madcap software

USER GUIDE

MADCAP DOC-TO-HELP 6

Targets

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 4

CHAPTER 5

CHAPTER 6

APPENDIX

CHAPTER 1

Introduction

Before you generate output from your project, there are various tasks that you are likely to perform while developing your targets. To begin with, you need to determine what kind and how many types of output you want to provide for your end users.

This chapter discusses the following:

Output Types

There are several types of online output (Eclipse Help, EPUB, Microsoft Help 2.0, Microsoft HTML Help, Microsoft Help Viewer, JavaHelp, NetHelp) and one type of print-based output (Manual) that you can produce in Doc-To-Help. Each output type has its own set of advantages. See "Output Types" on page 9.

Targets

It is easy to confuse output types with targets, but they are two different (although related) concepts. A target is one instance of an output type. It is the engine that takes all of your files and settings, and brings them together to produce the end result. When you build your final output, you are essentially building one or more of the targets in your project.

When you create a new Doc-To-Help project, one target of each output type is added to your project.

However, any targets added to the project are just a starting point for you. You can rename them to reflect the nature of your project. For example, if you are writing a Help system for a software program called FictionSoftPro, you could rename a target "FictionSoftPro." Also, just because only one target was added when you first created the project, this does not mean that you are limited to just that target in your project. You can add as many new targets as you need, using any of the available formats. Each target has properties that you adjust to change the way the target behaves, as well as the way it looks and feels.

Conditions

A condition is a single-sourcing feature that you can apply to files or to different areas of your content, so that some information displays in some outputs but not in others. Although there are several settings that are important in each target, conditions are particularly important because they are central to your ability to single-source output from one set of source content. See the online Help or the *Documents, Templates, and Contents Guide*.

> ☆ EXAMPLE
>
> Let's say you have two different audiences—beginners and advanced users. The content in your project is the same in most places for both audiences. However, there are sections that apply only to the beginners, and other sections that apply only to the advanced users. You can use one condition to mark the sections for the beginners only, and you can use another condition to mark the sections for advanced users only. This lets you create one output for the beginners and another output for the advanced users without having to create two separate projects.

Steps Associated With Targets

Following are the primary steps when developing targets:

1. Determine Output Type The first task in developing output for your project is to determine which type of output is most appropriate for your needs. You might even need to produce multiple outputs and require more than one output type. See "Output Types" on page 9.
2. Create/Delete Targets Every target in a project has a particular output type assigned to it. You can add multiple targets to a project. For example, your project might end up containing three targets that are all based on the NetHelp output type and eight that are based on the Manual output type. While you create new targets, you can also remove targets provided by Doc-To-Help that you don't plan to use. See "Creating and Deleting Targets" on page 52.
3. Edit Target Settings Using the Help Targets dialog, you can apply all kinds of settings to a target. This includes renaming the target to reflect the nature of your project, setting attributes (choosing the conditions to include or exclude from the output), and much more. The settings that are available are different for each output type. See "Editing Target Settings" on page 54.
4. Add Transformations Using the Transformation Wizard, you can insert boilerplate content or code into every topic in your targets. Examples include: headers, footers, or code needed for technology such as analytics. See "Transformations" on page 55.
5. Select Targets When you want to work with a target, or you are ready to build its output, you need to select that target.

   If your project has target-specific tables of contents, you can select the target you want to work with by opening the Doc-To-Help Project panel (Project ribbon> Project Panel), then opening the Contents window pane and selecting a target-specific table of contents from the drop-down.
6. Build Targets After selecting a target, you can generate output for it. See "Building Targets" on page 63.

CHAPTER 2

Output Types

There are several types of online output (Eclipse Help, EPUB, Microsoft Help 2.0, Microsoft HTML Help, Microsoft Help Viewer, JavaHelp, NetHelp) and one type of print-based output (Manual) that you can produce in Doc-To-Help. Each output type has its own set of advantages.

You can configure targets based on these output types by using the Help Targets dialog. See "Editing Target Settings" on page 54.

This chapter discusses the following:

NetHelp

NetHelp is a cross-platform, browser-based output that is displayed in the user's default browser. It is uncompiled HTML Help, so the number of files delivered for the project will depend on the number of topics in the project.

Also, NetHelp is jQuery-based. jQuery is a flexible web technology that separates content from presentation and is easy to customize.

If you need to create an output that can be hosted on your web server, but still be accessed through the application, you should create NetHelp. It produces HTML output that is browser-based and can be hosted locally or on any web server.

Benefits of NetHelp

Following are some of the greatest benefits of using NetHelp:

- True Separation of Content and Presentation Doc-To-Help outputs raw XML and NetHelp consumes it. NetHelp is merely the set of scripts and pages used to display and lay content out. In fact, you can use the raw XML output with your own systems. That is how we are able to create SharePoint wikis (see the online Help or the *Design Guide)*.
- Best Looking, Most Themable The promise of jQuery UI is to deliver a standardized and easy-to-edit set of styling standards. That promise carries through to NetHelp. It ships with many of JQuery UI's default themes and a few of our own. The icing on the cake is that it is simple to edit. You can use our theme designer, edit the CSS file directly, or use the Theme Roller found on jQuery UI's website. See the online Help or the *Design Guide*.
- Lightweight As already mentioned NetHelp output is only a set of CSS files and scripts. Files sizes are small and performance is fast.
- Unique URLs With NetHelp output, URLs to every topic are displayed in the address bar. This helps search engine optimization (SEO) if you are deploying publicly. It also makes sending links to topics easy.

NOTE: For more information on creating accessible NetHelp, see the online Help.

NOTE: If no index entries are added to a project, the Index tab or panel will automatically not display in NetHelp outputs.

Supported Browsers

Following are browsers that support NetHelp:

- Internet Explorer 9 or higher
- Firefox 3 or higher
- Opera 9.6 or higher
- Apple Safari 3.1 or higher
- Google Chrome (all versions)

Following are browsers that support the NetHelp responsive theme:

- Desktop Internet Explorer 9 or higher
- Mobile Platforms Android 4.0 or higher, iOS

Extending Targets

With Doc-To-Help, you can extend your NetHelp targets further in two ways.

- You can gather user feedback and build a user community by incorporating DISQUS into NetHelp systems posted on the web. See "Adding DISQUS Commenting to NetHelp Targets" below.
- You can embed NetHelp targets (the full target or single topics) into web pages with or without iframes. See "Embedding NetHelp Targets Into Web Pages" on page 16.

Adding DISQUS Commenting to NetHelp Targets

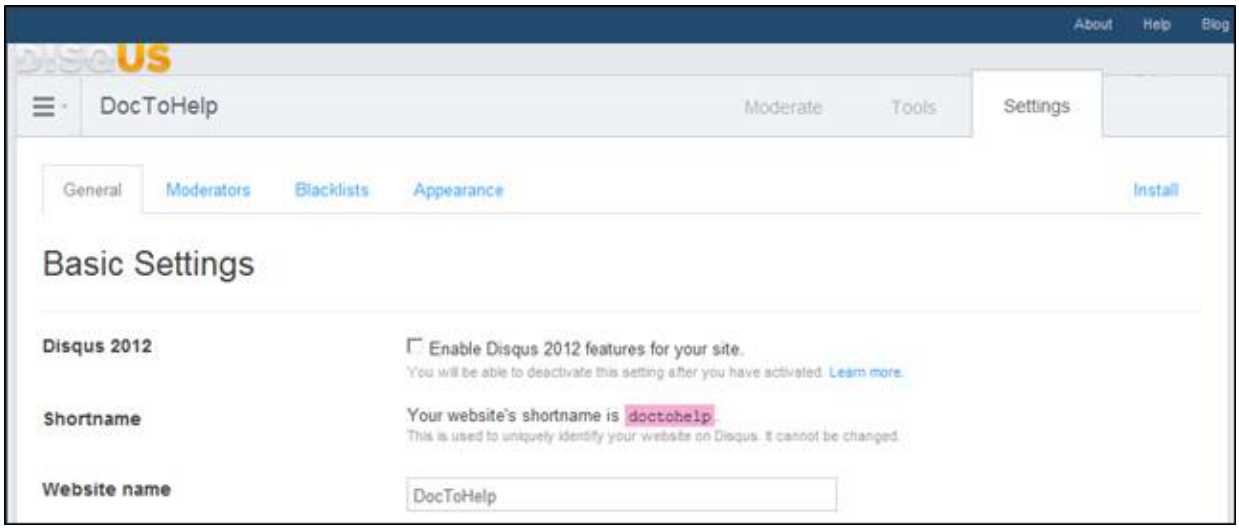
If you would like to gather user feedback for your online output—such as commenting and ratings—and build a user community, you can do so by incorporating DISQUS in NetHelp targets. DISQUS (http://disqus.com/) is an online discussion and commenting service for websites that uses a networked platform and works on the following browsers: Internet Explorer, Firefox, Chrome, Safari, and Opera.

All settings, comment moderation, appearance, etc. are managed through your DISQUS account.

By default, DISQUS comments are enabled for every topic in your NetHelp target, but it is possible to exclude specific topics, or entire topic types from commenting.

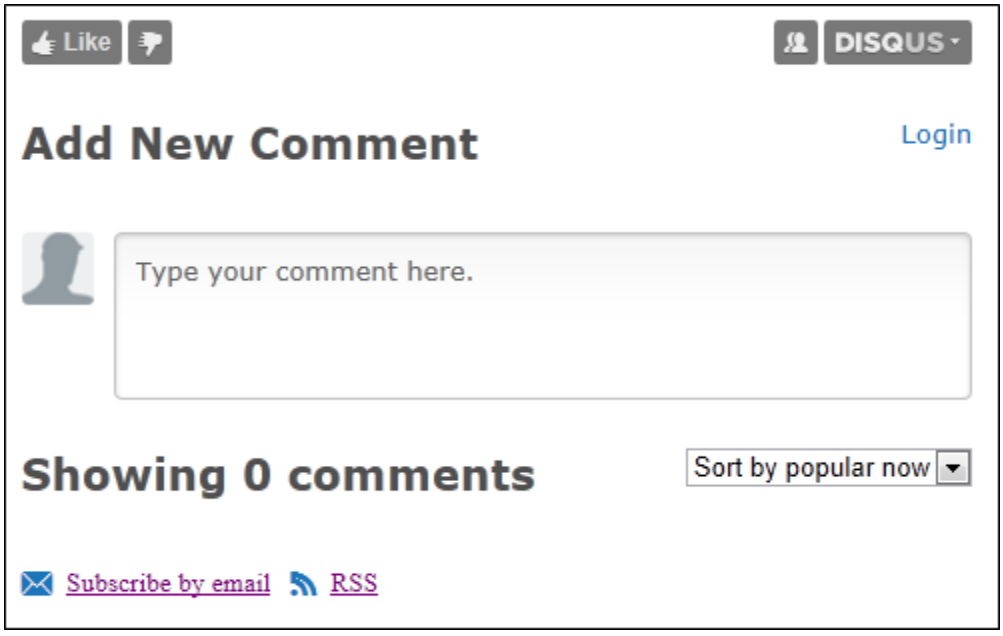
HOW TO ENABLE DISQUS IN NETHELP TARGETS

1. Create a DISQUS account at http://disqus.com/register and define your DISQUS shortname.



2. In Word, select the **Target** ribbon and click **Target Properties**. The Help Targets dialog opens.

3. On the left select the **NetHelp** target.
4. In the **Comments** section, set the **Use comments** field to **DISQUS**, and enter your **DISQUS shortname**.
5. Click **OK**.
6. On the **Target** ribbon, click the **Rebuild** button.
7. Post your NetHelp output on your website. At the bottom of each topic, DISQUS commenting and rating options will appear.



If you would like to change the skin of your DISQUS comments; set up comment moderators and moderation rules; set the language, Twitter, and Facebook features; or create blacklists, you can do so using your DISQUS account.

HOW TO EXCLUDE TOPIC TYPES FROM DISQUS COMMENTING

1. In Word, open the **Project** ribbon and click the **Styles** button. The Project Styles dialog opens.
2. On the left, select the topic type you would like to exclude.
3. In the **Display** section, clear the **Use comments** check box.
4. Click **OK**.
5. On the **Target** ribbon, click the **Rebuild** button.

HOW TO EXCLUDE INDIVIDUAL TOPICS FROM DISQUS COMMENTING

1. In Word, select the **Project** ribbon and click **Topics Panel**.
2. Right-click on the desired topic, and choose **Properties**. The Topic Properties dialog opens.
3. In the **Appearance** section, check the **Disable comments** check box.
4. Click **OK**.
5. On the **Target** ribbon, click the **Rebuild** button.

Embedding NetHelp Targets Into Web Pages

Doc-To-Help's NetHelp is architected to make it easy to embed a single topic, or an entire Help system, into a web page. This lets you stream updated content to your website without web development.

There is some lightweight front-end configuration that must be made to your website. Your output does not need to be hosted on the same web server as your main site. In that case, you would add an iframe, because using an iframe makes it possible to place the web site and the NetHelp target on different servers (domains). If both your output and your website will be hosted on the same web server, an iframe is not needed.

HOW TO EMBED A NETHELP TARGET IN AN IFRAME

This method allows to place the website and NetHelp output on different servers (domains).

1. Publish your built NetHelp output to a server.

   > ☆ EXAMPLE
   >
   > Here is an example of a published NetHelp location:
   >
   > http://mysite.com/help/mytarget/
   >
   > The full URL to the NetHelp output would be:
   >
   > http://mysite.com/help/mytarget/index.html

2. Add an iframe element to your web page that contains the NetHelp output. The iframe tag must have the "src" attribute set to the default NetHelp page ("index.html") in the target folder.

   > ☆ EXAMPLE
   >
   > To embed full NetHelp output without parameters add the following:
   >
   > ```
   > <iframe src="http://mysite.com/help/mytarget/index.html">
   > </iframe>
   > ```

The URL in the "src" attribute can be used to set NetHelp output parameters if desired. For explanations of these parameters, see the online Help or the *Context-sensitive Help Guide*.

☆ EXAMPLE—IFRAME EMBEDDED IN WEBSITE

Here is an example of code used for an iframe embedded in a website:

```
<iframe src="http://helpcentral.componentone.com/nethelp/
c1intellispell/index.html" width="1000px" height="350px">
</iframe>
```

To embed a single topic on a web page, use the following:

```
<iframe src="http://mysite.com/help/mytarget/index.html?topiconly=true">
</iframe>
```

☆ EXAMPLE—SINGLE-TOPIC IFRAME EMBEDDED IN WEBSITE

Here is an example of code used for a single-topic iframe embedded in a website:

```
<iframe src="http://helpcentral.componentone.com/nethelp/
c1intellispell/index.html?topiconly=true" width="300px"
height="500px"></iframe>
```

☆ EXAMPLE—GENERIC

Here is a generic example:

```
<!DOCTYPE html>
<html>
<head>
    <title>IFrame example</title>
    <meta http-equiv="Content-Type" content="text/html; char-
    set=utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
</head>
<body style="padding: 10px;">
    <p>Some text before help frame.</p>
    <!-- HELP FRAME -->
    <iframe src-
    c="http://mysite.com/help/mytarget/index.html"></iframe
    width="800px" height="400px" >
    <!-- END OF HELP FRAME -->
    <p>Some text after help frame.</p>
</body>
</html>
```

HOW TO EMBED A NETHELP TARGET WITHOUT AN IFRAME

You can embed NetHelp output into any HTML element (e.g., into a DIV element).

This option is only available if the NetHelp output and the container page (into which you embed the NetHelp output) are placed in the same domain.

1. Publish your built NetHelp output to a server.

   > ☆ EXAMPLE
   >
   > Here is an example of a published NetHelp location:
   >
   > http://mysite.com/help/mytarget/
   >
   > The full URL to the NetHelp output would be:
   >
   > http://mysite.com/help/mytarget/index.html

2. Add a container element (any HTML element) to your page, which will contain the NetHelp output.

   The container must have the CSS property **position** set to either the **relative**, **absolute**, or **fixed** value. This is a restriction of the Doc-To-Help standard themes (Tabs and Accordion). You must also set the height for the container if you want to display NetHelp output in full mode (you do not need to do this for topic-only mode).

   > ☆ EXAMPLE
   >
   > Here is an example:
   >
   > ```
   > 
   > 
   > ```
   >
   > If you want to display the NetHelp output in topic-only mode, the container element may look like this:
   >
   > ```
   > 
   > ```

3. Add the "script" tag to the "head" section of the page. This script tag must have the "src" attribute set to the "js/nethelp.connect.js" file located in the NetHelp target folder. You can also set the following optional attributes for this tag to set up the target:

   - "data-placeholder" This specifies the jQuery-selector of the container element created in step 2. If this attribute is missed, the NetHelp engine will search for an element with the attribute "data-c1-role" equal to the "nethelp" value. If no such element is found, the "body" element will be used as a container for the NetHelp output. The NetHelp output is placed into a container element and replaces all its content.
   - "data-start" This specifies the URL of the topic that will be shown instead of the default topic when the NetHelp output is loaded.
   - "data-topiconly" If set to "true," the NetHelp output will be displayed in the topic-only mode (without header, toolbars, and Contents, Index and Search tabs).
   - "data-settings" This specifies the URL of the NetHelp target settings file (by default it is the "settings.xml" file in the root target folder).
   - "data-responsive" This is used only for Responsive themes, set to "true."

☆ EXAMPLE—TABS AND ACCORDION THEMES

Here is an example for Tabs and Accordion themes:

```
<script type="text/javascript" src="/help/mytarget/
js/nethelp.connect.js" data-placeholder="#mytarget">
</script>
```

☆ EXAMPLE—RESPONSIVE THEME

Here is an example for the Responsive theme:

```
<script type="text/javascript" src="/help/mytarget/
js/nethelp.connect.js" data-placeholder="#mytarget"
data-responsive="true"></script>
```

NOTE: If you are viewing the website from the local file system, the NetHelp output folder must be located in a subfolder of your site. Otherwise your browser may block the NetHelp files.

EXAMPLE—GENERIC

Here is a generic example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline example</title>
    <meta http-equiv="Content-Type" content="text/html; char-
    set=utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <script type="text/javascript" src-
    c="/help/mytarget/js/nethelp.connect.js"
        data-settings="/help/mytarget/settings.xml"
        data-start="/help/mytarget/Documents/home_top-
        ic.htm"></script>
</head>
<body style="padding: 10px;">
    <p>Some text before help frame.</p>
    <!-- HELP FRAME -->
    <div data-c1-role="nethelp" style="position:relative;
    width:800px; height:400px;"></div>
    <!-- END OF HELP FRAME -->
    <p>Some text after help frame.</p>
</body>
</html>
```

Search Options for NetHelp

Doc-To-Help's NetHelp target includes powerful search features that you can customize for your needs, such as server-side search.

By default, JavaScript Client search is used in NetHelp targets and requires no server setup. JavaScript Server search requires setup, but will improve the search speed of NetHelp targets installed on a web server. See "Setting Up Server-Side Search for NetHelp Targets" on the next page.

EXACT PHRASE, BOOLEAN, AND FUZZY SEARCHES

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

- Exact phrase search means that if you enclose a phrase in double quotes, the search will be limited to that exact phrase in the online output.

  ☆ EXAMPLE

  "sports teams"

- Boolean search means that you can use "AND" or "OR" (no quotes) between words or exact phrases when searching. You can also use "AND NOT" or "NOT" before a word or exact phrase to exclude topics containing that phrase from the results. By default, if there is no "AND" or "OR" between words, "AND" is assumed.

  ☆ EXAMPLE

  football or hockey, sports and not baseball

- Fuzzy search will display alternative search options (and results) if the user enters a search term that is close to the term entered.

  ☆ EXAMPLE

  Let's say you search for sorts. The search will return "No topics found. Did you mean: sports."

SETTING UP SERVER-SIDE SEARCH FOR NETHELP TARGETS

The following explains how to set up server-side search for NetHelp targets.

GENERAL INFORMATION

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

SETTING UP THE SEARCH TYPE

1. In Word, select the **Target** ribbon, and click **Target Properties**. The Help Targets dialog opens.
2. On the left side of the dialog, select the **NetHelp** target.
3. In the **Advanced** section, in the **Search type** field, choose **JavaScript Server**.
4. Build the target.

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

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

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

SETTING UP A SERVER FOR A SINGLE TARGET

As a prerequisite to the following steps, Node.js must be installed on the server. It can be downloaded here: http://nodejs.or.

1. Copy your built target to the web server.

   > ☆ EXAMPLE
   >
   > c:\d2h\targets\MyOutput

2. Depending on the web server that you use (IIS, Apache, etc.), set up your server to serve the target files (e.g., in IIS, set up a virtual directory).

   > ☆ EXAMPLE
   >
   > http://localhost/d2h/MyOutput can point to c:\d2h\targets\MyOutput.

3. Run the search server in node.js. For Windows:
   - Open the command line (**Start > All Programs > Accessories > Command Prompt**).
   - Verify that the current directory is C:\. If it is not, type `C:` and press **Enter**. This will change current directory to C:\.
   - Change the current directory and set it to the **js\nodejs** subfolder in the built target folder (e.g., cd "c:\d2h\targets\MyOutput\js\nodejs" and press **Enter**.
   - Type `"node index.js"` and press **Enter**.

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

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

VIEWING AND TESTING SERVER-SIDE SEARCH

1. Open the built target in a browser, not from the local file system (i.e., the URL in the browser must start with the "http://" prefix).

   ☆ EXAMPLE

   http://localhost/d2h/MyOutput

2. Try searching for a word.

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

SETTING UP A SINGLE SEARCH SERVER FOR MULTIPLE TARGETS (FOR WINDOWS)

1. Repeat steps 1-2 from "Setting Up a Server for a Single Target" on page 25.
2. Copy the search server files to a separate folder (e.g., c:\d2h\nodejs).
3. Go to the folder containing the search server files (**c:\d2h\nodejs**) and open the **settings.json** file. This is a configuration file for the server, in JSON format. By default it has two settings:
   - **port** is the port number that the search server will use. The default value is 8256. Make sure that this port isn't used by another application and isn't blocked by firewalls. Otherwise, you must change this value to any available port number.
   - **host** is the IP address on which the search server will wait for requests. The default value is "null" and will work in most cases. You need to change this setting only if the system has several network interfaces and you want the search server to use a specific network interface.
4. To handle multiple targets by a single search server, you need to add the "targets" option to this file with information about your targets. After modifying the "settings.json" file, it could look like this:

```
{
   "port": 8256,
   "host": null,
      "targets": [
         {
            "path": "/d2h/MyOutput/",
            "index": "c:\\d2h\\tar-
            gets\\MyOutput\\searchindex.js"
         },
         {
            "path": "/d2h/MyOutput/",
            "index": "c:\\d2h\\tar-
            gets\\MyOutput\\searchindex.js"
         }
      ]
}
```

Each target is described by two values:

- **path** is the virtual path you set up on the web server (IIS, Apache, etc.).
- **index** is a physical path to the search data of the target, the "searchinddex.js" file in the Target folder. Note that the character '\' must be expressed as '\\' in these strings.

5. Make sure you save all changes made to the "settings.json" file.
6. Repeat step 3 from "Setting Up a Server for a Single Target" on page 25.

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

UPDATING TARGET CONFIGURATIONS

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

1. Open the command line window and go to the folder containing the search server files

   > **EXAMPLE**
   >
   > "c:\d2h\nodejs" or "c:\d2h\targets\MyOutput\js\nodejs"

   See step 3 of "Setting Up a Server for a Single Target" on page 25.
2. Execute **node searchserverconfig.js**. This updates the port number in the configuration files in Target(s) folder.

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

NetHelp Local Installation

NetHelp installed locally can be given the Mark of the Web (MOTW). This means that anyone viewing your NetHelp locally will not receive a browser security warning first. (This ActiveX security warning is dependent on the security settings of the machine.) MOTW adds commentary text to every HTML file in the NetHelp target. It is turned on using the **Mark of the Web** check box in the Help Targets dialog. See "Editing Target Settings" on page 54.

MOTW ISSUES

There are some issues with the MOTW. For example, links to PDF will not open from NetHelp files using the MOTW. To learn more see:

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

HOW TO INSTALL NETHELP ON A CLIENT MACHINE

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

You can rename the NetHelp destination folder using the **Folder field** of the Help Targets dialog (from the **Target** ribbon, click **Target Properties**). The default file name can be renamed using the **Default file** field.

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

Additional Details

Details	Pros	Issues	Deliverable File(s)	File Location
Uncompiled HTML; deliverable is multiple HTML files. NetHelp can be Section 508 compliant. See the online Help.	Can be delivered on the web, installed locally, or both. Delivering on the web makes a continuous publishing model possible. The user's web browser is the Help window.	Number of files to distribute is daunting to some. Security issues can cause NetHelp installed locally to display an ActiveX warning, which may be solved by using the "Mark of the Web." See "NetHelp Local Installation" on the previous page.	Multiple HTM/HTML files. Default home page is index.htm. (You can change the default page name in the Help Targets dialog.)	By default the NetHelp folder of your project. The entire contents (including subfolders) of this folder must be included with your software application or posted on your server. Server-side search may be enabled for NetHelp. See "Search Options for NetHelp" on page 22.

NOTE: If you open or install NetHelp locally, it will not display in Chrome, IE Edge, or Opera because of limitations with those browsers, but it will display in other browsers. NetHelp deployed on a web server will open in Chrome, IE Edge, and Opera.

Microsoft HTML Help

The deliverable is compiled HTML Help (a single CHM file). HTML Help displays in a tripane window. Newer PC security measures can block viewing of CHM files across a network.

Details	Pros	Issues	Deliverable File(s)	File Location
Compiled HTML; deliverable is CHM file.	One file; tripane interface is easy to use.	Format is 10+ years old. Locally installed HTML Help files (those installed on the end user's client machine, not their server), work without issues. If you plan to install HTML Help on a server, there are security issues. See the following: http://support.microsoft.com/kb/902225	CHM	By default, the HTML Help folder of your project.

EPUB

EPUB stands for "electronic publication" and is standard of the International Digital Publishing Forum (IDPF). This target is designed specifically for EPUB readers.

Details	Pros	Issues	Deliverable File(s)	File Location
EPUBs are readable on a variety of devices, including Nooks. The content display adjusts for the device, as well as the device settings.	You can create interactive, portable books that can be read on multiple devices. Doc-To-Help supports the latest specification, EPUB 3.0, as well as 2.0.1.	Dynamic content is not supported, but the content will be displayed.	EPUB	By default, the EPUB folder of your project. You must add your EPUB to your reader (such as Calibre) in order to view it. To set the default EPUB reader in Doc-To-Help, go to **File > Doc-to-Help > Common Tasks > OptionsFile > Tools > Options**. In the Options dialog, select the **Viewers** button and choose the **EPUB reader executable**.

NOTE: Many devices don't support Cascading Style Sheets or only support them partially, so your EPUB display can vary depending on the device.

NOTE: Doc-To-Help's EPUB target was tested with the following readers:

- Calibre E-book management (http://calibre-ebook.com/)
- Sony Reader for PC (http://ebookstore.sony.com/download/)
- NOOK for PC (http://www.barnesandnoble.com/u/free-nook-apps/379002321/)
- Adobe Digital Editions (http://www.adobe.com/products/digitaleditions/)

NOTE: Dynamic content is not supported in EPUBs, but the content will be displayed as follows:

- Popups and Glossary Terms will be opened in the current window.
- All content in Collapsible Sections (created with the Collapsible Section button) will be displayed (just as it would in Manual Targets).
- Topic links that were specified to open in secondary windows will open in the current window instead.
- Keywords with multiple entries and Groups don't open in a popup when clicked, but instead link to the Index, where those Keywords and Groups are displayed, along with the list of topics included in each. (The Index can be renamed in the Theme Designer. See the *Making It Look Guide* for more information, or the online help.)

NOTE: Expanding Text and Dropdown Text (both created with the Inline Text button) may be displayed or hidden in EPUBs. The option you prefer can be set in the Help Targets dialog using the Show expanding text and Show dropdown text check boxes. The hidden portion of inline text will always remain hidden.

Eclipse Help

The deliverable is Help for the Eclipse environment.

Details	Pros	Issues	Deliverable File(s)	File Location[1]
Help target for the Eclipse development environment. Eclipse was released by IBM in the early 2000s.	An Eclipse Help system is a plug-in for Eclipse. This plug-in supports a table of contents, index, search, and context-sensitive Help .	Eclipse Help is compatible with all browsers, but the Eclipse SDK must be installed on the same server as the Help system. Eclipse Help does not support opening topic links in secondary windows.	Multiple files See the following (search on "User Assistance Support"): http://help.eclipse.org/indigo/index.jsp	By default, the Eclipse folder of your project. To view Eclipse Help, you must have the Java Runtime Environment (JRE) and Eclipse installed (either the 32 or 64-bit versions). To view Eclipse Help using the View button for your target, you must have 32-bit Eclipse installed. Eclipse is available for download from: http://eclipse.org/downloads/

[1]After installation, the location of the eclipse.exe file should be set in the Doc-To-Help Options dialog through **File > Doc-To-Help > Common Tasks > OptionsFile > Doc-To-Help Options**.Click the Viewers tab to set the location of the eclipse.exe file.

JavaHelp

The deliverable is Help for Java applications.

Details	Pros	Issues	Deliverable File(s)	File Location
Java Help 1.1.3 is supported, as well as JavaHelp 2.0	JavaHelp software was developed to provide a standard Help solution for pure Java applications. JavaHelp software was released in April 1999, and is currently in release 2.0.	While there is no "standard" viewer, JavaHelp uses components from the HotJava browser for its display.	Multiple files; deliver the entire JavaHelp folder (its default name) in your project directory.	By Default, the JavaHelp folder of your project. To view a JavaHelp HelpSet, you must have the necessary files installed on your machine.

NOTE: To build and view JavaHelp, you must first install the necessary files from Oracle.

Do the following:

1. Install the Java Developer Kit (JDK).
   The JDK6 for Windows x86 and Windows x64 are available at:
   http://www.oracle.com/technetwork/java/javase/downloads/index.html
2. Download and unzip the JavaHelp files from http://olex.openlogic.com/packages/javahelp/2.0_05
3. Update your Environment Variables to tell your machine where to find the Java files.
   Go to Control Panel > System > Advanced Settings. The System Properties dialog will open.
   On the Advanced tab click the Environment Variables button. These variables need to be added or changed under System variables:
   - JAVA_HOME
     Value: C:\Program Files [or Program Files (x86)]\Java\jdk1.6.0 (use the location of your JDK installation)
   - JAVAHELP_HOME
     Value: C:\Program Files [or Program Files (x86)]\Java\jh2.0 (use the location of your jh2.0 directory)
   - JHHOME
     Value: C:\Program Files [or Program Files (x86)]\Java\jh2.0 (use the location of your jh2.0 directory)
   - Path
     This one will already exist, so edit it and append ;C:\Program Files [or Program Files (x86)]\Java\jdk1.6.0\bin (use the location of your JRE\bin directory)
     (Note: semicolons separate multiple paths)

As noted above, the exact file paths will vary based on Java version numbers and where you have JavaHelp and the JDK installed on your machine.

Microsoft Help Viewer

This target is only used to create Help for .NET components. Projects are integrated with Microsoft® Visual Studio 2010 and above.

See the following for MHV resources:

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

Details	Pros	Issues	Deliverable File(s)	File Location
Help target for those developing Help for Visual Studio 2010 and above only. Deliverable can be the reference doc-umentation only, or Doc-To-Help can be used to add narrative text.	Resembles Microsoft MSDN collection and uses some of the same underlying technology	Format can only be included in Help for Visual Studio 2010 and above.	HelpContentSetup.msha and <target base name>.mshc The MSHC is a zip file containing all of the Help content.	By default, the MSHelpViewer folder of your project. To specify the version of the Microsoft Help Viewer to use when view-ing Microsoft Help Viewer Targets, in Word, go to **File > Doc-To-Help > Common Tasks > Options**Doc-To-Help, go to **File > Tools > Options**. In the Options dialog, select the **Viewers** but-ton and choose the cor-rect version of the viewer. (Options avail-able depend on your Visual Studio install-ation.)

NOTE: In order to create and view Microsoft Help Viewer files you must install Visual Studio 2010 or greater. Other properties are set in the Help Targets dialog.

Microsoft Help 2.0

This target is only used to create Help for .NET components. These projects are integrated with Microsoft® Visual Studio 2002-2008.

Details	Pros	Issues	Deliverable File(s)	File Location
Help Target for those developing Help for Visual Studio 2002-2008 only. Deliverable can be the reference documentation only, or Doc-To-Help can be used to add narrative text.	Resembles Microsoft MSDN collection and uses some of the same underlying technology	Format can only be included in Help for Visual Studio 2002-2008. Programmers who distribute .NET components can integrate Help 2.0, because the .NET tools include the run-time components necessary to view Help 2.0 Help systems.	.HxS (topic files), .HxC (project file), .HxF (include file), .HxT (table of contents). .HxA (attributes), .HxK (index)Your deliverables may vary based on your application. For more on Microsoft Help 2.0 redistributables, see: http://msdn.microsoft.com/en-us/library/bb165722(VS.80).aspx	By default, the MSHelp folder of your project.

NOTE: To create and view .HxS, or Help 2.0, files, you must install Visual Studio.NET and VSHIK (Visual Studio Help Integration Kit).

The Namespace and Parent Namespace are set in the Help Targets dialog.

To build Help 2.0, Doc-To-Help also needs to know the locations of the Help 2.0 executable files. By default, Doc-To-Help assumes the following:

C:\Program Files\Microsoft Help 2.0 SDK\HxComp.exe

C:\Program Files\Microsoft Help 2.0 SDK\HxReg.exe

C:\Program Files\Common Files\Microsoft Shared\Help\dexplore.exe

You may have to create folders if they do not already exist.

Manual

Print-based output doesn't have to be printed, although it can be. With Doc-To-Help, you can generate Microsoft® Word files (DOC or DOCX), and PDF (portable document format) files. Either format can be sent to a printer, or provided as an online document. If you don’t want your file altered, it is best to provide it as a PDF.

Following are some tasks that are specific to the Manual output type:

- "Setting a Cover Image" on the next page
- "Setting Page Orientation" on page 47
- "Additional Details" on page 51

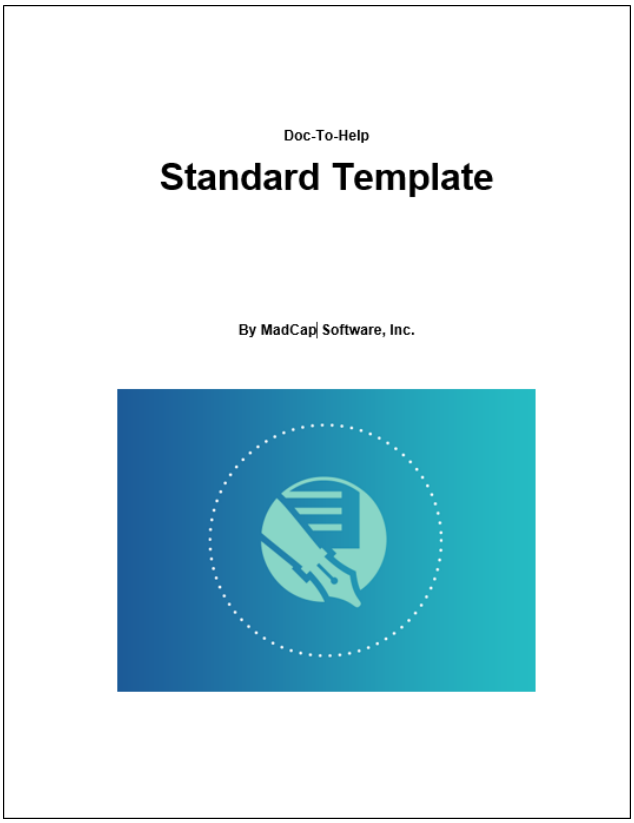
To make other changes to this target (as well as other types of output), see "Editing Target Settings" on page 54.

Setting a Cover Image

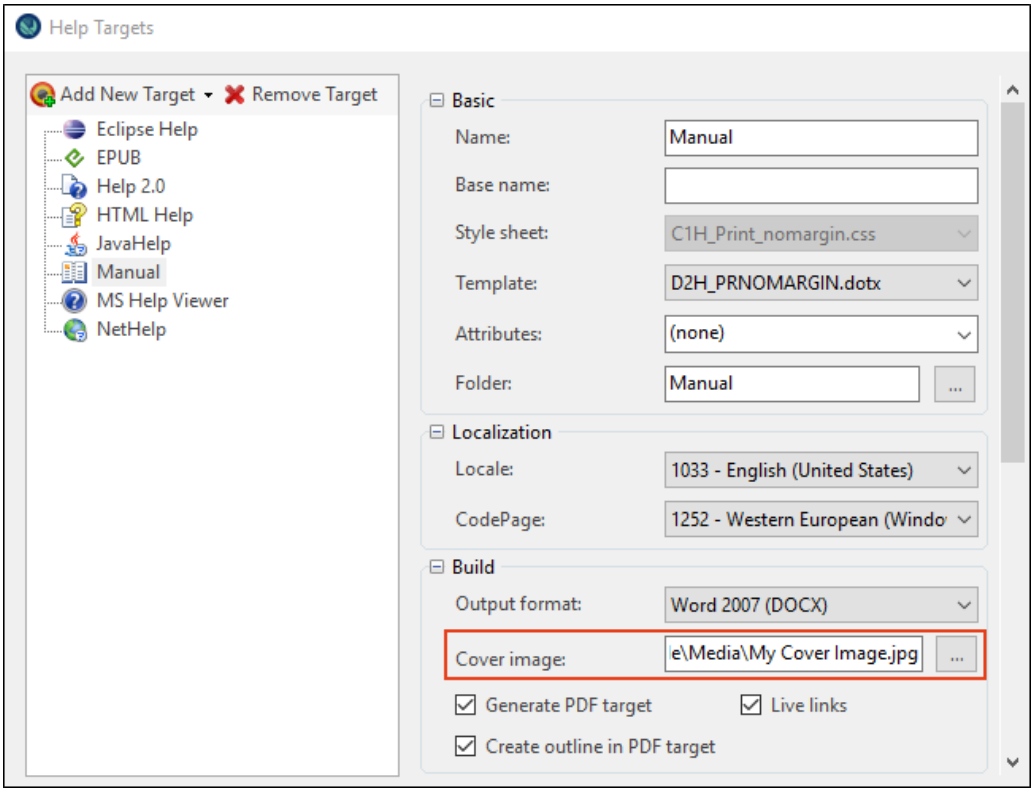
For the Manual target type, you can specify a cover image in the Help Targets dialog, as opposed to embedding it in the template file. This is simply an easier, more convenient option and lets you specify a cover image for specific targets.

EXAMPLE

Let's say your target template initially looks like this, with Doc-To-Help's default cover image embedded on the first page.



☆ You want to replace this image with your own. Instead of editing the template, replacing the Doc-To-Help image, you open the Help Targets dialog and point to your own image.



☆ After you generate the output, your image replaces the default Doc-To-Help cover image.

HOW TO SET A COVER IMAGE FOR A MANUAL TARGET

1. (Recommended) Add the desired image to your project. (Typically, images are stored in the Media folder.)
2. In Word, open the **Target** ribbon and click the **Target Properties** button. The Help Targets dialog opens.
3. On the left side of the dialog, select the **Manual** target.
4. In the **Build** section, specify the image in the **Cover image** field.

   In most cases, you will click  and select an image within your project, which produces a relative path to that image (e.g., Media\MyImage.png).

   > **NOTE:** It is also possible to use an absolute path and point to an external image, but a relative path to an image within the project is recommended.

5. Click **OK**.

COVER IMAGE BOOKMARK

This feature is based on a bookmark named "CoverImage." All source and Manual target templates provided by Doc-To-Help have this bookmark, and it is placed by default on the cover image at the beginning of the template. However, if you specify a different image in the Help Targets dialog, it will replace the default cover image that is embedded in the target template.

To define a new location for the cover image, you can edit the template, inserting a bookmark named "CoverImage" somewhere before the "ChapterHeading" bookmark (if it exists). Otherwise, the image will be inserted at the beginning of the document.

Setting Page Orientation

A property called "Section orientation" is available for Manual targets. It lets you specify a portrait or landscape configuration for pages within a given section break in a Word document.

You can set the section configuration on a specific paragraph style or an entire Manual target.

HOW TO SET THE SECTION CONFIGURATION ON A PARAGRAPH STYLE

Setting this on a paragraph style is a good option if you need the configuration to change from portrait to landscape within a single document. The configuration change continues in the document until it comes across another section break.

STEPS

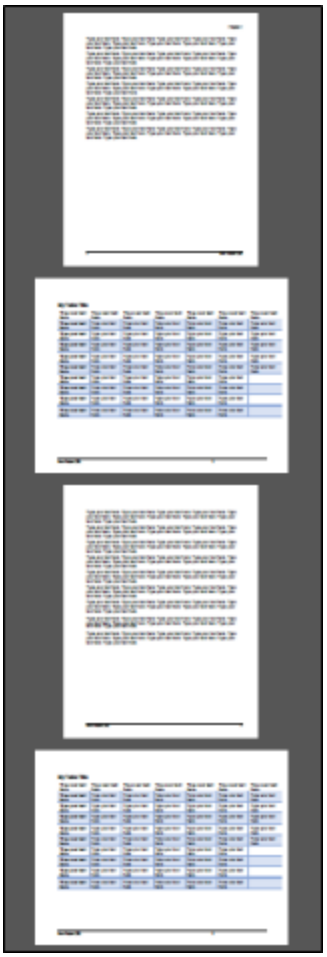
1. In Word, open the **Project** ribbon, and click the **Styles** button. The Project Styles dialog opens.
2. On the left side of the dialog, expand **Paragraph Styles** and select one of the styles.

   > NOTE: If you have are using a style in your source document that is not available in this dialog, click Add New Style. Then create a new style that has the same name as the style in your document.

3. On the right side of the dialog, in the **Basic** section, set the **Section break** field to one of the following:
   - **Next page**
   - **Even page**
   - **Odd page**
4. In the **Section orientation** field, select one of the following:
   - **Inherit from template** This is the default option. If selected, a new section created by the "Section break" property will have the same page orientation as that used in the template.
   - **Portrait**
   - **Landscape**
5. Click **OK**.

EXAMPLE

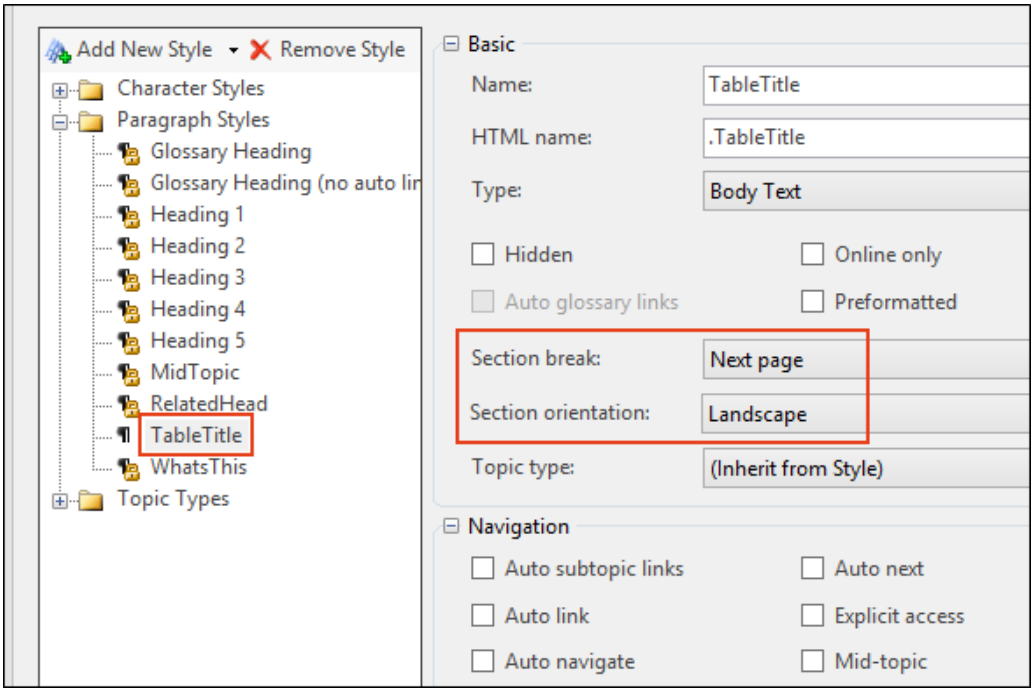
Let's say you have a document that uses a portrait landscape throughout. However, you have inserted several wide tables in the document, and you would like the pages showing those tables to be displayed with a landscape configuration whenever they appear in the Doc-To-Help print-based output. Like this:



Furthermore, in your Word template you've created a special style called "TableTitle" that you apply to a paragraph just above each table. It makes sense to associate the landscape configuration with that paragraph style because it clearly signifies the start of the content that you want to show as landscape.

You also insert a section break in the source document after each of those tables, because that will tell Doc-To-Help to end the custom configuration.

In Doc-To-Help you open the Project Styles dialog. In the **Paragraph Styles** section on the left, you add a new style called **TableTitle**, and you set the **Section configuration** to **Landscape**.



When you build the Manual target, the sections beginning with the TableTitle style are changed from portrait to landscape.

HOW TO SET THE SECTION CONFIGURATION ON A MANUAL TARGET

1. In Word, open the **Target** ribbon and click the **Target Properties** button. The Help Targets dialog opens.
2. On the left side of the dialog, select the **Manual** target.
3. In the **Advanced** section, set the **Section break** field to one of the following:
    - **Next page**
    - **Even page**
    - **Odd page**
4. In the **Section orientation** field, select one of the following:
    - **Inherit from template** This is the default option. If selected, a new section created by the "Section break" property will have the same page orientation as that used in the template.
    - **Portrait**
    - **Landscape**
5. Click **OK**.

Additional Details

Details	Pros	Issues	Deliverable File(s)	File Location
Can be online (PDF), hard-copy, or both.	**Hardcopy** Familiar format and navigation. **Online** Same familiar format, but no printing cost. Can create links to web-sites, video/audio files, etc. Searchable	**Hardcopy** Can become obsolete quickly. Printing and delivery costs. Customers may not work in an environment where they can access manual.	DOC and/or PDF	By default, the Manual folder of your project.

CHAPTER 3

Creating and Deleting Targets

In Doc-To-Help, targets are built to produce the final output of your project. You can generate several types of output with Doc-To-Help.

The Help Targets dialog is used to specify the targets that you would like to create and specify key settings for each output. You may create more than one output of a specific type (e.g., you could have two different HTML Help outputs, one for "Administrator HTML Help" and one for "Manager HTML Help"). Simply add those targets to the list and configure each as you wish. To make the content in each unique, use conditional text and variables. See the online Help or the *Documents, Templates, and Content Guide*.

HOW TO ADD A TARGET

1. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
2. In the box on the left, click **Add New Target**, and from the drop-down list, select the desired target type. The Add New Target dialog opens.
3. Enter the new target name and choose to either use the default target properties, or copy the properties from an existing target.
4. Click **OK**. The new target is added to the list. The target can be further customized by editing the fields.

After creating a target, you can build output for it. You can build output from the Doc-To-Help application or from Word. See "Building Targets" on page 63.

HOW TO DELETE A TARGET

1. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
2. In the box on the left, select the target you want to delete and click **Remove Target**.

CHAPTER 4

Editing Target Settings

Using the Help Targets dialog, you can apply all kinds of settings to a target that you have created. This includes renaming the target to reflect the nature of your project, setting attributes (choosing the conditions to include or exclude from the output), and much more. The settings that are available are different for each output type.

HOW TO EDIT TARGET SETTINGS

1. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
2. On the left side of the dialog, select the target whose settings you want to edit.
3. On the right side of the dialog, edit any of the fields, depending on the output type. For details on each field, see the online Help.
4. Click **OK**.

> **NOTE:** If you remove targets from your project in the Help Targets dialog , you will not see these targets when viewing Target drop-downs in Word (e.g., in the Contents window pane or in the Doc-To-Help Build panel).

CHAPTER 5

Transformations

Using the Transformation Wizard, you can insert boilerplate content or code into every topic in your targets. Examples include: headers, footers, or code needed for technology such as analytics.

You can add transformations to NetHelp, HTML Help, EPUB, JavaHelp, Eclipse Help, Microsoft Help Viewer, and Microsoft Help 2.0 Help targets.

Transformation files created with the wizard can be used in any project.

This chapter discusses the following:

Custom HTML5 Code

Custom HTML5 code can be defined for the following places in a topic:

- Inside the <head> element (e.g., scripts, CSS, metadata)
- At the beginning of topic text (i.e., above the topic title)
- At the end of topic text

How to Insert a Transformation

1. Select the **Target** ribbon.
2. Click the **Target Properties** button. The Help Targets dialog opens.
3. In the **Advanced** section, next to the **XML transformation** field, click  . The Transformation Wizard opens.
4. Enter your custom code in the appropriate section(s): <head> element, beginning of topic, or end of topic.
5. Click **Save**. You are prompted to name and save the file. The transformation file created (*.config) is stored at the root of your Doc-To-Help project folder. (This location is the default, but you can store it anywhere.)

   After you build your target, the code will be inserted in each topic of the target.

How to Reuse a Transformation File

If would like to reuse the transformation file in another project, do the following:

1. In Word, click the **Target Properties** button on the **Target** ribbon. The Help Targets dialog opens.
2. In the **Advanced** section, next to the **XML transformation** field, click  .
3. Navigate to the *.config file, select it, and click **Open**.

> ☆ EXAMPLE
>
> There is an example transformation named "timestamp.config" that will add timestamp text to each topic of a target. It is located at: **C:\Program Files (x86)\MadCap Software\DocToHelp\Transforms\Examples\Timestamp**

> ☆ EXAMPLE
>
> To insert keywords in the <head> element of each topic, add this to the first field of the Transformation Wizard:
>
> ```
> <meta name="keywords" content="help authoring tool, doc-to-help, doctohelp, online help"/>
> ```

> ☆ EXAMPLE
>
> To insert a distinctive header (green, bold, and italic) in each topic, add this to the middle field of the Transformation Wizard:
>
> ```
> 
> Draft Version 1.1, for Internal Review Only
> 
> ```

EXAMPLE

To insert a footer with a logo and hyperlink in each topic, add this to the last field of the Transformation Wizard:

```
<div style="width:90%">
    <hr style="color:#CCCCCC" />
    <div style="float: left">
        <img src-
        ="h-
        ttp://www.mad-
        capsoftware.com/newimages/company/icon.png" alt="MadCap
        Logo" />
    </div>
    <div style="float: right">
        MadCap Software<br />
        San Diego, CA<br />
        <a href="http://www.madcapsoftware.com" target="_
        blank">madcapsoftware.com</a>
    </div>
</div>
```

NOTE: Verify that the **Generate XHTML** check box is selected in the Help Targets dialog before building your target. It is selected by default in most targets.

NOTE: Transformations cannot be used with Manual targets.

NOTE: You can also add programmatic and XSLT transformations to your projects. See "Advanced XML Transformations" on the next page.

Advanced XML Transformations

Programmatic and XSLT transformations can be added to outputs to manipulate content. Programmatic transformations can be written in any .NET language, such as C# or VB.NET. This advanced feature gives you great flexibility, because it allows for post-build modification in virtually any imaginable custom way.

You can use the Transformation Wizard to insert boilerplate content or code into every topic in your targets.

> ☆ EXAMPLES
>
> Examples of advanced transformations include:
>
> - Replacing some text (tag) with variable text depending on a parameter specified in that tag.
> - The Expanding/Collapsing Sections feature in Doc-To-Help. See the online Help or the *Documents, Templates, and Content Guide*.

HOW TO USE A CUSTOM TRANSFORMATION

1. Select the **Target** ribbon, and click **Target Properties**. The Help Targets dialog opens.
2. In the **Advanced** section, next to the **XML transformation** field, click  .
3. Navigate to the file, select it, and click **Open**.
4. In the Help Targets dialog, click **OK** and build your target.

HOW TO CREATE A CUSTOM TRANSFORMATION

Refer to the notes below, and the two examples provided with Doc-To-Help:

- The expanding sections functionality is implemented as an XML transformation (a programmatic one). Its full source code can be found in \\Program Files [or Program Files (x86)]\MadCap Software\DocToHelp\Transforms\BuiltIn\Source\XMLInternal. (Don't modify these files; they are used in target compilation.)
- There is an example transformation named timestamp.config that will add timestamp text to each topic of a target. It is located at C:\Program Files (x86)\MadCap Software\MadCap DocToHelp 6\Tranforms\Examples\Timestamp.

NOTES ON CREATING XML TRANSFORMATIONS

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

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

- **<description>** Arbitrary string describing the transformation, for explanatory purpose only.
- **<assembly>** File name of the transformation assembly (including the .dll extension). For an XSLT transformation that assembly should always be C1D2HXMLInternal.dll. For programmatic transformations, it is a path to the file, not the file name. The path is relative to the configuration file (where transform assembly is specified).
- **<type>** Full class name in the assembly that implements the IXMLTransform interface. That is the class whose methods are called to perform the transformation. For an XSLT transformation that element should always be as follows: <type>C1.D2H.XMLTransform.XSLTTransform</type>
- **<params>** This element contains whatever elements the transformation may need as its parameters. All specific parameters will be passed to the transformation assembly when it is executed. For programmatic transformations, their interpretation is entirely dependent on the code implementing the transformation methods. XSLT transformations should be contained in one XSLT file that receives parameter values. The path to an XSL file is relative to the configuration file, so only the file name needs to be specified, for example: <xsl-file>C1D2HXMLInternal.xsl</xsl-file>.

NOTES ON CREATING PROGRAMMATIC TRANSFORMATIONS

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

- **void IXMLTransform.ReadParams(XmlNode node)**

  This method is called before executing the transformation, to initialize the transformation class with parameter values specified in the <params> tag in the configuration file. This method is called once in Doc-To-Help build, before this transformation is applied to any topics.

The 'node' parameter passed to this method contains the XmlDocument whose root element is the <params> element of the configuration file.

- void IXMLTransform.Execute(XmlDocument doc)

  This method executes the transformation, applies it to a topic. The 'doc' parameter is the topic XML before applying the transformation (all transformations preceding this transformation in the configuration file have already been applied to it. This method implementation modifies the XmlDocument passed to it.

- void IXMLTransform.CopyRequiredFiles(string targetPath)

  This method is used to copy any files that may be needed in the help target to support this transformation's functionality (these are usually resource files, such as graphics, scripts, style sheets, etc). If you don't need any files, leave this method empty. The 'targetPath' parameter contains the path to the TransformFiles subdirectory of the Help target directory (for example, \MyProject\MyHTMLHelpTarget\TransformFiles). This method implementation copies required files (if any) to that subdirectory (possibly creating subdirectories inside it, if needed).

CHAPTER 6

Building Targets

By building a target, Doc-To-Help can produce multiple types of outputs: NetHelp (including responsive), HTML Help, EPUB, Eclipse Help, JavaHelp, Manual, Microsoft Help Viewer, and Microsoft Help 2.0. See "Output Types" on page 9.

Target names and other details are customized using the Help Targets dialog. See "Editing Target Settings" on page 54.

This chapter discusses the following:

How to Build a Target

1. Save your Word source documents.
2. Select the **Doc-To-Help** ribbon. In the **Project** section, click **Build**. The Doc-To-Help Build panel opens.
3. From the **Active Target** drop-down in the local toolbar, select the target you want to build.

   > NOTE: When you *select* a target in the local toolbar, it does not affect the active target in the Doc-To-Help application and it does not affect other Word panels and other open Word source documents. However, when you *start building* a target, if you have other Word source documents open, the Target drop-down in those documents will synchronize to show the target that is being built.

4. In the local toolbar, click **Build** or **Rebuild**.
   - **Build** When you choose the Build button, Doc-To-Help recognizes all the source documents that have been edited since the last build and makes those changes in the output. This is the quickest way to build a project.
   - **Rebuild** When you choose the Rebuild button, Doc-To-Help deletes all of the content in the output folder of your target, then builds everything from scratch (not just source documents). If you make any global changes to your project (e.g., changing project settings, creating variables, glossary entries), you should do a rebuild. A rebuild will always take longer than a build, especially on large projects. You should always do a rebuild before reviews and releases.
5. When the target is done building, a dialog asks if you want to view the target. Click **Yes** to view it; otherwise click **No**. You can also view the target by clicking **View** in the Doc-To-Help Build panel's local toolbar.
6. (Optional) You can also do any of the following:
   - Use the **Cancel Build** button in the local toolbar to stop a build.
   - Click the **Open Build Log** link on the Build Log tab to view the build log.
   - Click the **Errors and Unresolved Links** tab to view and sort any errors that may have occurred during the build.

NOTE: If you are building a target in Word, you will not be able to open Doc-To-Help from Word until the build is completed.

NOTE: If you remove targets from your project in the Help Targets dialog , you will not see these targets when viewing Target drop-downs in Word (e.g., in the Contents window pane or in the Doc-To-Help Build panel).

EPUB Output

EPUB output will open in your default reader (such as Calibre) after it is built. To set the default EPUB reader in Doc-To-Help, go to **File > Doc-To-Help > Common Tasks > Options**. In the Options dialog, select **Viewers** and choose the **EPUB viewer executable**.

Manual Output

When you build a Manual target, you can build a PDF version at the same time. To do so, open the Help Targets dialog, choose the **Manual** target, and select the **Generate PDF** check box. After building the target, you can then open the **Home** ribbon and click the **View PDF** button. Projects will use Word's built-in PDF converter.

When you view the Word version of your Manual output, the resulting file will have a different Doc-To-Help ribbon than the source documents. In the target document, the ribbon will contain only the basic styles and tools necessary to make final adjustments to your manual before printing or conversion to PDF, such as the Cross-Reference button and the Margin Notes button.

> NOTE: When building output, you can schedule builds so that they run on a one-time, daily, or weekly basis. See "Scheduling Builds with the Build Scheduler" on page 72.

> NOTE: You can also build targets in batch mode using the command line (see "Building Targets in Batch Mode" on the next page) or using Windows Services (see "Building Targets Using Windows Services" on page 71).

> NOTE: You can edit Word source documents while building 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.

Building Targets in Batch Mode

You may build targets in batch mode from the command line through a special executable program, C1D2HBatch.exe.

If you need to start a build from Windows Services (such as the Task Scheduler or TFS Build), you should use a different executable, C1D2HAgent.exe. See "Building Targets Using Windows Services" on page 71.

Both of these executables are located in **C:\Program Files (x86)\MadCap Software\DocToHelp**.

COMMANDS

The commands are as follows:

Command	Description	Example
c1d2hbatch –build	Rebuilds the current target entirely.	c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide NetHelp" -p
c1d2hbatch –make	Updates the current target.	c1d2hbatch -make "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual"
c1d2hbatch –build or make -lLogFile (optional)	Saves the build log file with a specific name, rather than the default name, which is equal to the current date and time.	c1d2hbatch -build "c:\D2H Projects\StyleGuide\StyleGuide.d2h" "StyleGuide Manual" -lmyfilename.log Build log will be stored in the Temp\BuildLogs directory.

NOTE: There is currently only one flag allowed in 'flags': -p, enabling the output of progress messages.

ERROR AND LOG MESSAGES

All error and log messages are sent to the console standard output and have the following format:

- For errors aborting compilation:

  ```
  D2H:fatal error:[error code]: [message text]
  ```

  The [error code] is a numeric error code.

- For logged user errors, compilation continues and every log error is output to the Output window:

  ```
  D2H:log error document:[document name]
  ```

  ```
  D2H:log error topic:[topic title]
  ```

  ```
  D2H:log error message:[severity]:[message text]
  ```

  The [document name] and [topic title] can be empty, depending on the nature of the error.

  The [severity] will be listed as one of the following three strings: "note," "warning," or "error."

- A log message, for information only, shows what document is currently compiling, what action is being performed, and so on:

  ```
  D2H:log:[message text]
  ```

- A progress message appears only if the -p flag is present:

  ```
  D2H:progress:[done]/[total]
  ```

  Here [done] is an integer number indicating how many steps of the currently performed action have been completed so far.

  The [total] is an integer number indicating the total number of steps in the currently performed action.

- A progress completion message indicates that the current action has been completed, and it appears only if the -p flag is present:

  ```
  D2H:progress:done
  ```

- An unresolved links list is the output at the end of a successful build when the -build or -make commands are used. Every unresolved link is output to the Output window:

  ```
  D2H:unresolved link text:[link text]
  ```

  ```
  D2H:unresolved link style:[style name]
  ```

  ```
  D2H:unresolved link topic:[topic title]
  ```

  ```
  D2H:unresolved link document:[document name]
  ```

- For a successful build:

  ```
  D2H:succeeded
  ```

- For a failed build:

  ```
  D2H:failed
  ```

- The last message before Doc-To-Help batch mode exits:

  ```
  D2H:exit
  ```

NOTE: In Windows 7 and later, C1D2HBatch.exe must be run as administrator, or it will always throw an exception. If you run it directly, start it with "Run as Administrator," available in Windows 7/8 in the context menu of the C1D2HBatch.exe file. If you run it from a command line prompt, open the command line window with "Run as Administrator." If you run it from a script or a program, make sure the script or program runs as administrator.

Building Targets Using Windows Services

You can build Doc-To-Help targets using Windows Services such as the Task Scheduler or TFS Build. This is done using C1D2HBatch.exe and C1D2HAgent.exe, which can be found at C:\Program Files (x86)\MadCap Software\MadCap DocToHelp 6.

HOW TO BUILD TARGETS USING WINDOWS SERVICES

1. An interactive user should log on and launch C1D2HAgent.exe.

   C1D2HAgent.exe must be run on the same machine as the Windows Service. An interactive user has the rights to interact with the machine using the keyboard, mouse, etc. (Services are generally not interactive.)

2. Start C1D2HBatch.exe with the argument -buildagent.

   EXAMPLE

   "C:\Program Files (x86)\MadCap Software\MadCap DocToHelp 6\C1D2HBatch.exe" -build "C:\Users\[InteractiveUser]\Documents\My Doc-To-Help Projects\Samples\EmployeeHandbook\EmployeeHandbook.d2h" "Employee Handbook Manual" -buildagent

   This can be started by any Windows Service.

The interactive user who started C1D2HAgent.exe:

- Must be logged in when C1D2HBatch.exe is running.
- Must have read and write permissions for the project folder and its subfolders.

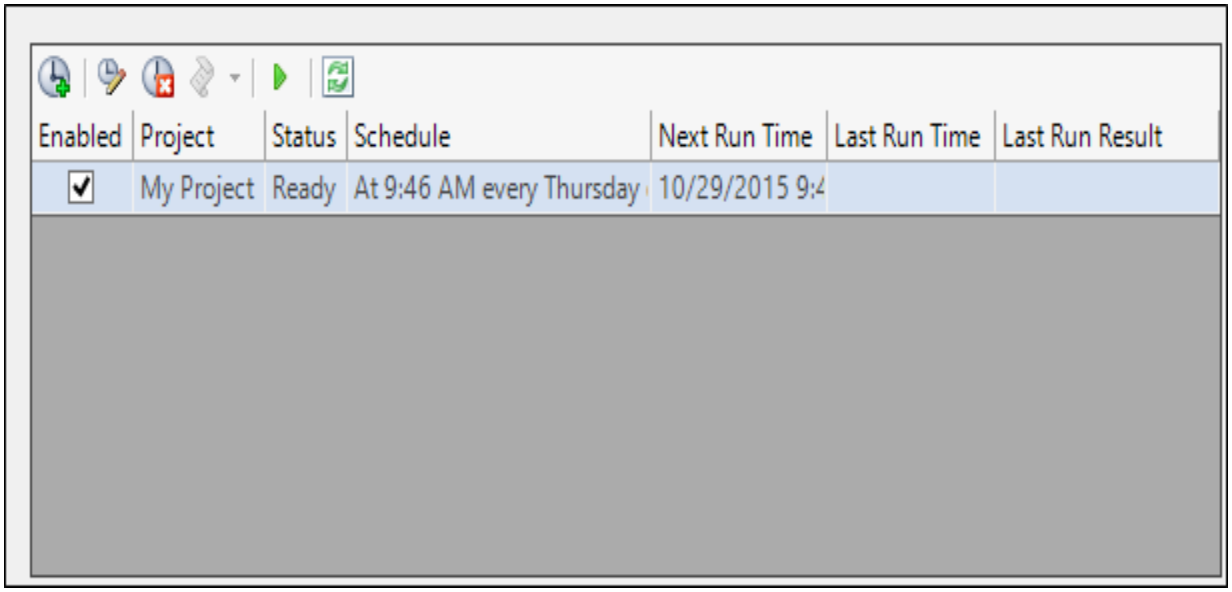
Scheduling Builds with the Build Scheduler

Using the Build Scheduler, you can schedule and monitor automatic builds of targets. One-time, daily, and weekly builds can be scheduled.

All of the scheduled builds created display in the Build Scheduler window. Only those with a check mark in the Enabled column will be run. The project name, status, build time, the time of the next and the last build, and the result of the last build are displayed for each scheduled build.

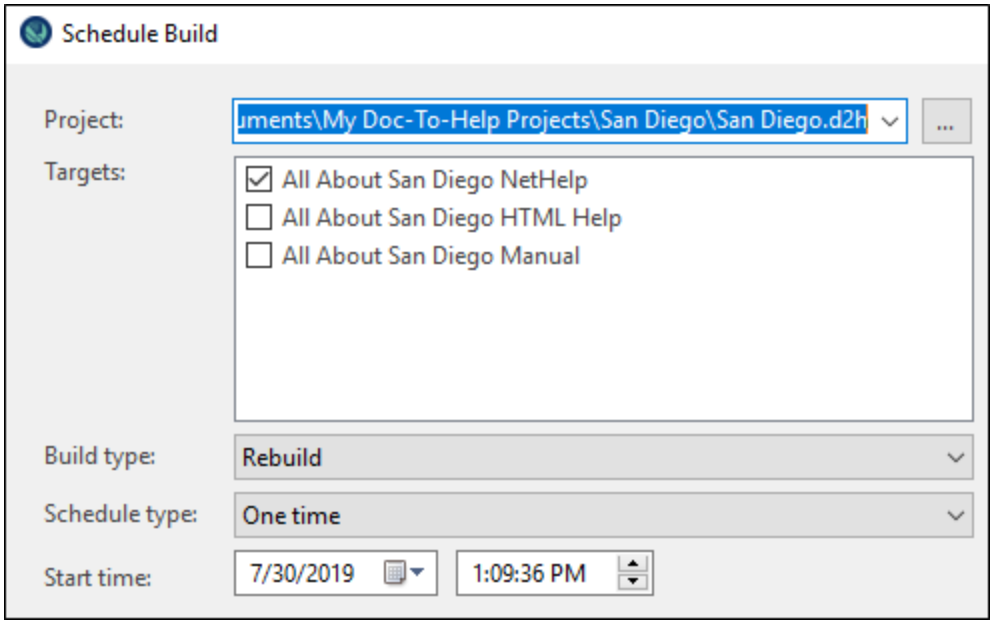
You can build all of the targets for a project from a single build schedule.

The toolbar in the Build Scheduler lets you create, edit, and delete builds; view build logs; run any build on the spot; and refresh the window.



HOW TO SCHEDULE A BUILD

1. Select **File > Doc-To-Help**.
2. Select **Common Tasks**, then click the **Build Scheduler** button. The Build Scheduler opens.
3. In the local toolbar, click  . The Schedule Build dialog opens.



   You can open the Schedule Build dialog directly from a Doc-To-Help project. Open the project and from the **Home** tab, click the drop-down arrow at the bottom of the **Build** or **Rebuild** buttons. Choose **Schedule Build** from the options.

4. In the **Project** field, click  and choose the project.
5. Select the check box next to each target in that project you would like to build.
6. Choose the **Build type**. For an explanation of the difference between Build and Rebuild, see "Building Targets" on page 63.
7. Choose the **Schedule type**. You have the choice of **One time**, **Daily**, and **Weekly**.
8. Choose the **Start time** and **date**. If you have chosen a Daily or Weekly build, you can then choose when the build should recur.
9. Click **OK**.

NOTE: If you choose to schedule daily builds, the default is for the targets to build every day at the chosen time (**Recur every 1 days**). If you would prefer to choose the exact days your builds will occur (e.g., every weeknight, choose the **Schedule type** of **Daily** and select the days of the week).

OTHER TASKS

After you create a new task using the previous set of steps, you can also do the following in the Build Scheduler:

- **Edit** To edit a build schedule, select it, and click the **Edit the Selected Task** button.
- **Delete Task** To delete a task, select it and click the **Delete Selected Task** button.
- **View Build Log** After your build tasks are run, the log files are stored within your project's Temp\BuildLogs folder. Every target will have a separate folder, and the log file names will have the "schedule_" prefix (e.g., "schedule_2010-11-15_16-60-50.log"). To view the build log of a task, select it and click the **View Build Log of Selected Task** button.
- **Run Task** To run a task immediately, select it and click the **Run Selected Task** button.
- **Refresh List** Click the **Refresh Task Status** button to refresh the task list.

NOTE: When the scheduled build for a project is building, you cannot open that project. If you open a project and it is scheduled to build within that time frame, the scheduled build will not begin (it will fail). You can open, edit, and build any other Doc-To-Help project while the scheduled build of another project is building.

NOTE: Windows Task Scheduler must be started for Doc-To-Help's Build Scheduler to run. To check the status of the Windows Build Scheduler:

Windows 7 and later: Click **Start > Control Panel > System and Security > Administrative Tools > Scheduled Tasks**.

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*