page.

Working in a Team Project

After connecting to a team project, each team member can work in his or her own working copy of the project.

HOW 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 68), the project will automatically open.
- To reopen it, choose **File > Doc-To-Help > Open Project** and open it just as you would any other project.

TASKS

When you work in a team project, you can perform any of the following tasks. Options for working with files are available from the Team ribbon in Doc-To-Help, as well as by right-clicking on a file in the Team Authoring window (see "Team Authoring" on page 57).

- **Add File** 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. See "Adding a New File to the Team Project" on page 72.
- **Check Out 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. See "Checking Out the Project File" on page 73.
- **Compare Documents** In TFS repositories, you can compare two versions of a document. See "Comparing Documents" on page 75.
- **Delete File** 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. See "Deleting a File From the Team Project" on page 76.
- **Edit Document** To edit a document for a team authoring project, 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.) See "Editing a Document" on page 77.
- **Get Latest Version** 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. See "Getting the Latest Version of a File " on page 78.

- **View Document History** In TFS repositories, you can view the history of a document. See "Viewing the Document History" on page 79.
- **View File Properties** In TFS repositories, you can view the properties of a document. See "Viewing File Properties" on page 80.
- **Work 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. See "Working Offline" on page 81.

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.

HOW 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. In the Team Authoring window, select the new file.
4. Click the **Check In** button. The Check In dialog opens.
5. 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 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 the next page.



NOTE: 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 (**File > Doc-To-Help > Common Tasks > Options > Team**).



NOTE: For other tasks, see "Working in a Team Project" on page 70.

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 target properties
- Change style properties
- Change topic properties
- Edit the table of contents

HOW TO CHECK OUT THE PROJECT FILE

1. In the Team Authoring window, select the project (D2H) file.
2. Click the **Check Out** button. The Check Out dialog opens.
 - **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 from the following:
 - **Unchanged** Will keep the existing lock.
 - **Check Out** Will prevent other users from checking the file in or out.
 - **Check in** Will allow 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 (**File > Doc-To-Help > Common Tasks > Options > Team**).

3. After edits are made to the project, in the Team Authoring window select the project (D2H) file.
4. Click the **Check In** button. The Check In dialog opens.
5. Confirm the check in and click the **Check In** or **OK** button.



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.



NOTE: For other tasks, see "Working in a Team Project" on page 70.

Comparing Documents

In TFS repositories, you can compare two versions of a document.

HOW TO COMPARE DOCUMENTS

1. In the Team Authoring window, select the file.
2. Click the **Compare** button. The Compare dialog opens.
3. Choose the file versions and click the **OK** button.



NOTE: To specify the compare and merge tools for team authoring, use the Doc-To-Help Options dialog (**File > Doc-To-Help > Common Tasks > Options > Team**).



NOTE: For other tasks, see "Working in a Team Project" on page 70.

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.

HOW TO DELETE A FILE FROM THE TEAM PROJECT

1. Click the **Team Authoring** button on the Project panel. The Team Authoring panel is displayed.
2. Click the **Check Out** button. The Check Out dialog opens.
 - **For File and Web Repositories** After the file is checked out, click the **Close** button.
 - **For TFS Repositories** Do not set the lock type to Check Out, as this will cause TFS to set two different Pending Change states for the file to be deleted.

To set the default lock type, use the Doc-To-Help Options dialog (**File > Doc-To-Help > Common Tasks > Options > Team**).
3. In the Documents window pane, select the file you would like to delete.
 - **For File and Web Repositories** : Check out the file to be deleted, then delete the file.
4. Right-click on the document, select **Team Authoring > Check In Pending Changes**.
5. Click the **Remove Document** button.
6. In the Team Authoring window, select the project (D2H) file.
7. On the Team Authoring panel, right-click on the Project file and select **Check In Pending Changes**.
8. Confirm the check in and click the **Check In** or **OK** button.



NOTE: For other tasks, see "Working in a Team Project" on page 70.

Editing a Document

To edit a document for a team authoring project, 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.)

HOW TO CHECK OUT A DOCUMENT OR FILE

In the Team Authoring window, you should click the **Refresh View** button before checking out a file to update.

1. In the Team Authoring window, select the file.
2. Click the **Check Out** button. The Check Out dialog opens.
 - **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 from the following:
 - **Unchanged** Will keep the existing lock.
 - **Check Out** Will prevent other users from checking the file in or out.
 - **Check in** Will allow 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 (**File > Doc-To-Help > Common Tasks > Options > Team**).

3. Click the **Open** button. The file opens.
4. Make your edits and save them. Then close the file.
5. Click the **Check In** button. The Check In dialog opens.
6. 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.



NOTE: For other tasks, see "Working in a Team Project" on page 70.

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.

HOW TO GET THE LATEST VERSION OF A FILE

1. In the Team Authoring window, select the file.
2. Click the **Get Latest Version** button.
3. If you are using a file or web repository and you selected the project (D2H) file, the Get Latest Version dialog opens. Confirm the information and click the **OK** button.



NOTE: In TFS projects you have the option of retrieving a specific version of a file from the repository. To do so, click the **Get Latest Version** drop-down and choose **Get Specific Version**. The Get Specific Version dialog 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.



NOTE: For other tasks, see "Working in a Team Project" on page 70.

Viewing the Document History

In TFS repositories, you can view the history of a document.

HOW TO VIEW THE DOCUMENT HISTORY

1. In the Documents view in the Doc-To-Help project panel, select the file.
2. Right-click on the file and select **Team Authoring > View History**. The History dialog opens.
3. Review the information and click the **Cancel** or **Apply** button.



NOTE: You can also enter a comment on a file using this dialog.



NOTE: For other tasks, see "Working in a Team Project" on page 70.

Viewing File Properties

In TFS repositories, you can view the properties of a document.

HOW TO VIEW FILE PROPERTIES

1. In the Document view on the Doc-To-Help Project panel, select the file.
2. Right-click and from the menu choose **Team Authoring > Properties**. The Properties dialog opens.
3. Review the information and click the **Close** button.




NOTE: For other tasks, see "Working in a Team Project" on page 70.


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.

HOW TO WORK OFFLINE

1. Open the working copy of the project.
2. Click the **Go Offline** button in the Team Authoring panel. The Go Offline dialog opens.
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 **OK**. If you chose Restricted Project editing, the Unrestricted Project Editing button will appear in the Team Authoring tab. It lets you 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 opens and displays the changes that occurred while offline. If you approve, click **Go Online**.

 **NOTE:** 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.

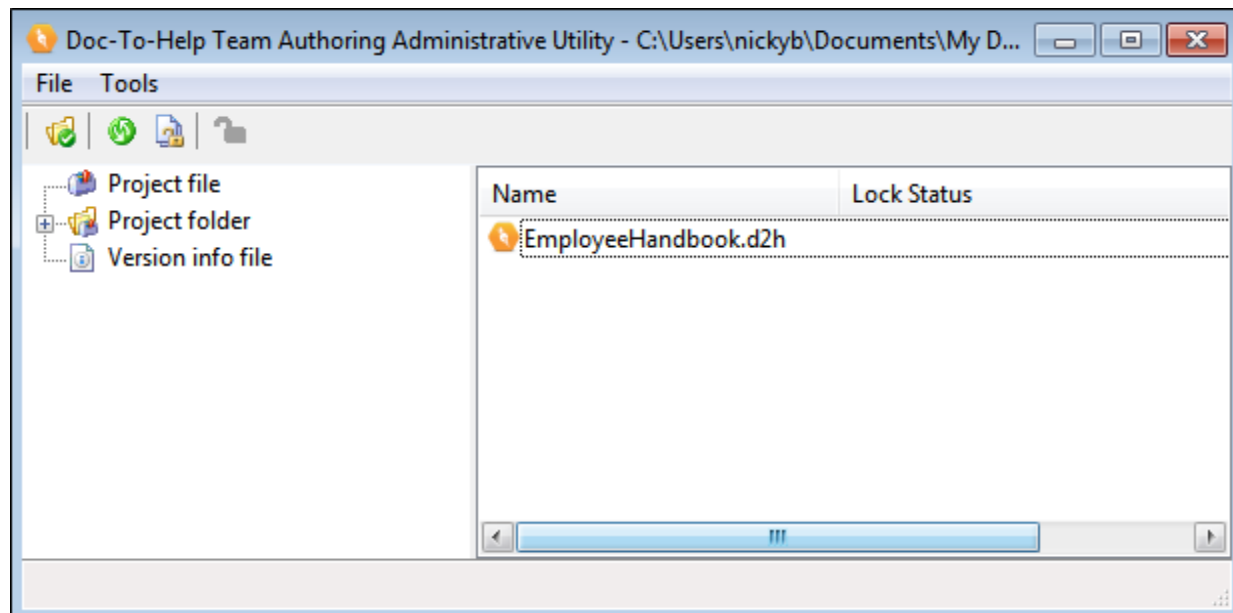
 **NOTE:** If you need to completely disconnect your working copy from team authoring support, the administrator must detach the project. See "Removing Team Authoring Support From a Working Copy" on page 87.

 **NOTE:** For other tasks, see "Working in a Team Project" on page 70.

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. See "Team Authoring" on page 57.

C1D2HTeamAdmin.exe can be found in \\Program Files (x86)\\MadCap Software\\DocToHelp. Double-click on it to open.



TASKS ASSOCIATED WITH ADMINISTRATIVE FUNCTIONS

In addition to setting up a team project (see "Setting Up a Team Project" on page 62), following are the primary tasks of an administrator:

- **Change 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. See "Changing Repository Settings" on the next page.
- **Remove Backup Files** To conserve the repository disk space, an administrator should delete unneeded backup files from time to time. See "Removing Backup Files From the Repository" on page 86.
- **Remove Team Authoring Support** You can remove team-authoring support from a local, working copy of a project. See "Removing Team Authoring Support From a Working Copy" on page 87.
- **Unlock Files in 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. See "Unlocking Files in the Repository" on page 88.
- **Unlock DHV File** In certain circumstances, you may need to unlock a special file, <project_name>.dhv, located in the D2HTeamInfo folder created by Doc-To-Help. See "Unlocking the DHV File" on page 89.
- **Upgrade 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. See "Upgrading a Team Authoring Project" on page 90.



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.

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.

HOW TO CHANGE THE REPOSITORY SETTINGS

1. Double-click **C1D2HTeamAdmin.exe**. See "Team Authoring Administrative Functions" on page 82.
2. Select **File > Open Team Project** to connect to the repository. The Load Remote Project dialog opens.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
4. Select a team project (or from the drop-down choose **Select Team Project** to browse for a project) and click **OK**.
5. Select **Tools > Repository Settings**. The Repository Settings dialog opens.
6. 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 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.
7. Click **OK**.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

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 by deleting all or part of the contents of the Backup folder in the repository. Each version number has its own subfolder inside the Backup folder, so you can delete only particular versions, if necessary. If you wish, back up those folders elsewhere before deleting them.

HOW TO DELETE BACKUP FILES IN THE REPOSITORY

1. Double-click **C1D2HTeamAdmin.exe**. See "Team Authoring Administrative Functions" on page 82.
2. Select **File > Open Team Project** to connect to the repository. The Load Remote Project dialog opens.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
4. Select a team project (or from the drop-down choose **Select Team Project** to browse for a project) and click **OK**.
5. Select **Tools > Clear Repository Backup**. The Clear Repository Backup dialog opens.
6. In the lower part of the dialog, select **Delete all backup files older than**.
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 are removed from the repository.

The administrator can also use this dialog to change the backup settings specified when the project was first shared. See "Changing Repository Settings" on page 84.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

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 target) or working offline (see "Working Offline" on page 81). 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 (see "Team Authoring Administrative Functions" on page 82).

HOW TO REMOVE TEAM AUTHORIZING SUPPORT FROM A WORKING COPY

1. Double-click **C1D2HTeamAdmin.exe**.
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 opens, 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.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

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.

HOW TO UNLOCK FILES IN THE REPOSITORY

1. Double-click **C1D2HTeamAdmin.exe**. See "Team Authoring Administrative Functions" on page 82.
2. Select **File > Open Team Project** to connect to the repository. The Load Remote Project dialog opens.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
4. Select a team project (or from the drop-down choose **Select Team Project** 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. From the **Project Folder** node on the left pane, select the folder, then from the right pane select the file.
6. On the toolbar click the **Unlock** button, or from the right-click menu choose **Unlock**.



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.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

Unlocking the DHV File

Most team authoring actions create a temporary lock in the repository for the duration of the action. This is done to prevent conflicts between different team members who are 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.

HOW TO UNLOCK THE DHV FILE IN THE REPOSITORY

1. Double-click **C1D2HTeamAdmin.exe**. See "Team Authoring Administrative Functions" on page 82.
2. Select **File > Open Team Project** to connect to the repository. The Load Remote Project dialog opens.
3. Select the **Repository Type**. (If necessary, enter your login credentials.)
4. Select a team project (or from the drop-down choose **Select Team Project** to browse for a project) and click **OK**.
5. On the left click the **Version info** node and on the right select the <project name>.dhv file.
6. On the toolbar click the **Unlock** button or from the right-click menu choose **Unlock**.



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.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

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.

HOW TO UPGRADE A TEAM AUTHORIZING PROJECT

The following steps should be performed by the administrator.

1. Install the new version of Doc-To-Help.
2. Double-click **C1D2HTeamAdmin.exe**. See "Team Authoring Administrative Functions" on page 82.
3. Select **File > Open Team Project** to connect to the repository. The Load Remote Project dialog opens.
4. Select the **Repository Type**. (If necessary, enter your login credentials.)
5. Select a team project (or from the drop-down choose **Select Team Project** 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 **File > Doc-To-Help > Team Authoring > Get Latest Version of the Project**.



NOTE: For other tasks, see "Team Authoring Administrative Functions" on page 82.

APPENDIX

PDFs

The following PDFs are available for download from the online Help.

On the Doc-To-Help Start Page, click the **PDF Guides** icon to access them.

Context-sensitive Help Guide

Design Guide

Documents, Templates, and Content Guide

Getting Started Guide

Getting Started Tutorial

Project Management Guide

Shortcuts Cheat Sheet

Targets Guide

Touring the Workspace Guide

Upgrading Old Projects Guide

What's New Guide