MadCap Software

Project Creation Guide

Flare 10

Copyright 2014 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
7777 Fay Avenue
La Jolla, California 92037
858-320-0387
www.madcapsoftware.com

THIS GUIDE WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1 Starting Projects 5

Ways to Start Projects 6
Files 7
Creating a New Project 11
Importing a RoboHelp Project 17
Importing an HTML Help Project 18
Importing CHM Files 19
Importing DITA Files 20
Importing FrameMaker Files 29
Importing HTML Files 74
Importing Word Files 82
Importing from Source Control 113

CHAPTER 2 Merging Projects 115

Linking to CHM Files 116
Merging Flare Projects Using Targets 117
Merging Output from DNH Projects 121
Merging Output from Flare WebHelp 128
Merging Output from HTML Help Projects (CHM Files) 135
Merging Output at Runtime Using HTML5 139
Merging Output at Runtime Using WebHelp Plus 147
Displaying Merged Navigation in HTML Help Child Outputs 155

CHAPTER 3 Exporting Projects 157

Adding an Export Project File 158
Exporting Projects 159
Exporting Projects Using the Command Line 184

CHAPTER 4 Other Project Tasks 189
Opening a Project 190
Closing Projects 191
Deleting Projects 192
Zipping Projects 193
APPENDIX PDF Guides 195

CHAPTER 1

Starting Projects

The first step in developing a project after you launch Flare is to start a project.

This chapter discusses the following:

Ways to Start Projects 6
Files 7
Creating a New Project 11
Importing a RoboHelp Project 17
Importing an HTML Help Project 18
Importing CHM Files 19
Importing DITA Files 20
Importing FrameMaker Files 29
Importing HTML Files 74
Importing Word Files 82
Importing from Source Control 113

Ways to Start Projects

You can start a project in one of several ways.

- **Create a New Project—Standard Method** Use the Start New Project Wizard to create a new project, basing it on a factory template or one that you have created. See "Creating a New Project" on page 11.
- **Import RoboHelp Project** Create a new project by importing a RoboHelp project (MPJ or XPJ file). See "Importing a RoboHelp Project" on page 17.
- **Import HTML Help Project (HHP)** Create a new project by importing an HTML Help project (HHP file). See "Importing an HTML Help Project" on page 18.
- **Import HTML Help File (CHM)** Create a new project by importing an HTML Help file (CHM file). See "Importing CHM Files" on page 19.
- **Import Microsoft Word** Create a project by importing Microsoft Word documents. In addition, after you create a project (using any method), you can create new topics in that project by importing Word documents. See "Importing Word Files" on page 82.
- **Import Adobe FrameMaker** Create a project by importing Adobe FrameMaker documents. Because you can import the source FrameMaker BOOK and FM files (rather than just MIF files), Flare has full access to FrameMaker variables, conditionals, autonumbering, and so on. This means that those features are converted to Flare seamlessly. In addition, after you create a project (using any method), you can create new topics in that project by importing FrameMaker documents. See "Importing FrameMaker Files" on page 29.
- **Import DITA Content** Create a project by importing DITA file content. When you do this, the DITA elements are converted to HTML tags (and if you later generate DITA output, they are then converted back to DITA elements). In addition, after you create a project (using any method), you can create new topics in that project by importing DITA file content. See "Importing DITA Files" on page 20.
- **Import From Source Control** Import a Flare project that has been added to a source control application, such as Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, or Apache Subversion. See "Importing from Source Control" on page 113.
- **Open a Project** Start a project by opening an existing one. See "Opening a Project" on page 190.

Files

When you create a new Flare project, a file with an .flprj extension is automatically generated (e.g., MyProject.flprj). This is the main file for the project and is stored at the root level of the project folder in Windows. You can open the project in Flare by double-clicking this file.

In addition to the main FLPRJ file, there are numerous other files that are part of your Flare project. All Flare files are separate XML documents—topics, TOCs, browse sequences, page layouts, targets, skins, snippets, glossaries, destinations, condition tag sets, variable sets, and more. This means that Flare projects are completely open, transparent, and accessible. When you look at the location in Windows where you stored your Flare project, you will see the main FLPRJ file, plus a handful of folders that hold the other files described below. You will always see Content and Project folders (which hold content and project files, respectively). In addition to those, you might see the following folders, depending on the actions you take in your project: Analyzer, FileSync, Output, SourceControl.

- **Content Files** These files are created as you add new topics or resources such as images, stylesheets, or snippets within the project. These files are stored in the Content Explorer.
  - **Topic Files** These are the main elements in a project that are used to hold text and other content. These XHTML files use the .htm extension.

    E X A M P L E

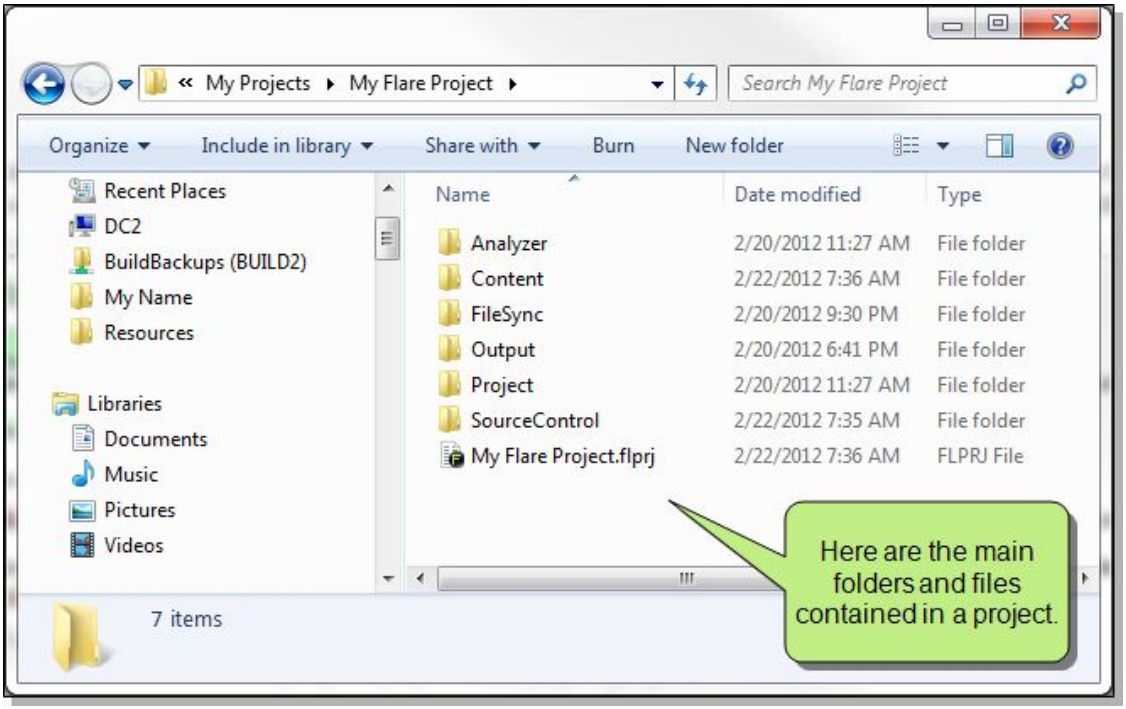
    Let's say you are writing a manual to teach people how to use expense reports at work. You want to create one topic called "Where to Find Expense Reports" and another called "How to Complete an Expense Report." After you create the topics, Flare will save one XHTML file for each of the topics (e.g., "Where to Find Expense Reports.htm" and "How to Complete an Expense Report.htm").

  - **Image Files** You can insert images into content files (e.g., topics, snippets). Flare supports the following types of raster and vector image files: BMP, EMF, EPS, EXPS, GIF, HDP, JPG, JPEG, PNG, PS, SVG, SWF, TIF, TIFF, WDP, WMF, XAML, XPS. When you insert a picture into your project, the image file is copied to your project and stored by default in the Resources\Images subfolder in the Content Explorer.
  - **Multimedia Files** You can insert audio and video multimedia files (Flash, Windows Media, Quicktime) into topics or snippets.When you insert a multimedia file into your project, the audio or video file is copied to your project and stored by default in the Resources\Multimedia subfolder in the Content Explorer.

- **Page Layout and Master Page Files** You can use page layouts and master pages to configure pages for print output (e.g., page size, orientation, headers, footers, page numbers). Page layouts allow for easy configuration through the use of content frames, a snap-to grid, dragging and dropping, alignment features, and more. Although they are similar to master pages, page layouts are more flexible and easier to use. Master pages, however, will continue to be available in Flare for users who have implemented them into past projects; plus, master pages are still useful for online output. The general rule is that page layouts are best for *print-based output* (when possible), and master pages are best for automatically adding headers and footers in multiple topics for *online output*. Neither master pages nor page layouts are used in EPUB targets. A page layout has an .flpgl extension, and a master page has an .flmsp extension. The traditional location to store a page layout in the Content Explorer is in the Resources\PageLayouts folder, and the traditional location for master pages is in the Resources\MasterPages folder. However, you can store them anywhere in the Content Explorer that you like.
- **Snippet Files** When you create and insert a snippet (re-usable, formatted content) into your project, a snippet file (.flsnp extension) is copied to your project and stored by default in the Resources\Snippets subfolder in the Content Explorer.
- **Stylesheet Files** You can use cascading stylesheets (CSS files) when formatting your topic content and tables. The traditional location to store stylesheet files in the Content Explorer is in the Resources folder. However, you can store these files anywhere in the Content Explorer that you like.

- **Project Files** These files are created as you add certain features to the project and develop outputs (targets). When you develop a project, you probably want it to have certain features and characteristics. For example, you might want to include variables or a glossary. When you add and develop elements such as these in Flare, specific project files are created for you so that the final output knows how to look and behave. One of the greatest benefits of Flare is that these files are independent, rather than being built into the main FLPRJ file or part of some proprietary database file. This means that they can easily be used in other projects, shared with other authors, and modified with any XML editor. These files are stored in the Project Organizer.
- **Analyzer Files** These files are automatically created with each Flare project if you also have Microsoft SQL Server Compact installed. They are database files used behind the scenes when you use Analyzer features in Flare. You do not need to do anything with these files.
- **FileSync Files** These files are automatically created when you create mappings between external files and copies in your project (e.g., if you are working with external resources or SharePoint integration). You do not need to do anything with these files.
- **Output files** These files are created when you build the final output for a Flare project. What is the ultimate goal of your Flare project? It is to produce an end result that you can provide to your users. This might be a single Help system, a group of Help systems, printed documentation, or a combination of these. That is the purpose of output files. They are automatically created when you build a project in Flare. You can then distribute these files to end users.
- **Source Control Files** These are files created by Flare if you bind your project to source control.

You can easily see any of these files in Windows Explorer by looking in the location where you told Flare to create your project. Unless you specify a different location when creating a project, your files are located in your My Document\My Projects folder. (Flare creates the My Projects subfolder for you when you install the program.)



Note: Much of the work in creating project files is done behind the scenes and you do not need to worry about it. As an author, your primary focus is to learn how to use the Flare interface and write the content for your project.

Note: Although it is possible to open any Flare files in a third-party editor, it recommended that you avoid editing these files in certain programs while Flare is running. For example, opening a stylesheet in Notepad is not an issue. But you might experience problems if you open a topic or snippet in Microsoft Word and edit it while Flare is running.

Creating a New Project

The first step in developing a project after you launch Flare is to start a project. You can start a project in one of several ways, such as by importing other projects or files. Following are the steps for creating a new project using the Start New Project Wizard.

HOW TO CREATE A NEW PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Start Page** Select the **Templates** tab. Use the slideshow to find a template that you want to use as a starting point for your project. When you find one, click it.
   - **Ribbon** Select **File>New Project**.

     You can use the Options dialog to switch between ribbons and the classic tool strip layout. For more information see the online Help.

     Keep in mind that the smaller the application window becomes, the more the options in a ribbon shrink. Therefore, you might only see a small icon instead of text, or you might see only a section name displayed with a down arrow to access the options in it. You can hover over small icons to see tooltips that describe them. You can also enlarge the application window or click one of the section drop-downs in the ribbon to locate a hidden feature.
   - **Menu** Select **File>New Project**.
   - **Standard Toolbar** Click .

     To open the Standard toolbar from the menu view, you can select **View>Toolbars>Standard**.

   The Start New Project Wizard opens.
2. In the **Project name** field, type an appropriate name for your project.
3. By default, a path to the Documents\My Projects folder on your hard drive is entered in the **Project folder** field. (Flare creates the My Projects folder when you install the program.) All subfolders and files related to the project will be placed in this folder as you work on the project. Continue with the next step, unless you want to have your project files placed in a different folder. If so, do the following.
   a. Click . The Browse For Folder dialog opens.
   b. Navigate to the folder you want, select it, and click **OK**.

   Note: In some operating systems, the folder is called "My Documents" instead of "Documents."
4. In the **Language** field select the language that you want to use for the project.

5. (Optional) If you want to integrate the new Flare project with a source control application (such as Microsoft Visual SourceSafe), select **Bind to Source Control**.
6. Click **Next**.
7. (Optional) If you have selected the "Bind to Source Control" option, click **Bind Project**. Then in the Bind Project dialog, complete the fields, depending on the source control application being used. When you are finished, click **Next**.

   IF USING TEAM FOUNDATION SERVER

   a. From the drop-down, select **Microsoft Team Foundation Server**.
   b. In the **Server** field, enter the name of the computer or the IP address of the server. You can also click Browse to select a "Team Project Collection." If you click this button, the Select Team Foundation Server Project Collection dialog opens, and you can do the following.

      i. To add a server, click . The Add Team Foundation Server dialog opens.
      ii. Enter the name or URL of the server.
      iii. Enter the path and port number.
      iv. Select the protocol (HTTP or HTTPS).

      Note: You may need to obtain this information from your system administrator. Click **OK** in the dialogs until you return to the main wizard page.

      Note: If you run into issues in this dialog, chances are your user permissions do not have access to all of the collections on the TFS server. Please try to enter in the server/collection path manually.

   c. Next to the **Team Project** field, click **Browse**. If the Log In dialog opens, complete the **User name** and **Password** fields.
   d. Click on the Team Foundation project to which you want to bind the Flare project.
   e. Click **OK**.
   f. (Optional) In the **Comment** field, you can enter any internal comments that you would like.
   g. (Optional) If you want the files in the project to be checked out when you are finished, click **Keep files checked out**.
   h. In the Bind Project dialog, click **OK**.

IF USING VISUAL SOURCESAFE

a. From the drop-down, select **Microsoft Visual SourceSafe**.
b. Next to the Database field, click **Browse**.
c. Find the appropriate SourceSafe database and double-click the **INI** file.
d. Next to the **Database Folder** field, click **Browse**. If the Log In dialog opens, complete the **User name** and **Password** fields.
e. Find and click on the folder for the new project files in Source Safe.
f. Click **OK**.
g. (Optional) In the **Comment** field, you can enter any internal comments that you would like.
h. (Optional) If you want the files in the project to be checked out when you are finished, click **Keep files checked out**.
i. In the Bind Project dialog, click **OK**.

IF USING SUBVERSION

a. From the drop-down, select **Subversion**.
b. In the **Server** field, enter the IP address.
c. Next to the **Project Path** field, click **Browse**.
d. Click on the Subversion folder to which you want to bind the Flare project.
e. Click **OK**.
f. (Optional) In the **Comment** field, you can enter any internal comments that you would like.
g. (Optional) If you want the files in the project to be checked out when you are finished, click **Keep files checked out**.
h. In the Bind Project dialog, click **OK**.
i. If the Log In dialog opens, complete the **User name** and **Password** fields and click **OK**.

IF USING PERFORCE

a. From the drop-down, select **Perforce**.
b. In the **Server** field, enter the server address.
c. Next to the **User** field, click **Browse**. The Select User dialog opens.
d. Select a user and click **OK**. (Users are set up separately in Perforce.)
e. Next to the **Stream** field, click **New**.

The Login dialog opens if you have not logged in before. If it does, enter the password for the user and click **OK**.

The New Perforce Stream dialog opens.

f. Next to the **Placeholder** field, click **Browse**. The Browse Depots dialog opens.

g. Select a depot. This is essentially a folder or directory in Perforce that contains multiple streams, one of which will be your Flare files.

h. Click **OK**.

i. (Optional) In the New Perforce Stream dialog, you can complete any of the other fields as necessary. For information about these fields, please refer to the documentation provided with Perforce.

j. Click **OK**.

k. (Optional) In the **Comment** field, you can enter any internal comments that you would like.

l. (Optional) If you want the files in the project to be checked out when you are finished, click **Keep files checked out**.

m. In the Bind Project dialog, click **OK**.

IF USING ANOTHER SOURCE CONTROL APPLICATION

a. From the drop-down, select **Third Party Plug-in**.

b. In the **Provider** field, select the appropriate source control program from the list.

c. (Optional) You might be able to click the Advanced button to enter more options. This depends on whether the provider has enabled advanced options; if not, the button is disabled. The advanced options available to you are different for each provider.

d. Next to the **Project Name** field, click **Browse**. If the Log In dialog opens, complete the **User name** and **Password** fields.

e. Find and select the project to which you want to bind the Flare project.

f. Click **OK**.

g. (Optional) In the **Comment** field, you can enter any internal comments that you would like.

h. (Optional) If you want the files in the project to be checked out when you are finished, click **Keep files checked out**.

i. In the Bind Project dialog, click **OK**.

> **Note:** If you have elected to bind the project to source control but do not complete the source control fields, the Finish button in the wizard is disabled.

8. If you used the Start Page to choose a template, it is already selected for you here and you can continue to the next page of the wizard. If you did not use the Start Page, do the following.

   In the **Source** area select one of the following:

   - **New from template** This lets you choose either the factory template file or one of your own customized template files as a starting point. The new file will take on all of the settings contained in the template. If you want to use the factory template provided by Flare, expand the **Factory Templates** folder and click on a template file. If you want to use your own customized template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.

     > **Note:** In some dialogs and wizards you can click the **Manage Templates** button if you want to open the Template Manager. This lets you manage any of your template files (e.g., add new templates, enter descriptions for templates). For more information see the online Help.

   - **New from existing** This lets you choose an existing file of the same type as a starting point for your new file. As with template files, your new file will take on all of the settings contained in the file you select. To use this option, click the browse button ..., use the Open File dialog to find a file, and double-click it.
   - **New from import** This lets you choose legacy files (e.g., Word, FrameMaker, HTML, DITA, HTML Help, RoboHelp) to import. If you select this option and click **Next**, a list of import options is displayed. You can select the file type, click **Next** again, and complete the options in the new wizard that opens, instead of completing the rest of the steps below. For more information on each of these import wizards and the steps to complete them, see any of the following.
     - "Importing Word Files" on page 82
     - "Importing FrameMaker Files" on page 29
     - "Importing HTML Files" on page 74
     - "Importing DITA Files" on page 20
     - "Importing an HTML Help Project" on page 18
     - "Importing CHM Files" on page 19
     - "Importing a RoboHelp Project" on page 17

9. Click **Next**.

10. From the **Available Targets** field, select the primary target for your project.
11. Click **Finish**.

Importing a RoboHelp Project

Following are the steps for creating a project by importing a RoboHelp project.

HOW TO IMPORT A ROBOHELP PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>RoboHelp Project**.
   - **Menu** Select **File>Import Project>RoboHelp Project**.
2. In the dialog that opens, browse for and double-click the RoboHelp project file (MPJ or XPJ file) to be imported. The Import Project Wizard opens.
3. In the **Project name** field, type a name for the new Flare project that will be created after you import the RoboHelp project.
4. In the **Project folder** field, either accept the default location for the new project or click ... to browse for and select a folder.
5. Click **Next**.
6. (Optional) Select **Convert all topics at once** if you want Flare to immediately convert all of your topic files to XHTML.

   If you remove the check mark from the box, Flare imports the topic files as they are. When you try to open an imported topic in Flare, a message asks if you want to convert it to XML. Also, if this option is not selected, Flare will not import index keywords from the source files.
7. (Optional) Select **Convert inline formatting to CSS styles** if you want Flare to create new styles based on any "local" formatting that exists in the RoboHelp project files.

   E X A M P L E

   If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.
8. Click **Next**.
9. Select a language for the project.
10. Click **Finish**. A message tells you that the project was converted successfully and will be opened.
11. Click **OK**.

Importing an HTML Help Project

Following are the steps for creating a project by importing an HTML Help project (HHP file).

HOW TO IMPORT AN HTML HELP PROJECT (HHP)

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>HTML Help Project (HHP)**.
   - **Menu** Select **File>Import Project>HTML Help Project (HHP)**.
2. In the dialog that opens, browse for and double-click the HTML Help file (HHP file) to be imported. The Import Project Wizard opens.
3. In the **Project name** field, type a name for the new Flare project that will be created after you import the HTML Help project.
4. In the **Project folder** field, either accept the default location for the new project or click ... to browse for and select a folder.
5. Click **Next**.
6. (Optional) Select **Convert all topics at once** if you want Flare to immediately convert all of your topic files to XHTML.

   If you remove the check mark from the box, Flare imports the topic files as they are. When you try to open an imported topic in Flare, a message asks if you want to convert it to XHTML. Also, if this option is not selected, Flare will not import index keywords from the source files.
7. (Optional) Select **Convert inline formatting to CSS styles** if you want Flare to create new styles based on any "local" formatting that exists in the HTML Help project files.

   E X A M P L E

   If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.
8. Click **Next**.
9. Select a language for the project.
10. Click **Finish**. A message tells you that the project was converted successfully and will be opened.
11. Click **OK**.

Importing CHM Files

Following are the steps for creating a project by importing an HTML Help (CHM) file.

HOW TO IMPORT AN HTML HELP (CHM) FILE

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>HTML Help File (CHM)**.
   - **Menu** Select **File>Import Project>HTML Help File (CHM)**.
2. In the dialog that opens, browse for and double-click the CHM file to be imported. The Import Project Wizard opens.
3. Click **Next**.
4. In the **Project name** field, type a name for the new Flare project that will be created after you import the CHM file.
5. In the **Project folder** field, either accept the default location for the new project or click ... to find and select a folder.
6. Click **Next**.
7. (Optional) Select **Convert all topics at once** if you want Flare to immediately convert all of your topic files to XHTML.

   If you remove the check mark from the box, Flare imports the topic files as they are. When you try to open an imported topic in Flare, a message asks if you want to convert it to XHTML. Also, if this option is not selected, Flare will not import index keywords from the source files.
8. (Optional) Select **Convert inline formatting to CSS styles** if you want Flare to create new styles based on any "local" formatting that exists in the HTML Help file.

   E X A M P L E

   If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.
9. Click **Next**.
10. Select a language for the project.
11. Click **Finish**. A message tells you that the project was converted successfully and will be opened.
12. Click **OK**.

Importing DITA Files

Darwin Information Typing Architecture (DITA) file content is supported in Flare. DITA is an XML-based markup language with its own schema for authoring, producing, and delivering technical information. It is a standard of the Organization for the Advancement of Structured Information Standards (OASIS), and it consists of a set of design principles for creating "information-typed" modules at a topic level and for using that content in various delivery modes.

You can import DITA files in the following ways:

- Import Into a New Project
- Import Into an Existing Project (From the Wizard)
- Import Into an Existing Project (From the Import Editor)

HOW TO IMPORT DITA FILES INTO A NEW PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>DITA Document Set** or **Project>Import>DITA Document Set**.
   - **Menu** Select **File>Import Project>DITA Document Set**.

   The Import DITA Wizard opens.
2. Select **Import into a new project**.
3. In the **Project name** field, type a name for the new Flare project that will be created after you perform the import.
4. In the **Project folder** field, either accept the default location for the new project or click ... to find and select a folder.
5. From the **Output Type** field, select the primary target for your project.
6. Click **Next**.
7. Click .
8. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DITA or DITAMAP files.

   Note: You cannot select multiple DITAMAP files from different folders during the same import process. You must first import the DITAMAP file(s) from one folder. Then in the project that is created as a result, you can import the DITAMAP file(s) from a second folder.

9. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original DITA files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original DITA files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

10. Click **Next**.
11. You can use the following options as necessary:
    - **Import all Content files to one folder** Select this option if you want all of the imported DITA file content to be placed in just one folder in the Content Explorer.
    - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your DITA source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import DITA files when you attempt to build output.
    - **Preserve ID attributes for elements** Every element inside a DITA file has an ID. This ID is not needed in Flare. However, if you intend to send your output back out to DITA at any point, you can select this option to make sure the ID is preserved.
12. Click **Next**.

13. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your DITA files.

    - **Conversion Styles** This opens the DITA Import Styles Editor, which lets you specify how to convert each property of the DITA elements. If you do not enter a property value, the value from the DITA file is used. If you enter a property value, it overrides the value from the DITA file. You can also use the dialog to import and export styles.

      > **Note:** When you import content from DITA files, there is a one-to-one conversion that occurs. For each DITA element in your file, a style class is created in Flare. For example, let's say you have a paragraph-level DITA element called "topictitle," after you import the file, a style class called "h1.topictitle" might be created in Flare. Or if you have a character-level DITA element called "cmdname," after you import the file, a style class called "span.cmdname" might be created as a result in Flare. If necessary, you can later edit those style classes in Flare. If you generate DITA output from your project, the style classes are converted back to DITA elements.

    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
14. Click **Next**.
15. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
16. When you are finished previewing the files to be created, click **Accept**. The files are imported and the project is created.

HOW TO DITA FILES INTO AN EXISTING PROJECT—VIA THE WIZARD

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **Project>Import>DITA Document Set**.
   - **Menu** Select **File>Import Project>DITA Document Set**.

   The Import DITA Wizard opens.
2. Select **Import into this project**.
3. Click **Next**.
4. Click .
5. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DITA or DITAMAP files.

   Note: You cannot select multiple DITAMAP files from different folders during the same import process. You must first import the DITAMAP file(s) from one folder. Then in the project that is created as a result, you can import the DITAMAP file(s) from a second folder.

6. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original DITA files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original DITA files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

7. Click **Next**.
8. You can use the following options as necessary:
   - **Import all Content files to one folder** Select this option if you want all of the imported DITA file content to be placed in just one folder in the Content Explorer.
   - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your DITA source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import DITA files when you attempt to build output.
   - **Preserve ID attributes for elements** Every element inside a DITA file has an ID. This ID is not needed in Flare. However, if you intend to send your output back out to DITA at any point, you can select this option to make sure the ID is preserved.
9. Click **Next**.
10. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your DITA files.
    - **Conversion Styles** This opens the DITA Import Styles Editor, which lets you specify how to convert each property of the DITA elements. If you do not enter a property value, the value from the DITA file is used. If you enter a property value, it overrides the value from the DITA file. You can also use the dialog to import and export styles.

> **Note:** When you import content from DITA files, there is a one-to-one conversion that occurs. For each DITA element in your file, a style class is created in Flare. For example, let's say you have a paragraph-level DITA element called "topictitle," after you import the file, a style class called "h1.topictitle" might be created in Flare. Or if you have a character-level DITA element

called "cmdname," after you import the file, a style class called "span.cmdname" might be created as a result in Flare. If necessary, you can later edit those style classes in Flare. If you generate DITA output from your project, the style classes are converted back to DITA elements.

» **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.

11. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
12. When you are finished previewing the files to be created, click **Accept**.

HOW TO IMPORT DITA FILES INTO AN EXISTING PROJECT—VIA THE IMPORT EDITOR

1. Do one of the following:

   IF YOU DO NOT YET HAVE A DITA IMPORT FILE

   a. Complete the steps for adding an import file.

   b. Open the file in the DITA Import Editor.

   IF YOU ALREADY HAVE A DITA IMPORT FILE

   a. In the Project Organizer, expand the **Imports** subfolder.

   b. Double-click the appropriate import file. The file opens in the DITA Import Editor to the right.

2. Select the **Source Files** tab.

3. Click .

4. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DITA or DITAMAP files.

   Note: You cannot select multiple DITAMAP files from different folders during the same import process. You must first import the DITAMAP file(s) from one folder. Then in the project that is created as a result, you can import the DITAMAP file(s) from a second folder.

5. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original DITA files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original DITA files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.

	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

6. Select the **Options** tab.
7. You can use the following options as necessary:
   - **Import all Content files to one folder** Select this option if you want all of the imported DITA file content to be placed in just one folder in the Content Explorer.
   - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your DITA source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import DITA files when you attempt to build output.
   - **Preserve ID attributes for elements** Every element inside a DITA file has an ID. This ID is not needed in Flare. However, if you intend to send your output back out to DITA at any point, you can select this option to make sure the ID is preserved.
8. Select the **Stylesheet** tab.
9. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your DITA files.
   - **Conversion Styles** This opens the DITA Import Styles Editor, which lets you specify how to convert each property of the DITA elements. If you do not enter a property value, the value from the DITA file is used. If you enter a property value, it overrides the value from the DITA file. You can also use the dialog to import and export styles.

> **Note:** When you import content from DITA files, there is a one-to-one conversion that occurs. For each DITA element in your file, a style class is created in Flare. For example, let's say you

have a paragraph-level DITA element called "topictitle," after you import the file, a style class called "h1.topictitle" might be created in Flare. Or if you have a character-level DITA element called "cmdname," after you import the file, a style class called "span.cmdname" might be created as a result in Flare. If necessary, you can later edit those style classes in Flare. If you generate DITA output from your project, the style classes are converted back to DITA elements.

» **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.

10. In the local toolbar click **Import** (if this is the first time importing files) or **Reimport** (if files have been imported previously). The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
11. When you are finished previewing the files to be created, click **Accept**.

Note: The Imported Files tab displays the files that have been imported previously.

Importing FrameMaker Files

You can import Adobe FrameMaker files in the following ways:

- Import Into a New Project
- Import Into an Existing Project (From the Wizard)
- Import Into an Existing Project (From the Import Editor)

Note: Before diving in to the import process, it is recommended that you first download and review the *Transition from FrameMaker Guide*. This guide provides tips for making sure the FrameMaker import process goes as smoothly as possible. See "PDF Guides" on page 195.

HOW TO IMPORT FRAMEMAKER FILES INTO A NEW PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>FrameMaker Documents** or **Project>Import>FrameMaker Documents**.
   - **Menu** Select **File>Import Project>FrameMaker Documents**.

   The Import FrameMaker Documents Wizard opens.
2. Select **Import into a new project**.
3. In the **Project name** field, type a name for the new Flare project that will be created after you perform the import.
4. In the **Project folder** field, either accept the default location for the new project or click ... to find and select a folder.
5. From the **Output Type** field, select the primary target for your project.
6. Click **Next**.
7. Click .
8. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select BOOK, FM, or MIF files.

   Tip: When possible, it is recommended that you select a Adobe FrameMaker BOOK file for import and let Flarelocate and import all the associated document files within the Adobe FrameMaker book.

9. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original FrameMaker documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original FrameMaker documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

10. Click **Next**. The FrameMaker documents are scanned and the next page of the wizard opens. The paragraph styles used in the FrameMaker documents are shown on the left side of the wizard page ("Used FrameMaker Styles").

Note: You can skip the rest of the wizard pages by clicking **Finish**.

11. If you want Flare to split the FrameMaker documents into smaller topics based on any of the styles shown on the left side of the page ("Used FrameMaker Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

E X A M P L E

If you have a style called "Heading 2" in your FrameMaker documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

12. Click **Next**.
13. You can use the following options as necessary:
    - **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
    - **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
    - **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a heading in each topic that is connected to another topic included in the split. That text will be used in the link instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.
        - **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.
    - **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
        - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
    - **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the FrameMaker documents without any content.
        - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
    - **Anchored Frames With Images** You can use this area to specify how Flare should handle anchored frames that contain images as well as other content (e.g., text callouts).
        - **Generate Images Without Callouts** If the anchored frame contains an image along with callout

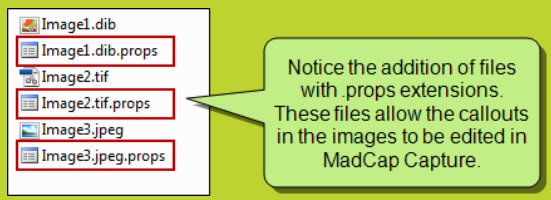
text, the original image is imported *without* the callout text. You might select this option if you have resized the image in FrameMaker. With this option, the imported image is likely to be of a higher quality than it would be otherwise. You can then add a callout to the image once it is inside Flare.

» **Generate Images With Callouts** If the anchored frame contains an image along with callout text, Flare creates a PROPS (i.e., properties) file along with the image file when that document is imported. This means that you can open those image files in MadCap Capture to edit those callouts after the import process is completed.

E X A M P L E

If you import anchored frame images and tell Flare not to include the callouts, your imported image files will look something like this in Windows Explorer.

On the other hand, if you import the same images and tell Flare to include the callouts, it will look something like this in Windows Explorer.



Note: Some FrameMaker elements—such as arcs and nested frames—are not supported with this option.

- **Generate Flattened Images** If the anchored frame contains an image along with callout text, a new flattened image will be created as a result. The callout is included, but you cannot edit it.

- **Preserve Image Size** This option affects how the size of imported images are handled.
  - **Option Selected** The original image is imported. However, the <img> tag is modified in the imported file to closely reflect the height and width of the image in the FrameMaker document. This is done regardless of whether you are importing linked or embedded images from FrameMaker documents.
  - **Option NOT Selected** The <img> tag is not modified in the imported file. Instead, the image is referenced at 100% of its original value.

- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your FrameMaker source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import FrameMaker files when you attempt to build output.

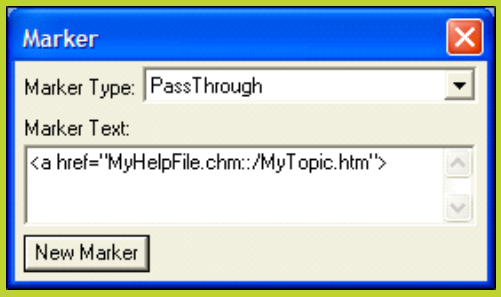
- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.

- **Enable 'Passthrough' Markers** Select this check box to include a check mark if you have created passthrough markers in your FrameMaker source documents.A passthrough marker is a special marker that you can insert into your FrameMaker source content when you have information or code that you plan to import to Flare and want left alone (or "passed through," leaving it exactly as you have authored it, rather than processing it). A passthrough marker can be just about anything, as long as supports it in the XHTML code. You can specify how the marker content should be treated when the FrameMaker document is imported. The first option is that you can import the marker content as regular text (which is the default setting). The second option is that you can import the marker content as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). The third option is that you can import the marker content as a complete XML tag. You might use a passthrough marker for various reasons, such as for importing a marker as XHTML or JavaScript code.

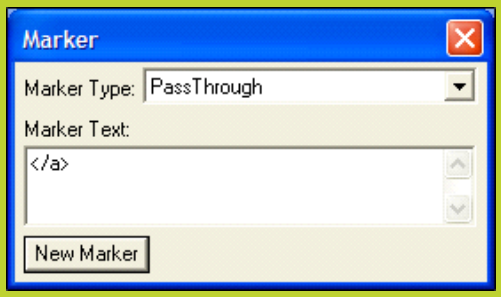
E X A M P L E

Let's say you plan to import some FrameMaker documents to Flare and you have locations in those documents where you want to link to CHM files. The problem is that FrameMaker does not allow you to create links to CHM files in such a way that those links can then be imported into another software application.

Therefore, you create a passthrough marker in the FrameMaker document, providing the beginning "href" tag and path to the CHM file. Like this:



Then you create a second passthrough marker, providing the end tag for the link. Like this:



When you import the FrameMaker document(s), you can specify that the passthrough markers should be imported as XML fragments. In Flare, the link to the CHM file will look and work as it should.

» **Passthrough Marker Format** After you enable passthrough markers, click the down arrow in this field and select the type of format that you want to use for the import.

  » **text** The marker content will be imported as regular text (default setting).

- **fragment** The marker content is imported as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). If you select this option, you will probably need a second marker in the FrameMaker document to complete the XML tag.
- **xml** The marker content is imported as a complete XML tag.

- **Convert equations to MathML** Select this option to convert MathFullForm equations (the FrameMaker-specific format) to MathML (the web standard and Flare format). If you disable this option, equations from FrameMaker are converted to images.
- **Reimport Table Styles** This option displays only if you are working in the Import Editor, rather than the wizard. This option is useful after you have already imported FrameMaker documents and converted the formatting in your tables into at least one table stylesheet in Flare. You can use this option to determine whether tables should be imported again as table styles when you re-import. You might want to keep this check box selected for some re-imports, but other times you might want to deselect it when re-importing.

E X A M P L E

Let's say you want the formatted tables in your FrameMaker documents to be converted to table styles when you perform the initial import into a Flare project. Therefore, in the import wizard, you turn on the "Convert Table Styles" option. As a result, let's say that Flare creates a new table style and calls it "FormatA.css."

After the initial import, you realize you want the tables to look a little different. Therefore, in the Flare project, you modify the properties of the FormatA.css table stylesheet.

Awhile later, let's say you want to re-import the FrameMaker documents. The problem is that you've already changed the table stylesheet in Flare. You probably want to keep the tweaked table style so that you don't have to modify it again after the import.

This is where the new "re-import" option comes into play. It determines whether or not a second new table stylesheet will be created, based on the old look from the tables in the FrameMaker documents.

Here's one scenario. Let's say that before you re-imported the FrameMaker documents, you selected the "Reimport Table Styles" option in the Frame Import Editor. And during the import when you were prompted, you selected not to overwrite the existing FormatA.css table stylesheet. In that case, Flare keeps your tweaked stylesheet in the project, but it also creates another table stylesheet called "FormatA1.css" that has the old look and feel. All of the re-imported content now links to the FormatA1.css stylesheet instead of FormatA.css.

Here's a different scenario. Let's say that you perform the same steps described above, except this time you deselected the "Reimport Table Styles" option in the Frame Import Editor. In that case, the second FormatA1.css file is not created. The imported content is linked to the FormatA.css table stylesheet that you previously modified, since it already exists in the project.

14. Click **Next**.
15. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your FrameMaker files.

    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
    - **Preserve FrameMaker Styles** This retains any style formatting from your FrameMaker documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).
    - **Don't Preserve FrameMaker Styles** This does not keep the style formatting used in the FrameMaker documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that either keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).
    - **Conversion Styles** This opens the Import Styles Editor, which lets you specify how to convert each property of the FrameMaker styles. If you do not enter a property value, the value from the FrameMaker document is used. If you enter a property value, it overrides the value from the FrameMaker document. This button is used only if you have selected "Preserve FrameMaker Styles."

E X A M P L E

You might use this button, for example, if you need to change a cross-reference format coming from FrameMaker into something more meaningful in Flare. There are some cross-reference building blocks in FrameMaker that do not have an equivalent in Flare. In cases such as these, the formats are preserved after conversion to Flare. However, the formats may therefore appear to

be broken, but they are preserved to let you know that there was some formatting in a cross-reference style that Flare did not understand; you can then make changes to the cross-reference style in the stylesheet. Therefore, if you already know ahead of time that you have a cross-reference style that will need to be modified for use in Flare, you can use the Conversion Styles button and change the cross-reference format to something that Flare understands.

16. Click **Next**.
17. Use this page to map paragraph styles from the FrameMaker documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. To map a style, click the style in the **FrameMaker Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the FrameMaker styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(FrameMaker Style).

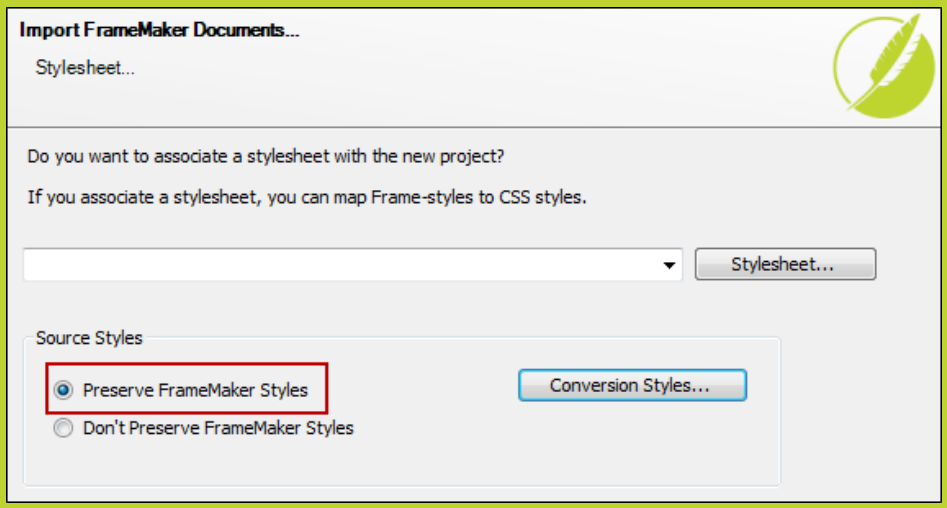
    The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

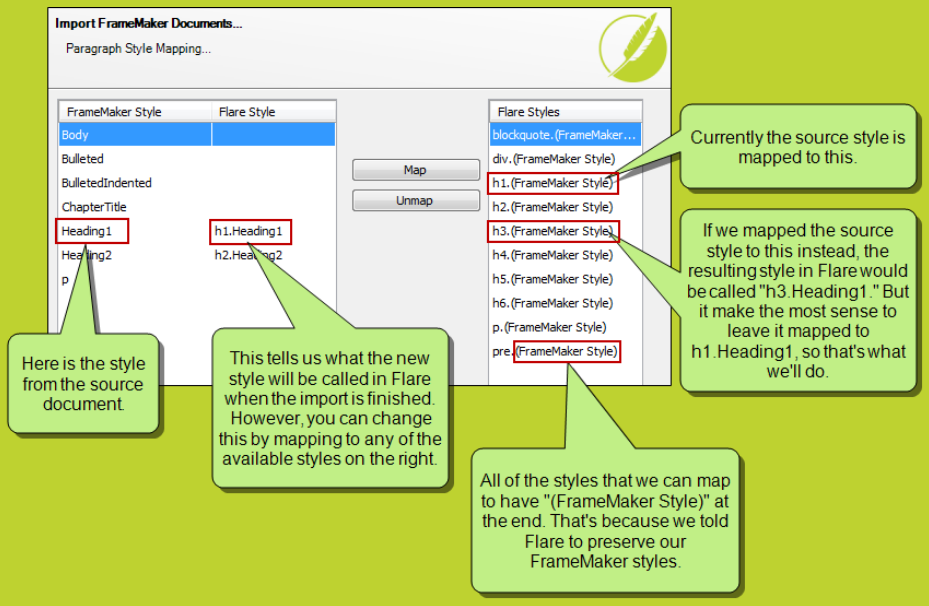
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *preserve* your FrameMaker styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading1 in the source document is still using Arial 14 pt and is red, just like it was before in FrameMaker. However, the style is now called "h1.Heading1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the FrameMaker style, Flare added it as a child under its parent, h1.

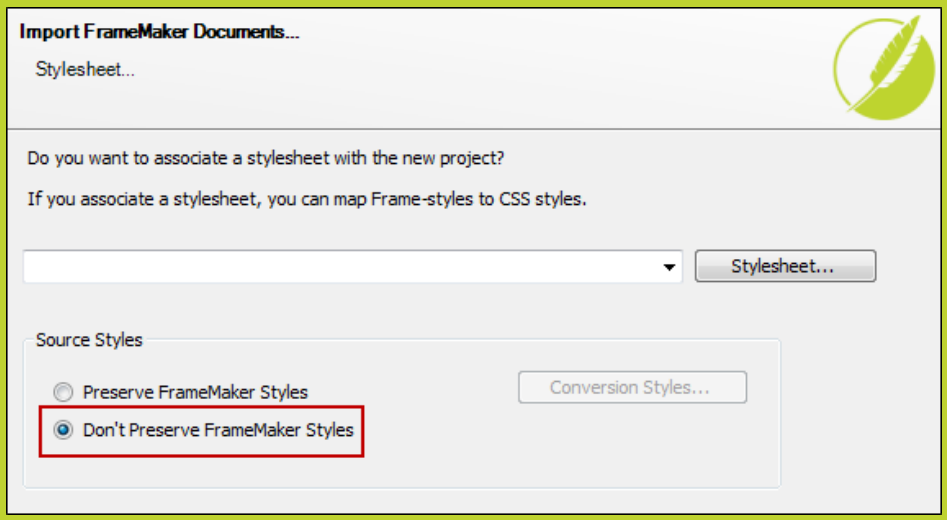
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

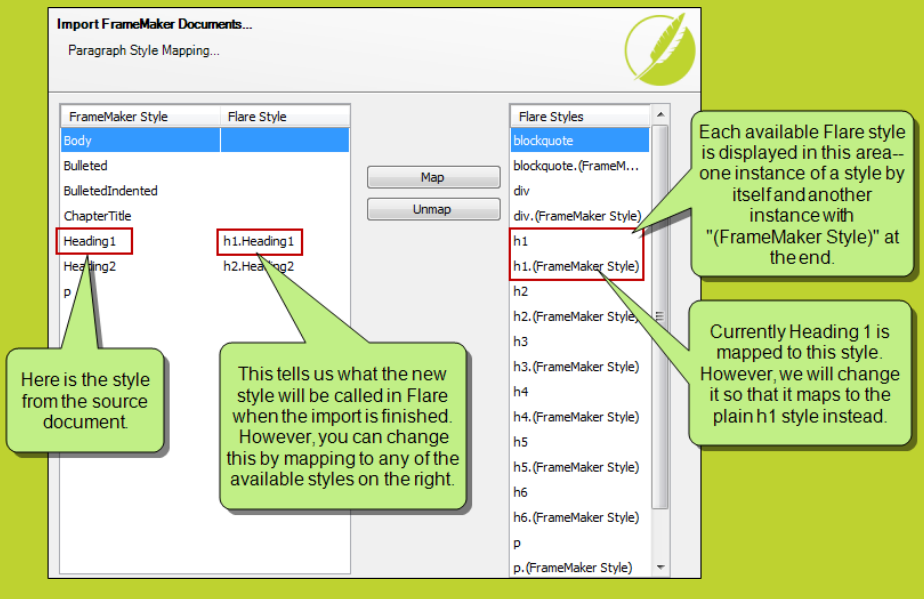
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

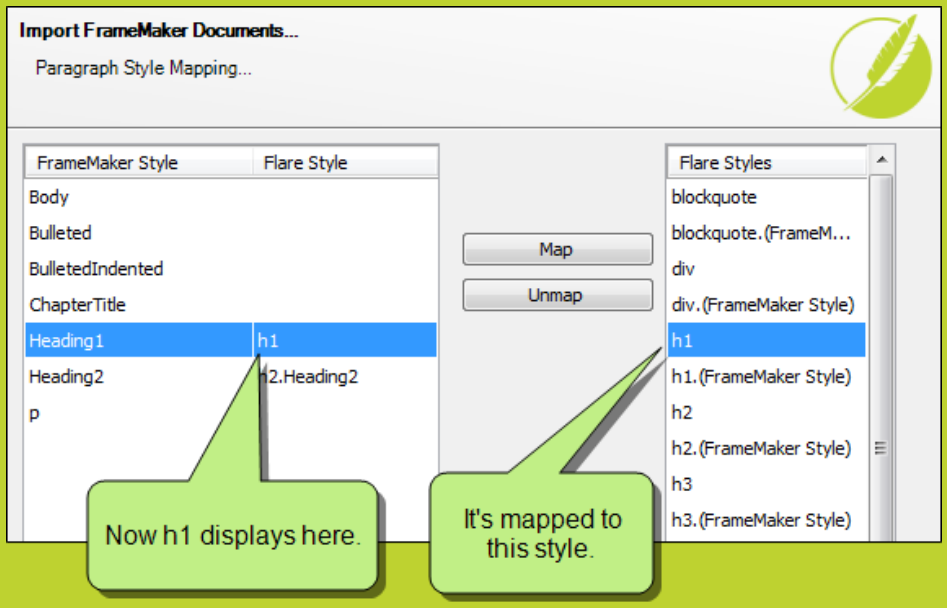
Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *not preserve* your FrameMaker styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1. (FrameMaker Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

18. Click **Next**.
19. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style,

click the style in the **FrameMaker Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

20. Click **Next**.
21. Use this page to map cross-reference (x-ref) styles from the FrameMaker documents to Flare's cross-reference styles, including those from the stylesheet you may have selected. In this way, you can have your FrameMaker style take on the appearance of the Flare style that you map it to. To map a style, click the style in the **FrameMaker Style** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button.

    The style is added to the Flare Style column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the FrameMaker style in the FrameMaker document will now be associated with a new style that has the appearance of the style that you mapped it to.

    What happens if you do not map a style? In the case of cross-reference styles, they are automatically added as style classes under the <MadCap|xref> tag. For example, let's say you import a style called "PageOnly" from your source document and do not map it to anything. In that case, it will be called "MadCap|xref.PageOnly" in the resulting project.
22. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
23. When you are finished previewing the files to be created, click **Accept**. The files are imported and the project is created.

HOW TO IMPORT FRAMEMAKER FILES INTO AN EXISTING PROJECT—VIA THE WIZARD

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **Project>Import>FrameMaker Documents**.
   - **Menu** Select **File>Import Project>FrameMaker Documents**.

   The Import FrameMaker Documents Wizard opens.
2. Select **Import into this project**.
3. Click **Next**.
4. Click .
5. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select BOOK, FM, or MIF files.

   Tip: When possible, it is recommended that you select a Adobe FrameMaker BOOK file for import and let Flarelocate and import all the associated document files within the Adobe FrameMaker book.
6. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original FrameMaker documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original FrameMaker documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

7. Click **Next**. The FrameMaker documents are scanned and the next page of the wizard opens. The paragraph styles used in the FrameMaker documents are shown on the left side of the wizard page ("Used FrameMaker Styles").

> **Note:** You can skip the rest of the wizard pages by clicking **Finish**.

8. If you want Flare to split the FrameMaker documents into smaller topics based on any of the styles shown on the left side of the page ("Used FrameMaker Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

> **E X A M P L E**
>
> If you have a style called "Heading 2" in your FrameMaker documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

9. Click **Next**.
10. You can use the following options as necessary:

   - **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
   - **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
   - **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a

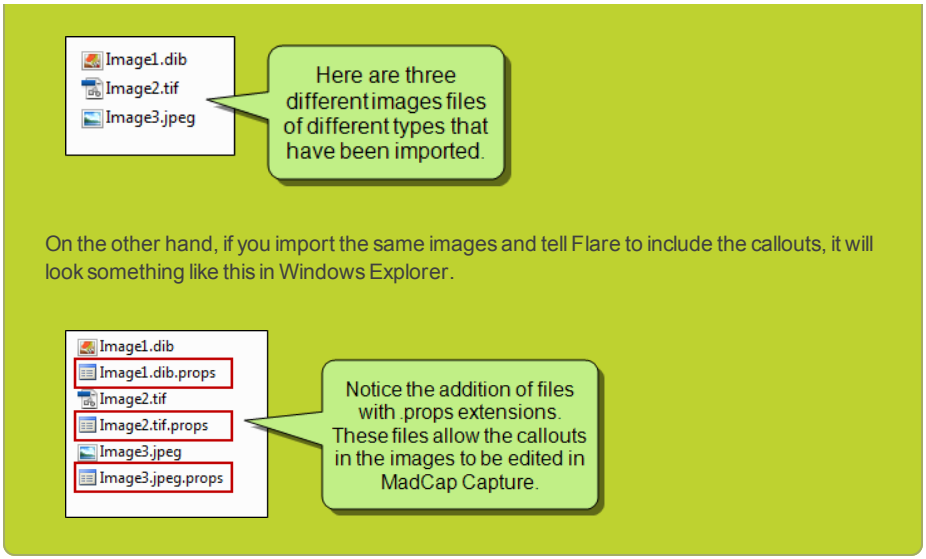
heading in each topic that is connected to another topic included in the split. That text will be used in the link instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.

- **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.

- **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
  - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
- **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the FrameMaker documents without any content.
  - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
- **Anchored Frames With Images** You can use this area to specify how Flare should handle anchored frames that contain images as well as other content (e.g., text callouts).
  - **Generate Images Without Callouts** If the anchored frame contains an image along with callout text, the original image is imported *without* the callout text. You might select this option if you have resized the image in FrameMaker. With this option, the imported image is likely to be of a higher quality than it would be otherwise. You can then add a callout to the image once it is inside Flare.
  - **Generate Images With Callouts** If the anchored frame contains an image along with callout text, Flare creates a PROPS (i.e., properties) file along with the image file when that document is imported. This means that you can open those image files in MadCap Capture to edit those callouts after the import process is completed.

E X A M P L E

If you import anchored frame images and tell Flare not to include the callouts, your imported image files will look something like this in Windows Explorer.



Note: Some FrameMaker elements—such as arcs and nested frames—are not supported with this option.

- **Generate Flattened Images** If the anchored frame contains an image along with callout text, a new flattened image will be created as a result. The callout is included, but you cannot edit it.

- **Preserve Image Size** This option affects how the size of imported images are handled.
  - **Option Selected** The original image is imported. However, the <img> tag is modified in the imported file to closely reflect the height and width of the image in the FrameMaker document. This is done regardless of whether you are importing linked or embedded images from FrameMaker documents.
  - **Option NOT Selected** The <img> tag is not modified in the imported file. Instead, the image is referenced at 100% of its original value.

- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your FrameMaker source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing

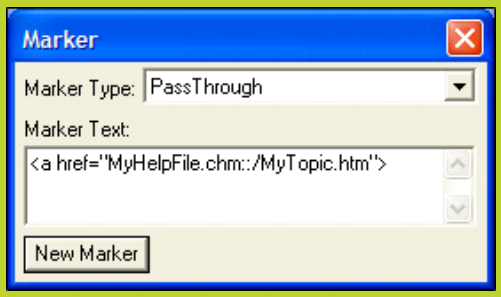
the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import FrameMaker files when you attempt to build output.

- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.
- **Enable 'Passthrough' Markers** Select this check box to include a check mark if you have created passthrough markers in your FrameMaker source documents.A passthrough marker is a special marker that you can insert into your FrameMaker source content when you have information or code that you plan to import to Flare and want left alone (or "passed through," leaving it exactly as you have authored it, rather than processing it). A passthrough marker can be just about anything, as long as supports it in the XHTML code. You can specify how the marker content should be treated when the FrameMaker document is imported. The first option is that you can import the marker content as regular text (which is the default setting). The second option is that you can import the marker content as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). The third option is that you can import the marker content as a complete XML tag. You might use a passthrough marker for various reasons, such as for importing a marker as XHTML or JavaScript code.

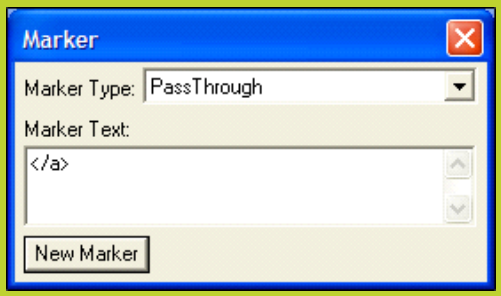
E X A M P L E

Let's say you plan to import some FrameMaker documents to Flare and you have locations in those documents where you want to link to CHM files. The problem is that FrameMaker does not allow you to create links to CHM files in such a way that those links can then be imported into another software application.

Therefore, you create a passthrough marker in the FrameMaker document, providing the beginning "href" tag and path to the CHM file. Like this:



Then you create a second passthrough marker, providing the end tag for the link. Like this:



When you import the FrameMaker document(s), you can specify that the passthrough markers should be imported as XML fragments. In Flare, the link to the CHM file will look and work as it should.

- **Passthrough Marker Format** After you enable passthrough markers, click the down arrow in this field and select the type of format that you want to use for the import.
  - **text** The marker content will be imported as regular text (default setting).
  - **fragment** The marker content is imported as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). If you select this option, you will probably need a second marker in the FrameMaker document to complete the XML tag.
  - **xml** The marker content is imported as a complete XML tag.
- **Convert equations to MathML** Select this option to convert MathFullForm equations (the FrameMaker-specific format) to MathML (the web standard and Flare format). If you disable this option, equations from FrameMaker are converted to images.
- **Reimport Table Styles** This option displays only if you are working in the Import Editor, rather than the wizard. This option is useful after you have already imported FrameMaker documents and converted the formatting in your tables into at least one table stylesheet in Flare. You can use this option to determine whether tables should be imported again as table styles when you re-import. You might want to keep this check box selected for some re-imports, but other times you might want to deselect it when re-importing.

EXAMPLE

Let's say you want the formatted tables in your FrameMaker documents to be converted to table styles when you perform the initial import into a Flare project. Therefore, in the import wizard, you turn on the "Convert Table Styles" option. As a result, let's say that Flare creates a new table style and calls it "FormatA.css."

After the initial import, you realize you want the tables to look a little different. Therefore, in the Flare project, you modify the properties of the FormatA.css table stylesheet.

Awhile later, let's say you want to re-import the FrameMaker documents. The problem is that you've already changed the table stylesheet in Flare. You probably want to keep the tweaked table style so that you don't have to modify it again after the import.

This is where the new "re-import" option comes into play. It determines whether or not a second new table stylesheet will be created, based on the old look from the tables in the FrameMaker documents.

Here's one scenario. Let's say that before you re-imported the FrameMaker documents, you selected the "Reimport Table Styles" option in the Frame Import Editor. And during the import when you were prompted, you selected not to overwrite the existing FormatA.css table stylesheet. In that case, Flare keeps your tweaked stylesheet in the project, but it also creates another table stylesheet called "FormatA1.css" that has the old look and feel. All of the re-imported content now links to the FormatA1.css stylesheet instead of FormatA.css.

Here's a different scenario. Let's say that you perform the same steps described above, except this time you deselected the "Reimport Table Styles" option in the Frame Import Editor. In that case, the second FormatA1.css file is not created. The imported content is linked to the FormatA.css table stylesheet that you previously modified, since it already exists in the project.

11. Click **Next**.
12. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your FrameMaker files.
    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
    - **Preserve FrameMaker Styles** This retains any style formatting from your FrameMaker documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to

make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).

- **Don't Preserve FrameMaker Styles** This does not keep the style formatting used in the FrameMaker documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that either keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).
- **Conversion Styles** This opens the Import Styles Editor, which lets you specify how to convert each property of the FrameMaker styles. If you do not enter a property value, the value from the FrameMaker document is used. If you enter a property value, it overrides the value from the FrameMaker document. This button is used only if you have selected "Preserve FrameMaker Styles."

E X A M P L E

You might use this button, for example, if you need to change a cross-reference format coming from FrameMaker into something more meaningful in Flare. There are some cross-reference building blocks in FrameMaker that do not have an equivalent in Flare. In cases such as these, the formats are preserved after conversion to Flare. However, the formats may therefore appear to be broken, but they are preserved to let you know that there was some formatting in a cross-reference style that Flare did not understand; you can then make changes to the cross-reference style in the stylesheet. Therefore, if you already know ahead of time that you have a cross-reference style that will need to be modified for use in Flare, you can use the Conversion Styles button and change the cross-reference format to something that Flare understands.

13. Click **Next**.
14. Use this page to map paragraph styles from the FrameMaker documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. To map a style, click the style in the **FrameMaker Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the FrameMaker styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(FrameMaker Style).

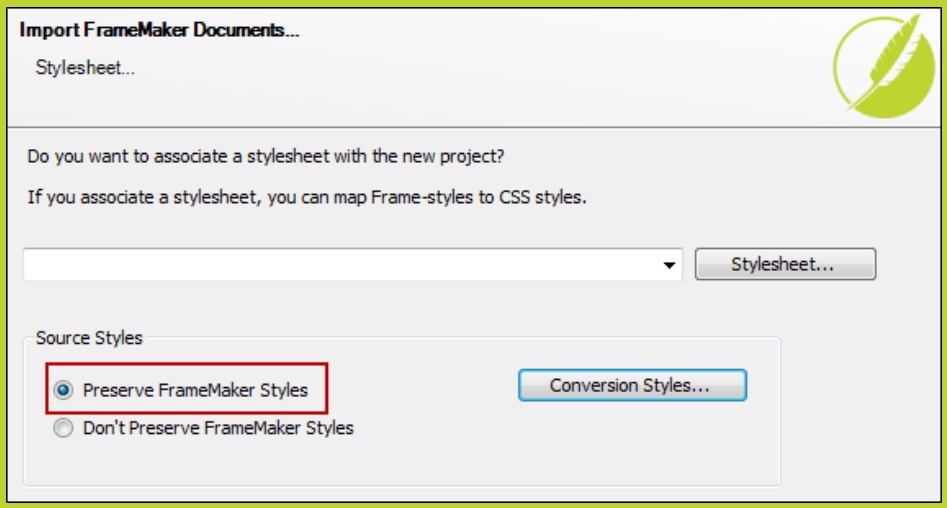
    The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

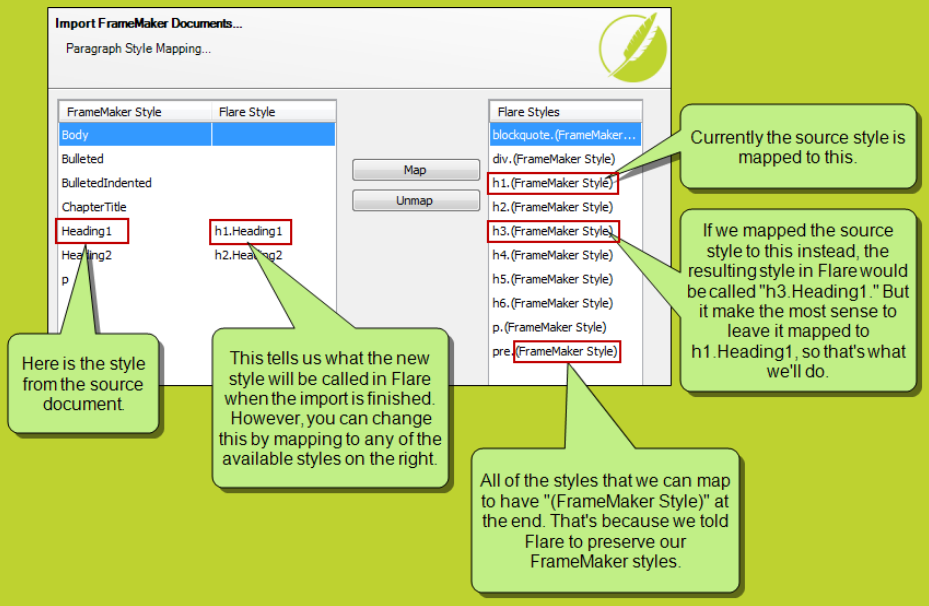
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *preserve* your FrameMaker styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading1 in the source document is still using Arial 14 pt and is red, just like it was before in FrameMaker. However, the style is now called "h1.Heading1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the FrameMaker style, Flare added it as a child under its parent, h1.

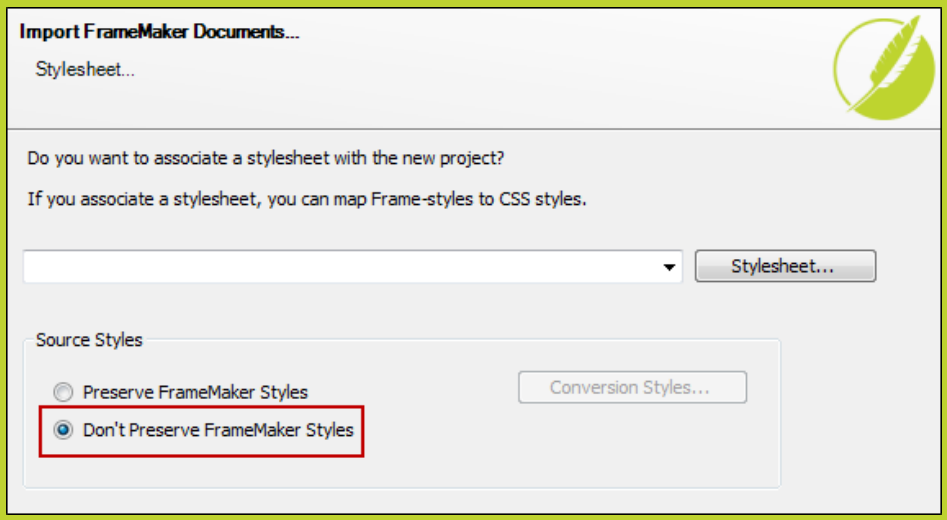
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

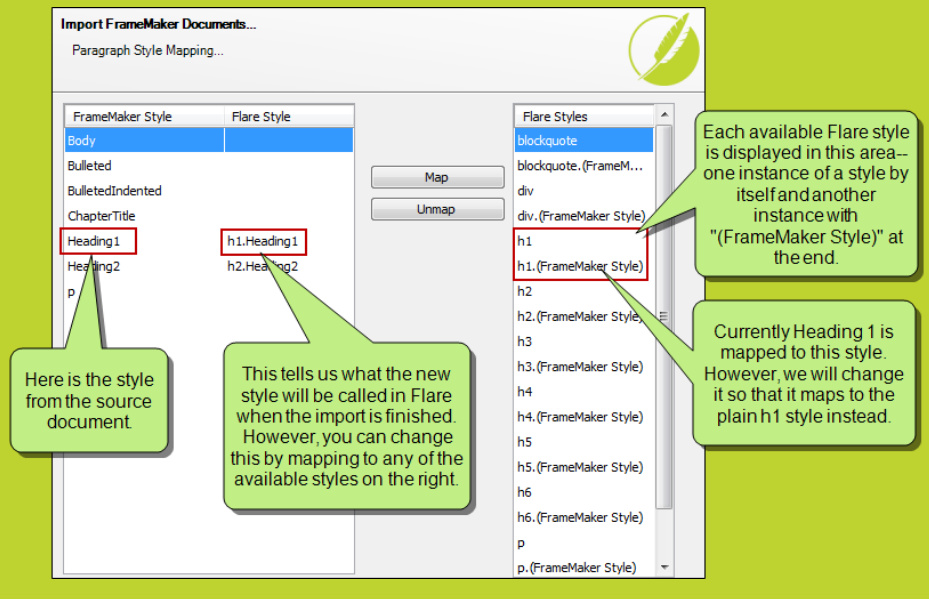
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

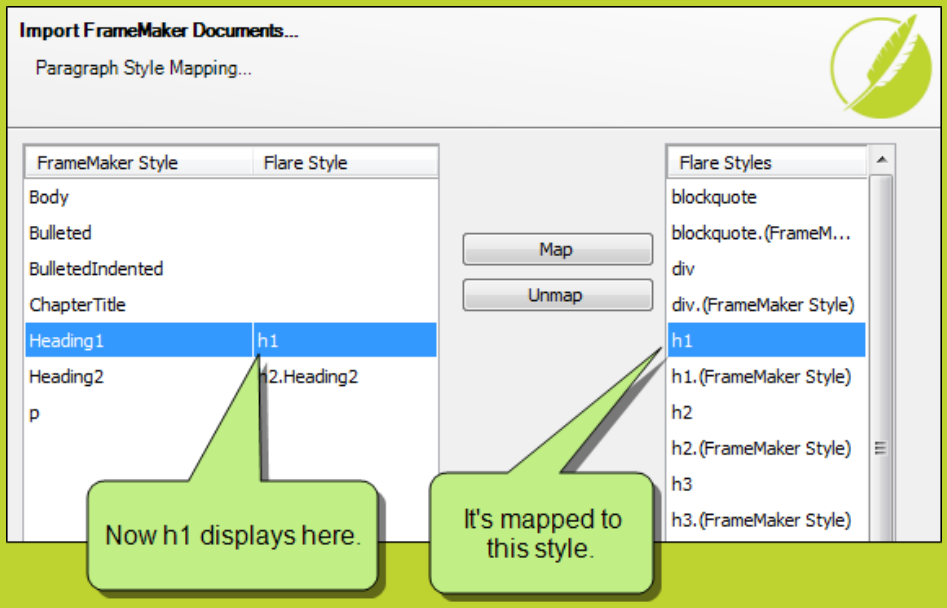
Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *not preserve* your FrameMaker styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1. (FrameMaker Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

15. Click **Next**.
16. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style,

click the style in the **FrameMaker Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

17. Click **Next**.
18. Use this page to map cross-reference (x-ref) styles from the FrameMaker documents to Flare's cross-reference styles, including those from the stylesheet you may have selected. In this way, you can have your FrameMaker style take on the appearance of the Flare style that you map it to. To map a style, click the style in the **FrameMaker Style** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button.

    The style is added to the Flare Style column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the FrameMaker style in the FrameMaker document will now be associated with a new style that has the appearance of the style that you mapped it to.

    What happens if you do not map a style? In the case of cross-reference styles, they are automatically added as style classes under the <MadCap|xref> tag. For example, let's say you import a style called "PageOnly" from your source document and do not map it to anything. In that case, it will be called "MadCap|xref.PageOnly" in the resulting project.
19. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
20. When you are finished previewing the files to be created, click **Accept**.

HOW TO IMPORT FRAMEMAKER FILES INTO AN EXISTING PROJECT—VIA THE IMPORT EDITOR

1. Do one of the following:

   IF YOU DO NOT YET HAVE A FRAMEMAKER IMPORT FILE

   a. Complete the steps for adding an import file.

   b. Open the file in the Frame Import Editor.

   IF YOU ALREADY HAVE A FRAMEMAKER IMPORT FILE

   a. In the Project Organizer, expand the **Imports** subfolder.

   b. Double-click the appropriate import file. The file opens in the Frame Import Editor to the right.

2. Select the **Source Files** tab.

3. Click .

4. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select BOOK, FM, or MIF files.

   Tip: When possible, it is recommended that you select a Adobe FrameMaker BOOK file for import and let Flarelocate and import all the associated document files within the Adobe FrameMaker book.

5. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original FrameMaker documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original FrameMaker documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

6. Select the **New Topic Styles** tab.
7. If you want Flare to split the FrameMaker documents into smaller topics based on any of the styles shown on the left side of the page ("Used FrameMaker Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

> **E X A M P L E**
>
> If you have a style called "Heading 2" in your FrameMaker documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

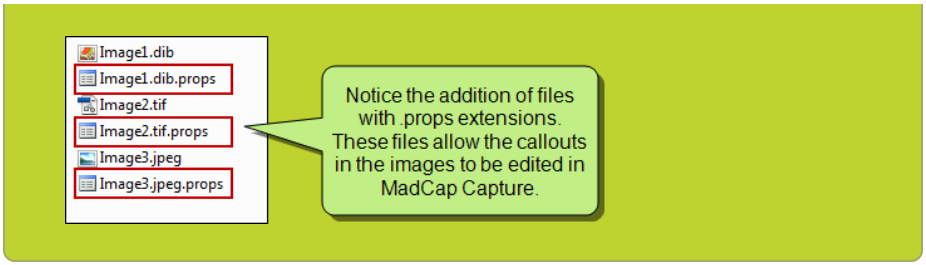
8. Select the **Options** tab.
9. You can use the following options as necessary:
   - **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
   - **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
   - **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a heading in each topic that is connected to another topic included in the split. That text will be used in the link instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.
     - **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.

- **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
  - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
- **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the FrameMaker documents without any content.
  - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
- **Anchored Frames With Images** You can use this area to specify how Flare should handle anchored frames that contain images as well as other content (e.g., text callouts).
  - **Generate Images Without Callouts** If the anchored frame contains an image along with callout text, the original image is imported *without* the callout text. You might select this option if you have resized the image in FrameMaker. With this option, the imported image is likely to be of a higher quality than it would be otherwise. You can then add a callout to the image once it is inside Flare.
  - **Generate Images With Callouts** If the anchored frame contains an image along with callout text, Flare creates a PROPS (i.e., properties) file along with the image file when that document is imported. This means that you can open those image files in MadCap Capture to edit those callouts after the import process is completed.

EXAMPLE

If you import anchored frame images and tell Flare not to include the callouts, your imported image files will look something like this in Windows Explorer.

On the other hand, if you import the same images and tell Flare to include the callouts, it will look something like this in Windows Explorer.



Note: Some FrameMaker elements—such as arcs and nested frames—are not supported with this option.

- **Generate Flattened Images** If the anchored frame contains an image along with callout text, a new flattened image will be created as a result. The callout is included, but you cannot edit it.

- **Preserve Image Size** This option affects how the size of imported images are handled.
  - **Option Selected** The original image is imported. However, the <img> tag is modified in the imported file to closely reflect the height and width of the image in the FrameMaker document. This is done regardless of whether you are importing linked or embedded images from FrameMaker documents.
  - **Option NOT Selected** The <img> tag is not modified in the imported file. Instead, the image is referenced at 100% of its original value.

- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your FrameMaker source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import FrameMaker files when you attempt to build output.

- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.

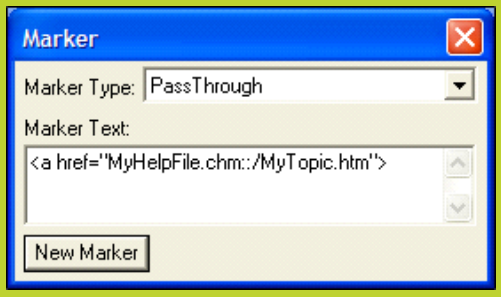
- **Enable 'Passthrough' Markers** Select this check box to include a check mark if you have created passthrough markers in your FrameMaker source documents.A passthrough marker is a special marker that you can insert into your FrameMaker source content when you have information or code that you plan to import to Flare and want left alone (or "passed through," leaving it exactly as you have authored it, rather than processing it). A passthrough marker can be just about anything, as long as supports it in the XHTML

code. You can specify how the marker content should be treated when the FrameMaker document is imported. The first option is that you can import the marker content as regular text (which is the default setting). The second option is that you can import the marker content as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). The third option is that you can import the marker content as a complete XML tag. You might use a passthrough marker for various reasons, such as for importing a marker as XHTML or JavaScript code.

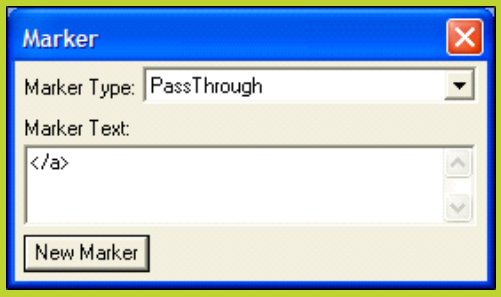
E X A M P L E

Let's say you plan to import some FrameMaker documents to Flare and you have locations in those documents where you want to link to CHM files. The problem is that FrameMaker does not allow you to create links to CHM files in such a way that those links can then be imported into another software application.

Therefore, you create a passthrough marker in the FrameMaker document, providing the beginning "href" tag and path to the CHM file. Like this:



Then you create a second passthrough marker, providing the end tag for the link. Like this:



When you import the FrameMaker document(s), you can specify that the passthrough markers should be imported as XML fragments. In Flare, the link to the CHM file will look and work as it should.

- **Passthrough Marker Format** After you enable passthrough markers, click the down arrow in this field and select the type of format that you want to use for the import.
  - **text** The marker content will be imported as regular text (default setting).
  - **fragment** The marker content is imported as an XML fragment (e.g., the first part of a bold tag—<b>—but not the second part). If you select this option, you will probably need a second marker in the FrameMaker document to complete the XML tag.
  - **xml** The marker content is imported as a complete XML tag.
- **Convert equations to MathML** Select this option to convert MathFullForm equations (the FrameMaker-specific format) to MathML (the web standard and Flare format). If you disable this option, equations from FrameMaker are converted to images.
- **Reimport Table Styles** This option displays only if you are working in the Import Editor, rather than the wizard. This option is useful after you have already imported FrameMaker documents and converted the formatting in your tables into at least one table stylesheet in Flare. You can use this option to determine whether tables should be imported again as table styles when you re-import. You might want to keep this check box selected for some re-imports, but other times you might want to deselect it when re-importing.

EXAMPLE

Let's say you want the formatted tables in your FrameMaker documents to be converted to table styles when you perform the initial import into a Flare project. Therefore, in the import wizard, you turn on the "Convert Table Styles" option. As a result, let's say that Flare creates a new table style and calls it "FormatA.css."

After the initial import, you realize you want the tables to look a little different. Therefore, in the Flare project, you modify the properties of the FormatA.css table stylesheet.

Awhile later, let's say you want to re-import the FrameMaker documents. The problem is that you've already changed the table stylesheet in Flare. You probably want to keep the tweaked table style so that you don't have to modify it again after the import.

This is where the new "re-import" option comes into play. It determines whether or not a second new table stylesheet will be created, based on the old look from the tables in the FrameMaker documents.

Here's one scenario. Let's say that before you re-imported the FrameMaker documents, you selected the "Reimport Table Styles" option in the Frame Import Editor. And during the import when you were prompted, you selected not to overwrite the existing FormatA.css table stylesheet. In that case, Flare keeps your tweaked stylesheet in the project, but it also creates another table stylesheet called "FormatA1.css" that has the old look and feel. All of the re-imported content now links to the FormatA1.css stylesheet instead of FormatA.css.

Here's a different scenario. Let's say that you perform the same steps described above, except this time you deselected the "Reimport Table Styles" option in the Frame Import Editor. In that case, the second FormatA1.css file is not created. The imported content is linked to the FormatA.css table stylesheet that you previously modified, since it already exists in the project.

10. Select the **Stylesheet** tab.
11. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your FrameMaker files.
    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
    - **Preserve FrameMaker Styles** This retains any style formatting from your FrameMaker documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to

make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).

- **Don't Preserve FrameMaker Styles** This does not keep the style formatting used in the FrameMaker documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the FrameMaker styles to new Flare styles that either keep the name of the FrameMaker style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).
- **Conversion Styles** This opens the Import Styles Editor, which lets you specify how to convert each property of the FrameMaker styles. If you do not enter a property value, the value from the FrameMaker document is used. If you enter a property value, it overrides the value from the FrameMaker document. This button is used only if you have selected "Preserve FrameMaker Styles."

E X A M P L E

You might use this button, for example, if you need to change a cross-reference format coming from FrameMaker into something more meaningful in Flare. There are some cross-reference building blocks in FrameMaker that do not have an equivalent in Flare. In cases such as these, the formats are preserved after conversion to Flare. However, the formats may therefore appear to be broken, but they are preserved to let you know that there was some formatting in a cross-reference style that Flare did not understand; you can then make changes to the cross-reference style in the stylesheet. Therefore, if you already know ahead of time that you have a cross-reference style that will need to be modified for use in Flare, you can use the Conversion Styles button and change the cross-reference format to something that Flare understands.

12. Select the **Paragraph Styles** tab.
13. Use this page to map paragraph styles from the FrameMaker documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. To map a style, click the style in the **FrameMaker Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the FrameMaker styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(FrameMaker Style).

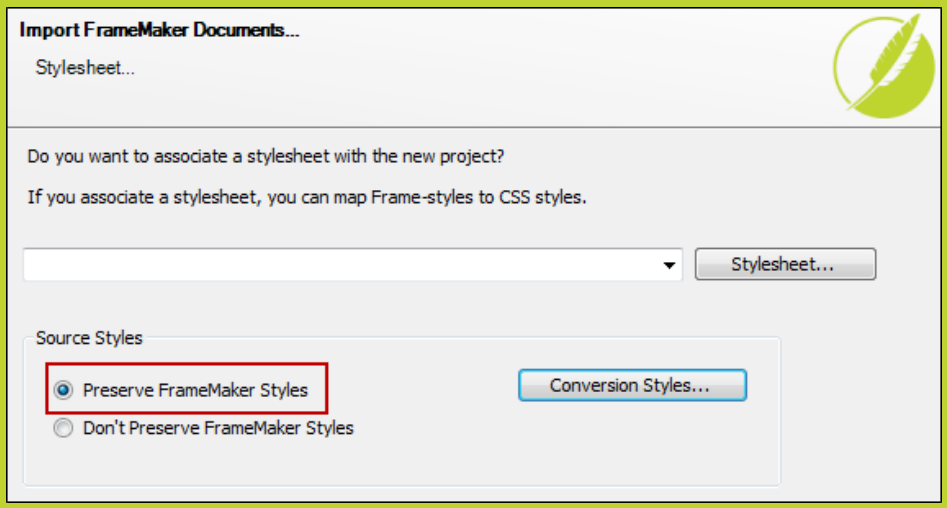
    The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

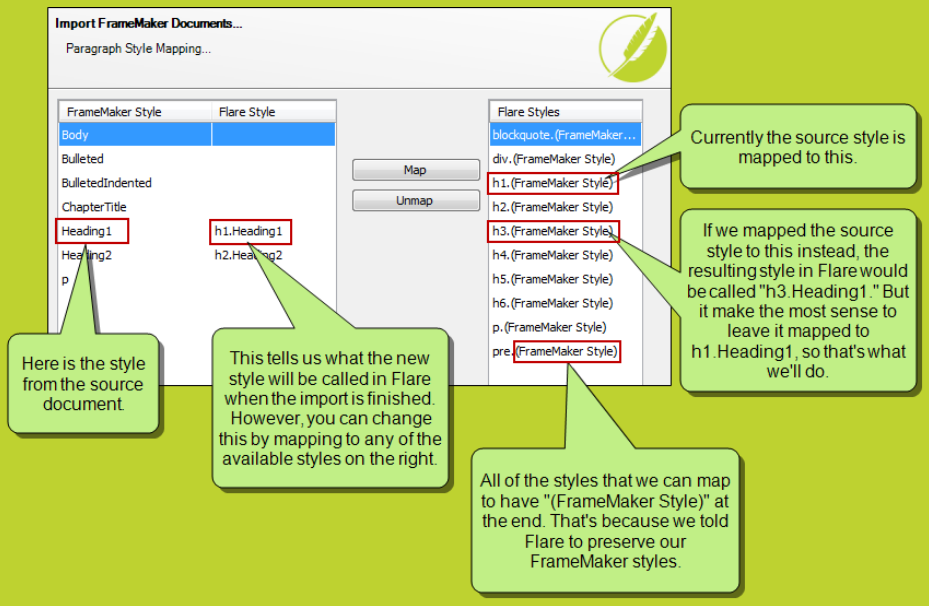
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *preserve* your FrameMaker styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading1 in the source document is still using Arial 14 pt and is red, just like it was before in FrameMaker. However, the style is now called "h1.Heading1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the FrameMaker style, Flare added it as a child under its parent, h1.

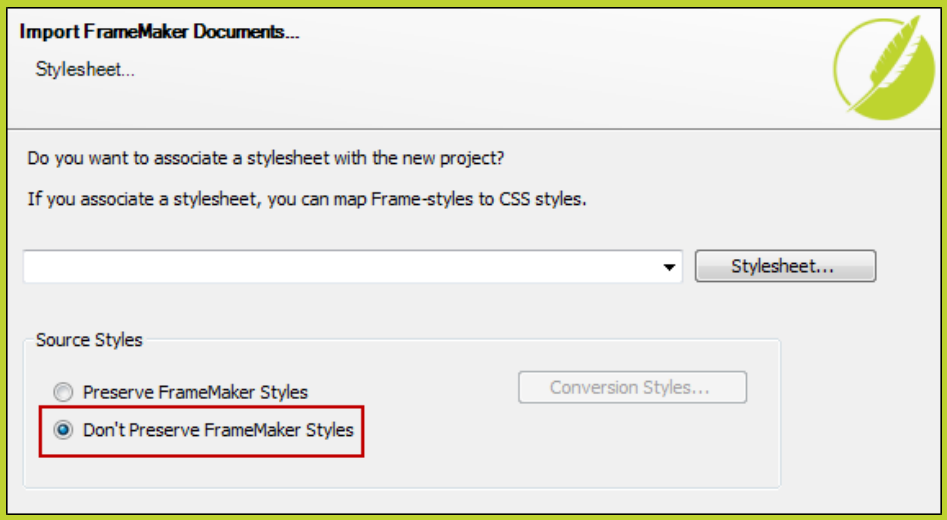
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

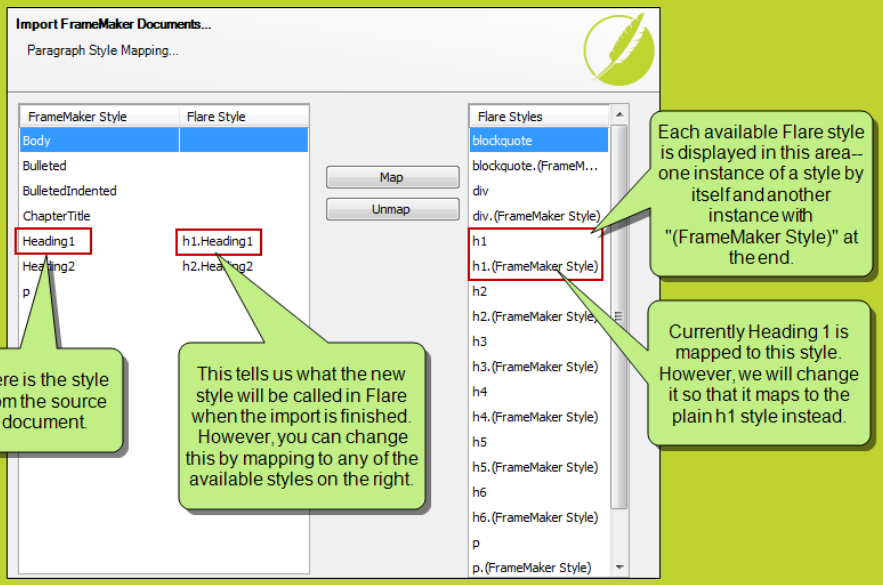
Let's say you have a style in your FrameMaker source document called "Heading1" that is using Arial 14 pt and is red, like this.

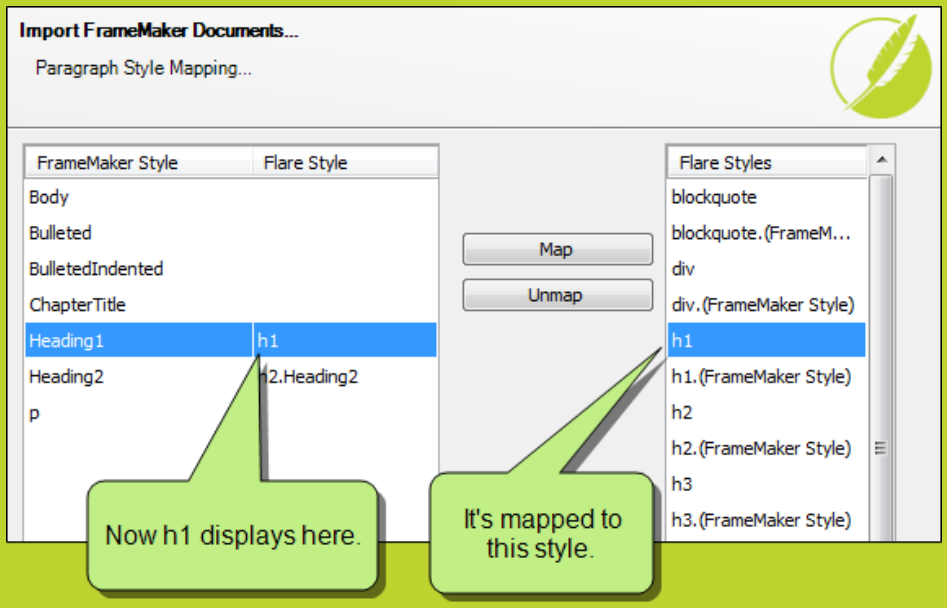
Hello

During the process of importing your FrameMaker document using the Import FrameMaker Wizard, you select the option to *not preserve* your FrameMaker styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1. (FrameMaker Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

14. Select the **Character Styles** tab.
15. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your FrameMaker style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style,

click the style in the **FrameMaker Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the FrameMaker document will now be associated with a new style that you mapped it to.

16. Select the **Cross-Reference Styles** tab.
17. Use this page to map cross-reference (x-ref) styles from the FrameMaker documents to Flare's cross-reference styles, including those from the stylesheet you may have selected. In this way, you can have your FrameMaker style take on the appearance of the Flare style that you map it to. To map a style, click the style in the **FrameMaker Style** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button.

    The style is added to the Flare Style column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the FrameMaker style in the FrameMaker document will now be associated with a new style that has the appearance of the style that you mapped it to.

    What happens if you do not map a style? In the case of cross-reference styles, they are automatically added as style classes under the <MadCap|xref> tag. For example, let's say you import a style called "PageOnly" from your source document and do not map it to anything. In that case, it will be called "MadCap|xref.PageOnly" in the resulting project.
18. In the local toolbar click **Import** (if this is the first time importing files) or **Reimport** (if files have been imported previously). The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
19. When you are finished previewing the files to be created, click **Accept**.

Note: The Imported Files tab displays the files that have been imported previously.

Note: You must have FrameMaker installed on your computer in order to import FrameMaker files.

Note: For each master page used in your FrameMaker documents, a corresponding page layout is automatically created in your Flare project. You can use the page layouts to configure pages for print-based output. You can then create chapter breaks for your print-based output and assign these page layouts to the different topics in the output.

Note: Flare supports FrameMaker 7.0 and newer versions.

Note: A link icon displays on tabs in the XML Editor next to file names that are imported from and linked to another file or Flare project. However, if you are also using the built-in source control technology, the source control icons have a higher precedence and will therefore be displayed instead.

Importing HTML Files

You can import XHTML and HTML files (automatically converting them to XHTML) in the following ways:

- Import Into a New Project
- Import Into an Existing Project (From the Wizard)
- Import Into an Existing Project (From the Import Editor)

HOW TO IMPORT HTML FILES INTO A NEW PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>HTML File Set** or **Project>Import>HTML File Set**.
   - **Menu** Select **File>Import Project>HTML File Set**.

   The Import HTML Files Wizard opens.
2. Select **Import into a new project**.
3. In the **Project name** field, type a name for the new Flare project that will be created after you perform the import.
4. In the **Project folder** field, either accept the default location for the new project or click ... to find and select a folder.
5. From the **Output Type** field, select the primary target for your project.
6. Click **Next**.
7. Click one of the following:

	In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.
	In the dialog that opens, find and select a folder containing HTML files you want to import. When you select a folder to import, the wizard imports all files within that folder that have an .htm, .html, or .xhtml extension. If you later re-import HTML files into the project, Flare checks to see if any of the files in the source folder have changed. It also determines whether files have been deleted or added, and it updates the source files list accordingly. When you are finished, click **Open**.

8. You can use the following options as necessary:

	This opens the file that is selected in the list.
	This opens the HTML to XHTML Conversion dialog, which lets you see how the selected file looks in HTML and how it will look after its conversion to XHTML.

Link Generated Files to Source Files	This creates a connection between the original HTML files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original HTML files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

9. Click **Next**.
10. You can use the following options as necessary:
    - **Import linked HTML files** Select this option if you want to automatically bring in other files that are linked to those you selected on the previous page. For example, if you import Doc1.htm, which contains a hyperlink to Doc2.htm, this option imports Doc2.htm as well.
    - **Import resources** Select this option if you want to include any supporting resource files (e.g., stylesheets, images, multimedia files) in the import. Then select one of the following:
        - **Keep existing structure** The supporting resources files will be copied into folders with the same names and hierarchy as those used in the source.
        - **To project resources folder** The supporting files will be placed in the Resources folder in your Flare project.
    - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your HTML source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import HTML files when you attempt to build output.

11. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
12. When you are finished previewing the files to be created, click **Accept**. The files are imported and the project is created.

HOW TO IMPORT HTML FILES INTO AN EXISTING PROJECT—VIA THE WIZARD

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **Project>Import>HTML File Set**.
   - **Menu** Select **File>Import Project>HTML File Set**.

   The Import HTML Files Wizard opens.
2. Select **Import into this project**.
3. Click **Next**.
4. Click one of the following:

	In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.
	In the dialog that opens, find and select a folder containing HTML files you want to import. When you select a folder to import, the wizard imports all files within that folder that have an .htm, .html, or .xhtml extension. If you later re-import HTML files into the project, Flare checks to see if any of the files in the source folder have changed. It also determines whether files have been deleted or added, and it updates the source files list accordingly. When you are finished, click **Open**.

5. You can use the following options as necessary:

	This opens the file that is selected in the list.
	This opens the HTML to XHTML Conversion dialog, which lets you see how the selected file looks in HTML and how it will look after its conversion to XHTML.
Link Generated Files to Source Files	This creates a connection between the original HTML files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original HTML files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

6. Click **Next**.
7. You can use the following options as necessary:
   - **Import to project content folder** Select this option if you want to store the files at the root level of the Content Explorer.
   - **Import to folder** Select this option if you want to store the files in a specific folder. Use the **Browse** button to select the folder. When you are finished click **Open**.
   - **Import linked HTML files** Select this option if you want to automatically bring in other files that are linked to those you selected on the previous page. For example, if you import Doc1.htm, which contains a hyperlink to Doc2.htm, this option imports Doc2.htm as well.
   - **Import resources** Select this option if you want to include any supporting resource files (e.g., stylesheets, images, multimedia files) in the import. Then select one of the following:
     - **Keep existing structure** The supporting resources files will be copied into folders with the same names and hierarchy as those used in the source.
     - **To project resources folder** The supporting files will be placed in the Resources folder in your Flare project.
     - **To folder** The supporting files will be placed into whatever folder in your project that you specify.
   - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your HTML source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import HTML files when you attempt to build output.
8. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
9. When you are finished previewing the files to be created, click **Accept**.

HOW TO IMPORT HTML FILES INTO AN EXISTING PROJECT—VIA THE IMPORT EDITOR

1. Do one of the following:

   IF YOU DO NOT YET HAVE AN HTML IMPORT FILE

   a. Complete the steps for adding an import file.

   b. Open the file in the HTML Import Editor.

   IF YOU ALREADY HAVE AN HTML IMPORT FILE

   a. In the Project Organizer, expand the **Imports** subfolder.

   b. Double-click the appropriate import file. The file opens in the HTML Import Editor to the right.

2. Select the **Source Files** tab.

3. Click one of the following:

	In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.
	In the dialog that opens, find and select a folder containing HTML files you want to import. When you select a folder to import, the wizard imports all files within that folder that have an .htm, .html, or .xhtml extension. If you later re-import HTML files into the project, Flare checks to see if any of the files in the source folder have changed. It also determines whether files have been deleted or added, and it updates the source files list accordingly. When you are finished, click **Open**.

4. You can use the following options as necessary:

	This opens the file that is selected in the list.
	This opens the HTML to XHTML Conversion dialog, which lets you see how the selected file looks in HTML and how it will look after its conversion to XHTML.
Link Generated Files to Source Files	This creates a connection between the original HTML files and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original HTML files, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.

	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

5. Select the **Options** tab.
6. You can use the following options as necessary:
   - **Import to project content folder** Select this option if you want to store the files at the root level of the Content Explorer.
   - **Import to folder** Select this option if you want to store the files in a specific folder. Use the **Browse** button to select the folder. When you are finished click **Open**.
   - **Import linked HTML files** Select this option if you want to automatically bring in other files that are linked to those you selected on the previous page. For example, if you import Doc1.htm, which contains a hyperlink to Doc2.htm, this option imports Doc2.htm as well.
   - **Import resources** Select this option if you want to include any supporting resource files (e.g., stylesheets, images, multimedia files) in the import. Then select one of the following:
     - **Keep existing structure** The supporting resources files will be copied into folders with the same names and hierarchy as those used in the source.
     - **To project resources folder** The supporting files will be placed in the Resources folder in your Flare project.
     - **To folder** The supporting files will be placed into whatever folder in your project that you specify.
   - **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your HTML source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import HTML files when you attempt to build output.
7. In the local toolbar click **Import** (if this is the first time importing files) or **Reimport** (if files have been imported previously). The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
8. When you are finished previewing the files to be created, click **Accept**.

Note: The Imported Files tab displays the files that have been imported previously.

Importing Word Files

You can import Microsoft Word files in the following ways:

- Import Into a New Project
- Import Into an Existing Project (From the Wizard)
- Import Into an Existing Project (From the Import Editor)

HOW TO IMPORT WORD FILES INTO A NEW PROJECT

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>MS Word Documents** or **Project>Import>MS Word Documents**.
   - **Menu** Select **File>Import Project>MS Word Documents**.

   The Import Microsoft Word Wizard opens.
2. Select **Import into a new project**.
3. In the **Project name** field, type a name for the new Flare project that will be created after you perform the import.
4. In the **Project folder** field, either accept the default location for the new project or click ... to find and select a folder.
5. From the **Output Type** field, select the primary target for your project.
6. Click **Next**.
7. Click .
8. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DOC, DOCX, or RTF files.

   Note: DOCX is Microsoft Word's platform-independent, open XML format. You must have Microsoft Word 2007 installed in order to import this file type.

9. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original Word documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original Word documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.
	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

10. Click **Next**. The Word documents are scanned and the next page of the wizard opens. The paragraph styles used in the Word documents are shown on the left side of the wizard page ("Used Word Styles").

Note: You can skip the rest of the wizard pages by clicking **Finish**.

11. If you want Flare to split the Word documents into smaller topics based on any of the styles shown on the left side of the page ("Used Word Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

E X A M P L E

If you have a style called "Heading 2" in your Word documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

12. Click **Next**.
13. You can use the following options as necessary:

- **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
- **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
- **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a heading in each topic that is connected to another topic included in the split. That text will be used in the link instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.
  - **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.
- **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
  - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
- **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the Word documents without any content.
  - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your Word source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-

imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import Word files when you attempt to build output.

- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.
- **Convert all tables to "Auto-Fit to Contents"** Select this option if you want to automatically set tables to "Auto-Fit to Contents" when they are imported into Flare. This simply ensures that column widths are not specified on the imported tables.
- **Create a Page Layout for each section header/footer** Select this option if you want Flare to create new page layouts when you import Word documents that have section breaks, along with headers or footers. For each new section in the Word document that has a different header or footer than the previous section, Flare creates a unique page layout. After the import is finished, you can open and edit the page layouts if necessary. You can then create chapter breaks for your print-based output and assign these page layouts to the different topics in the output.
- **Automatically set topic title** If this option is enabled, the properties setting for the topic title automatically uses the first heading in the topic. Therefore, if you change the heading text in the future, the topic title changes automatically as well. If this option is disabled, the properties setting for the topic title explicitly uses the first topic heading text found during the import, and it remains so unless you manually change it later.

14. Click **Next**.
15. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your Word files.

    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
    - **Preserve MS Word Styles** This retains any style formatting from your Word documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).
    - **Don't Preserve MS Word Styles** This does not keep the style formatting used in the Word documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that either keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).

» **Convert inline formatting to CSS styles** This creates new styles based on any local formatting that exists in the Word documents.

> **EXAMPLE**
>
> If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.

> **Tip:** To avoid having several character styles in Flare that you do not want, it is a good idea to clean up your source documents as much as possible before importing.

16. Click **Next**.
17. Use this page to map paragraph styles from the Word documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. To map a style, click the style in the **MS Word Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the Word styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(MS Word Style).

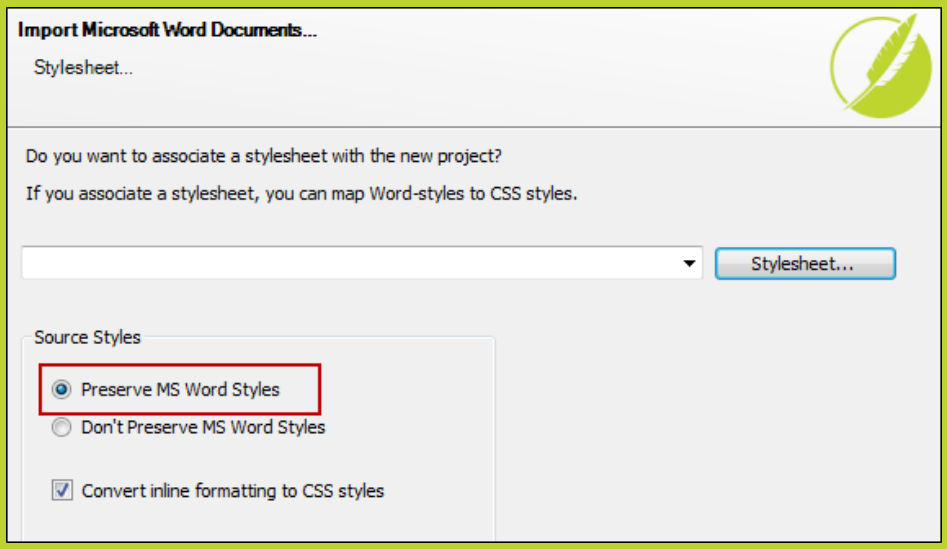
    The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

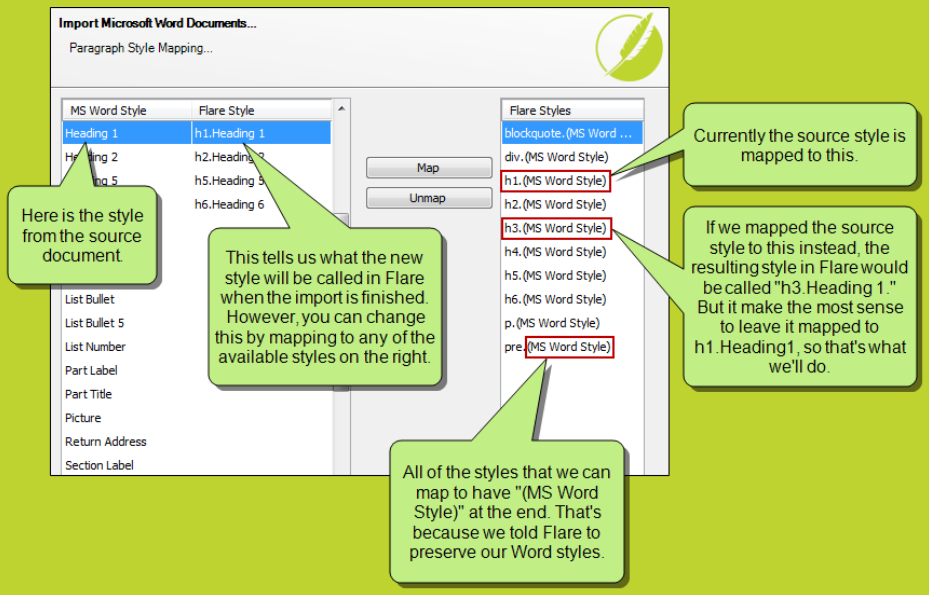
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *preserve* your Word styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading 1 in the source document is still using Arial 14 pt and is red, just like it was before in Word. However, the style is now called "h1.Heading 1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the Word style, Flare added it as a child under its parent, h1.

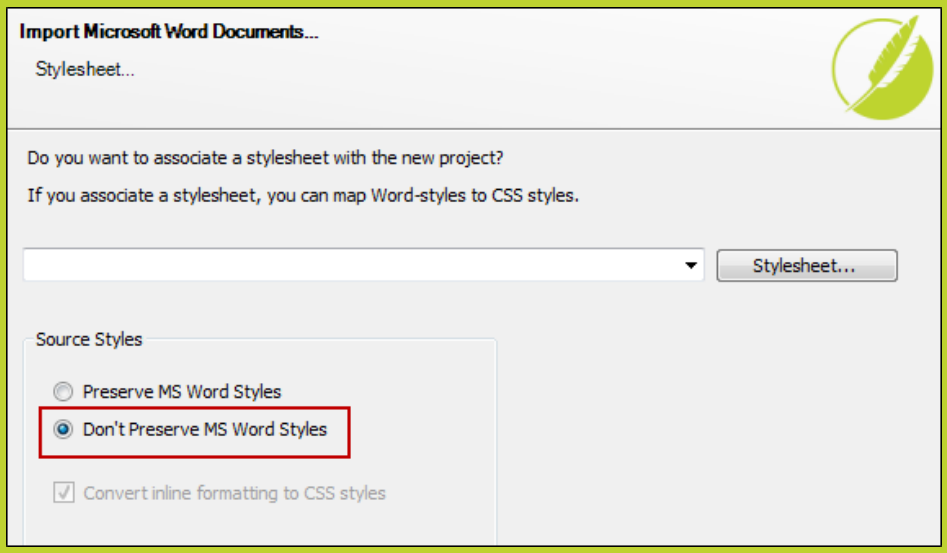
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading 1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading 1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

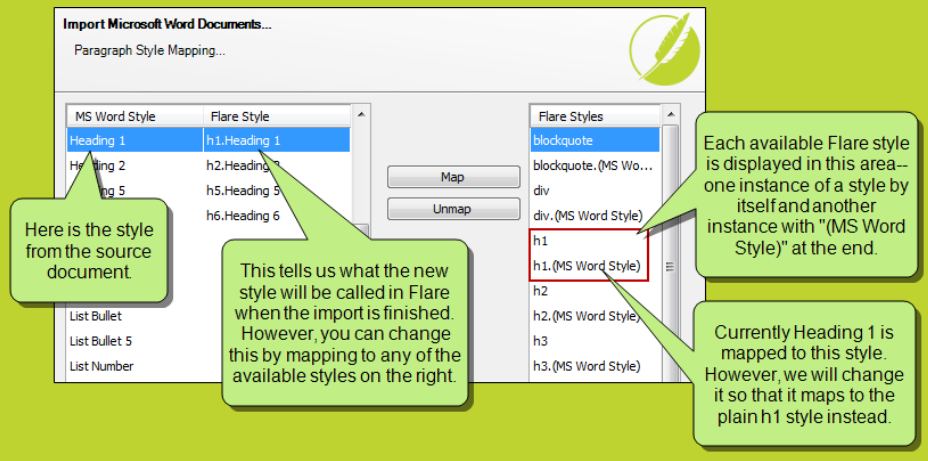
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

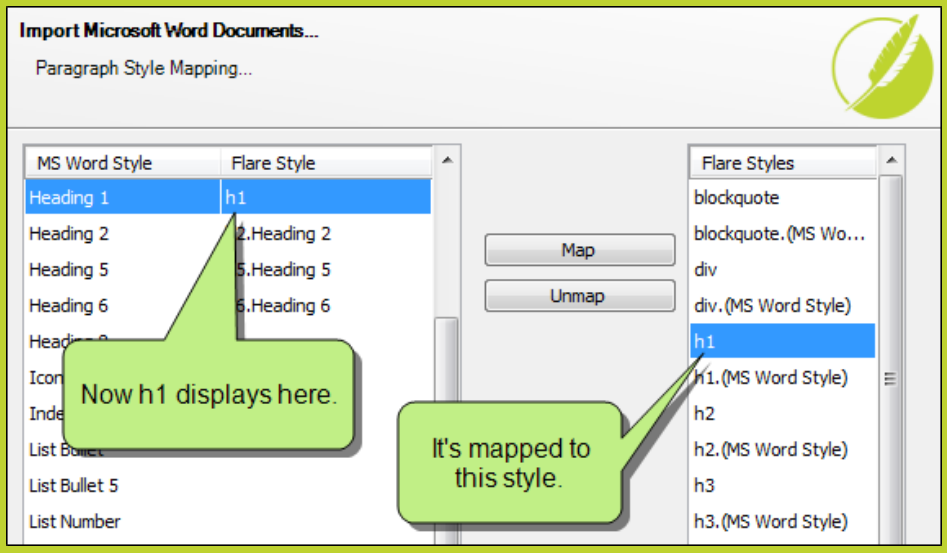
Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *not preserve* your Word styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading 1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1.(Word Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

18. Click **Next**.
19. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style, click the style in the **MS Word Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

20. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
21. When you are finished previewing the files to be created, click **Accept**. The files are imported and the project is created.

HOW TO IMPORT WORD FILES INTO AN EXISTING PROJECT—VIA THE WIZARD

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **Project>Import>MS Word Documents**.
   - **Menu** Select **File>Import Project>MS Word Documents**.

   The Import Microsoft Word Wizard opens.
2. Select **Import into this project**.
3. Click **Next**.
4. Click .
5. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DOC, DOCX, or RTF files.

   Note: DOCX is Microsoft Word's platform-independent, open XML format. You must have Microsoft Word 2007 installed in order to import this file type.
6. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original Word documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original Word documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

7. Click **Next**. The Word documents are scanned and the next page of the wizard opens. The paragraph styles used in the Word documents are shown on the left side of the wizard page ("Used Word Styles").

> **Note:** You can skip the rest of the wizard pages by clicking **Finish**.

8. If you want Flare to split the Word documents into smaller topics based on any of the styles shown on the left side of the page ("Used Word Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

> **E X A M P L E**
>
> If you have a style called "Heading 2" in your Word documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

9. Click **Next**.
10. You can use the following options as necessary:

   - **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
   - **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
   - **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a heading in each topic that is connected to another topic included in the split. That text will be used in the link

instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.

- **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.

- **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
  - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
- **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the Word documents without any content.
  - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your Word source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import Word files when you attempt to build output.
- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.
- **Convert all tables to "Auto-Fit to Contents"** Select this option if you want to automatically set tables to "Auto-Fit to Contents" when they are imported into Flare. This simply ensures that column widths are not specified on the imported tables.
- **Create a Page Layout for each section header/footer** Select this option if you want Flare to create new page layouts when you import Word documents that have section breaks, along with headers or footers. For each new section in the Word document that has a different header or footer than the previous section, Flare creates a unique page layout. After the import is finished, you can open and edit the page layouts if necessary. You can then create chapter breaks for your print-based output and assign these page layouts to the different topics in the output.
- **Automatically set topic title** If this option is enabled, the properties setting for the topic title automatically uses the first heading in the topic. Therefore, if you change the heading text in the future, the topic title changes automatically as well. If this option is disabled, the properties setting for the topic title explicitly uses the first topic heading text found during the import, and it remains so unless you manually change it later.

11. Click **Next**.

12. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your Word files.

    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the **Stylesheet** button. Then navigate to the stylesheet and select it.
    - **Preserve MS Word Styles** This retains any style formatting from your Word documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).
    - **Don't Preserve MS Word Styles** This does not keep the style formatting used in the Word documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that either keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).
    - **Convert inline formatting to CSS styles** This creates new styles based on any local formatting that exists in the Word documents.

    E X A M P L E

    If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.

    Tip: To avoid having several character styles in Flare that you do not want, it is a good idea to clean up your source documents as much as possible before importing.

13. Click **Next**.

14. Use this page to map paragraph styles from the Word documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. To map a style, click the style in the **MS Word Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the Word styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(MS Word Style).

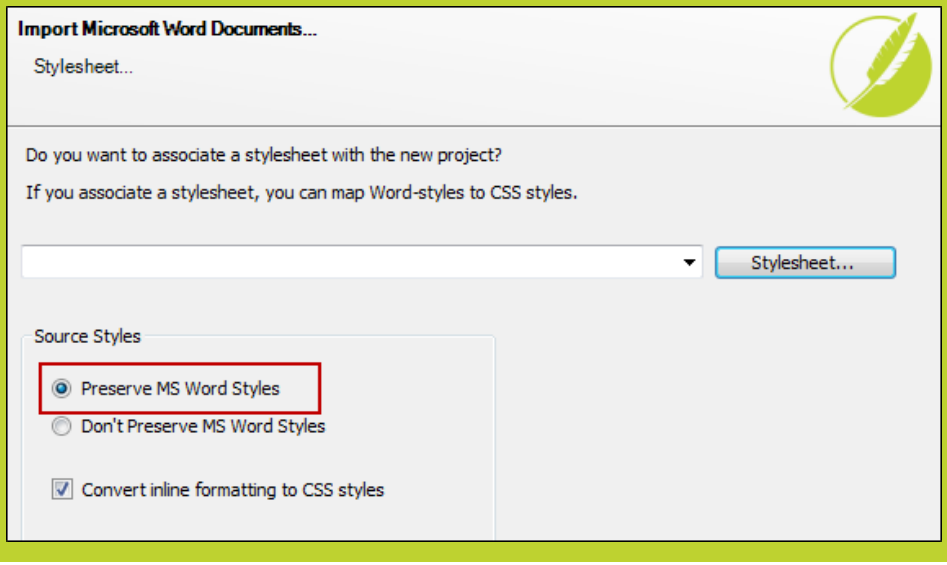
The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

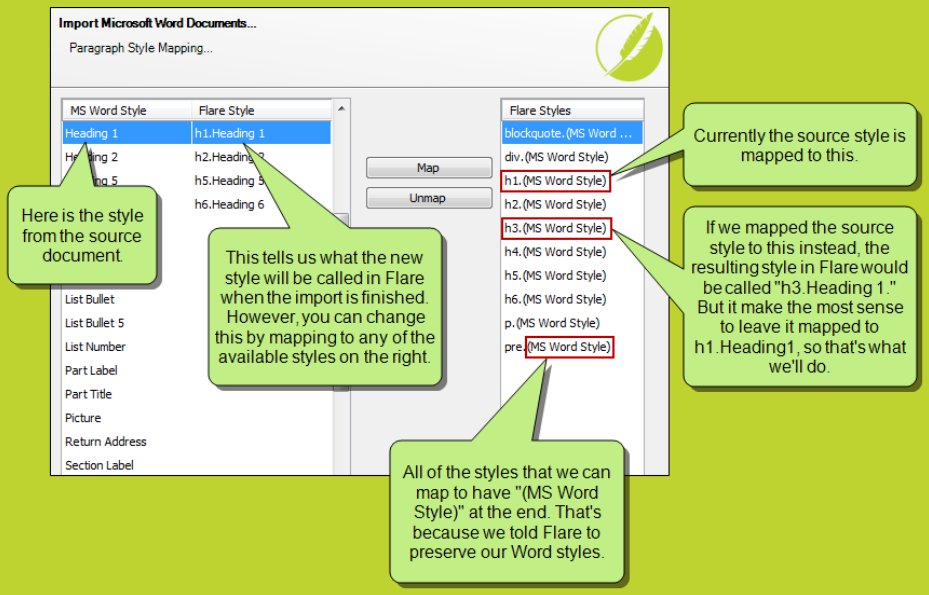
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *preserve* your Word styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading 1 in the source document is still using Arial 14 pt and is red, just like it was before in Word. However, the style is now called "h1.Heading 1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the Word style, Flare added it as a child under its parent, h1.

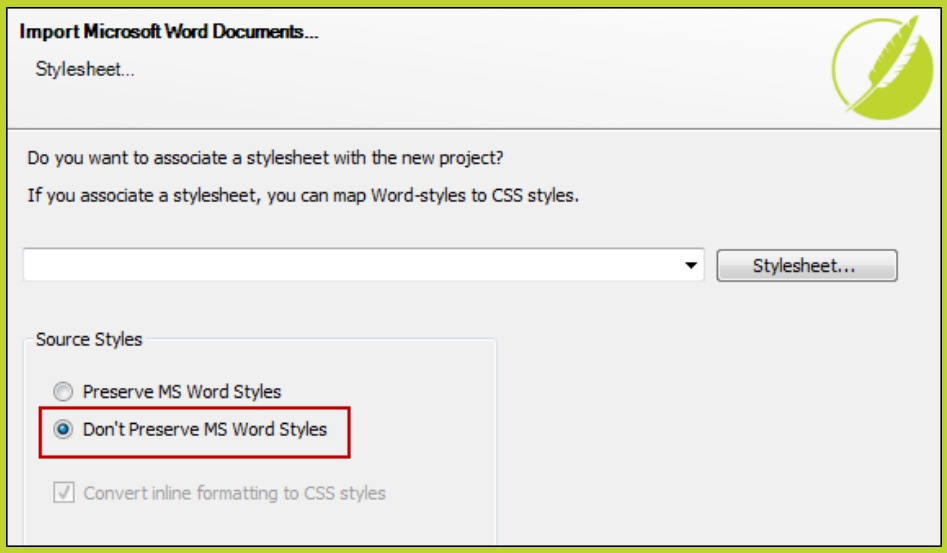
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading 1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading 1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

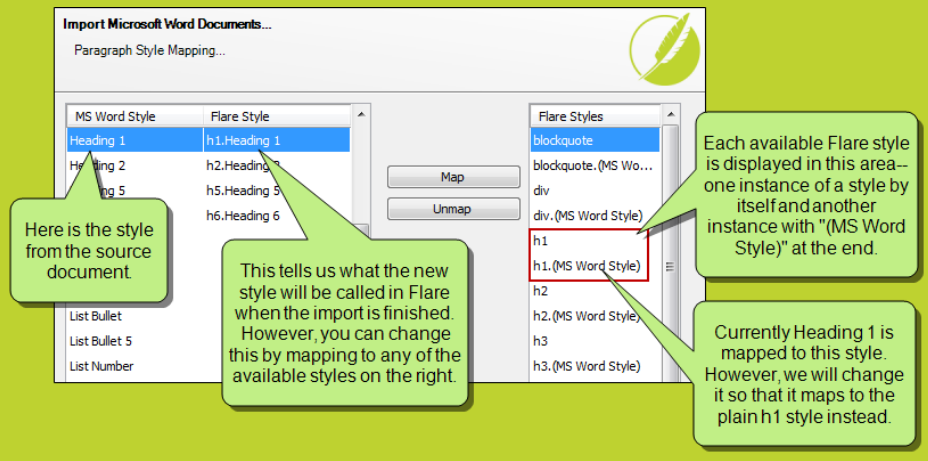
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

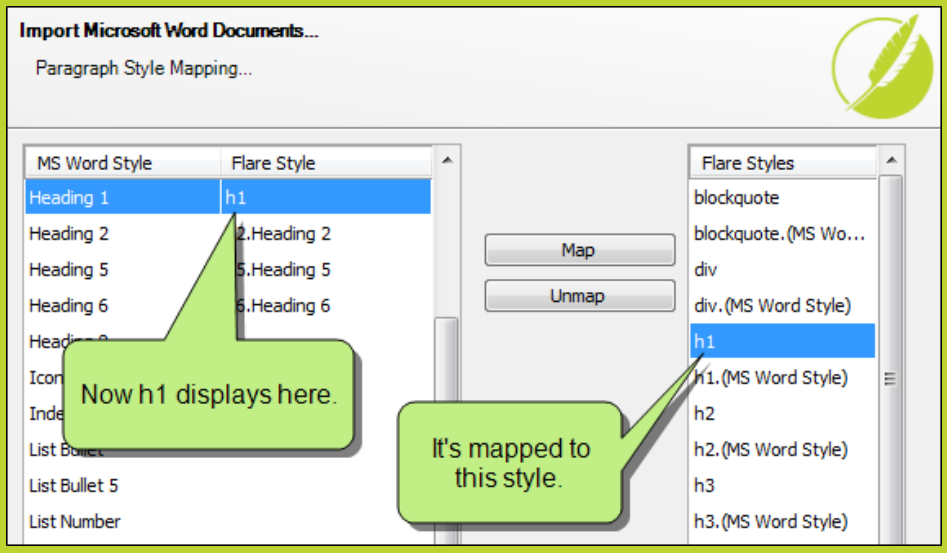
Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *not preserve* your Word styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading 1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1.(Word Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

15. Click **Next**.
16. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style, click the style in the **MS Word Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

17. Click **Finish**. The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
18. When you are finished previewing the files to be created, click **Accept**.

HOW TO IMPORT WORD FILES INTO AN EXISTING PROJECT—VIA THE IMPORT EDITOR

1. Do one of the following:

   IF YOU DO NOT YET HAVE A WORD IMPORT FILE

   a. Complete the steps for adding an import file.

   b. Open the file in the Word Import Editor.

   IF YOU ALREADY HAVE A WORD IMPORT FILE

   a. In the Project Organizer, expand the **Imports** subfolder.

   b. Double-click the appropriate import file. The file opens in the Word Import Editor to the right.

2. Select the **Source Files** tab.

3. Click .

4. In the dialog that opens, find and select the files you want to import. You can hold the **SHIFT** key to select a range, or you can hold the **CTRL** key to select individual items. When you are finished, click **Open**.

   You can select DOC, DOCX, or RTF files.

   > **Note:** DOCX is Microsoft Word's platform-independent, open XML format. You must have Microsoft Word 2007 installed in order to import this file type.

5. You can use the following options as necessary:

	This opens the file that is selected in the list.
Link Generated Files to Source Files	This creates a connection between the original Word documents and the files that are created as a result of the import. This is useful if you want to continue editing the content in the original Word documents, instead of editing in the Flare project. Flare recognizes when changes have been made to the source documents and reminds you to re-import the documents to ensure the Flare project also reflects the changes. If you use this option, a link icon is added to the top of a linked file in the Flare interface. This lets you know that you need to edit the source file, rather than editing this file. If you remove the connection to the source file, this icon no longer displays on the file. Please note that if you have bound the project to source control, the icons used for source control take precedence over the link icon.
	This removes the selected document(s) from the list.

	This moves the selected document higher in the list (if you have more than one document to import). The document at the top is used for the name of the content folder holding the imported topics in Flare. Also, the order determines how the imported documents are arranged in the Flare TOC that is created as a result.
	This moves the selected document lower in the list (if you have more than one document to import).

6. Select the **New Topic Styles** tab.
7. If you want Flare to split the Word documents into smaller topics based on any of the styles shown on the left side of the page ("Used Word Styles"), double-click that style to move it to the right side of the page ("New Topic Styles").

> **E X A M P L E**
>
> If you have a style called "Heading 2" in your Word documents, you might want new topics to be created whenever Flare finds a Heading 2 style in a document. So you would double-click Heading 2 and move it to the right side of the page.

8. Select the **Options** tab.
9. You can use the following options as necessary:
   - **Add "Topic Continued" links when appropriate** Select this option to place a "Topic Continued" link at the bottom of pages when a long topic has been split into multiple ones.
   - **Add "Topic Continued From" links when appropriate** Select this option to place a "Topic Continued From" link at the top of continued pages when a long topic has been split into multiple ones.
   - **Cross-Reference Format** Use this field to specify the format for the "Topic Continued" and "Topic Continued From" links. Flare provides a cross-reference format for you—(continued in {title}) or (continued from {title}). With this cross-reference format, the link contains the words "continued in" or "continued from" within parentheses, followed by the text of the first paragraph in the connected topic. If you do not want the link to use that particular text, you have a couple of options. First, in Flare, you could manually enter a heading in each topic that is connected to another topic included in the split. That text will be used in the link instead (after you update the cross-references in Flare). Another option is to modify the format by clicking the **Edit** button.
     - **Edit** If you want to modify the cross-reference format provided, click this button, which opens the Cross-Reference Format dialog.

- **Split Long Topics** Select this option if you have long sections in your source documents and want to make sure that they are converted to multiple topics (rather than one very long topic).
  - **Threshold** Enter the maximum number of characters to be converted to a topic before a new topic is created. Flare will break the topic at the nearest paragraph to the threshold value. That way, a new topic will not start in the middle of a sentence or word, but at the beginning of a paragraph.
- **Avoid Creating 'Empty' Topics** Select this option if you want to ensure that new topics are not created when large sections are found in the Word documents without any content.
  - **Threshold** Enter the maximum number of empty character spaces allowed in a topic. If this number is exceeded, Flare will not create a new topic from that empty space.
- **Auto-reimport before 'Generate Output'** This is also known as "Easy Sync." If you created a connection between your Word source files and the Flare project earlier in the wizard, you will likely make future content changes in the source files. When you make such changes, the source files need to be re-imported into the project so that they can be included in the output. You have the option of re-importing the files manually. However, you can also tell Flare to do this for you automatically, so that you do not have to. Select this option if you want Flare to automatically re-import Word files when you attempt to build output.
- **Approximate Filename Length** Enter the maximum number of characters to use for naming new topic files that are automatically created after splitting a long topic. The default is 24.
- **Convert all tables to "Auto-Fit to Contents"** Select this option if you want to automatically set tables to "Auto-Fit to Contents" when they are imported into Flare. This simply ensures that column widths are not specified on the imported tables.
- **Create a Page Layout for each section header/footer** Select this option if you want Flare to create new page layouts when you import Word documents that have section breaks, along with headers or footers. For each new section in the Word document that has a different header or footer than the previous section, Flare creates a unique page layout. After the import is finished, you can open and edit the page layouts if necessary. You can then create chapter breaks for your print-based output and assign these page layouts to the different topics in the output.
- **Automatically set topic title** If this option is enabled, the properties setting for the topic title automatically uses the first heading in the topic. Therefore, if you change the heading text in the future, the topic title changes automatically as well. If this option is disabled, the properties setting for the topic title explicitly uses the first topic heading text found during the import, and it remains so unless you manually change it later.

10. Select the **Stylesheet** tab.
11. Use this page to specify whether the imported topics should be associated with a stylesheet and/or styles from your Word files.
    - **Stylesheet** If you already have a CSS file that you want to associate with the imported files, click the

Stylesheet button. Then navigate to the stylesheet and select it.

- **Preserve MS Word Styles** This retains any style formatting from your Word documents so that you can continue to use that look and feel in Flare. For example, if you use a style called "Heading 1" in your source documents and that style is blue, it remains blue after you finish the import to Flare and the new style is created. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare).
- **Don't Preserve MS Word Styles** This does not keep the style formatting used in the Word documents. For example, if you use a style called "Heading 1" in your source documents and that style is blue, that color (and any other settings for that style) are not kept after you finish the import to Flare. You will still have styles associated with your content, but it will not look like it did in the source documents. Also, selecting this option affects which mapping styles are available as you continue to make your import selections. If you select this option, you can map the Word styles to new Flare styles that either keep the name of the Word style (e.g., *Heading 1* becomes *h1.Heading 1* in Flare) or do not (e.g., *Heading 1* becomes *h1* in Flare).
- **Convert inline formatting to CSS styles** This creates new styles based on any local formatting that exists in the Word documents.

EXAMPLE

If you have applied bold and italic formatting to some text (without using a style), Flare will create a new style based on that formatting.

Tip: To avoid having several character styles in Flare that you do not want, it is a good idea to clean up your source documents as much as possible before importing.

12. Select the **Paragraph Styles** tab.
13. Use this page to map paragraph styles from the Word documents to Flare's paragraph styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. To map a style, click the style in the **MS Word Styles** column on the left, click a style in the **Flare Styles** section on the far right, and then click the **Map** button. If you previously elected not to preserve the Word styles, it is recommended that you map to a standard CSS parent style—e.g., map your first-level heading style to h1, not to h1.(MS Word Style).

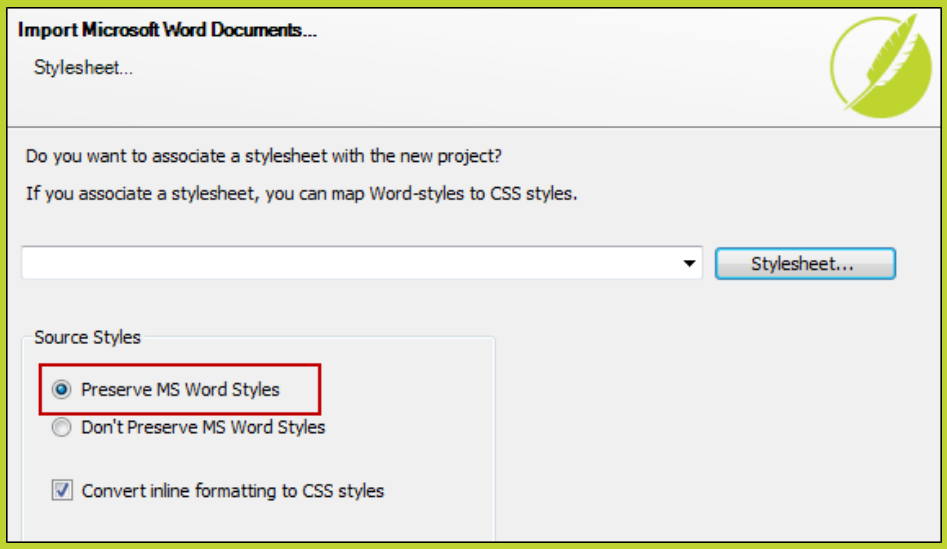
    The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

EXAMPLE—PRESERVE STYLES

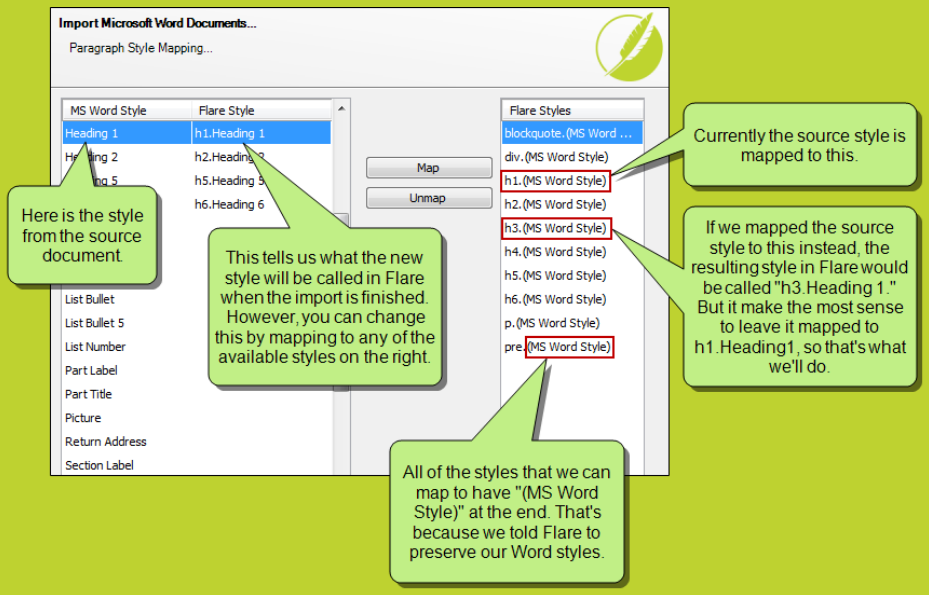
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *preserve* your Word styles.



The next page of the wizard looks something like this.



When you finish importing, the content that was associated with Heading 1 in the source document is still using Arial 14 pt and is red, just like it was before in Word. However, the style is now called "h1.Heading 1." In the world of cascading style sheets (CSS)—which is what Flare uses for controlling the look of content—you've created a class of the h1 style (h1 is the standard style for first-level headings). But because you wanted to keep the look of the Word style, Flare added it as a child under its parent, h1.

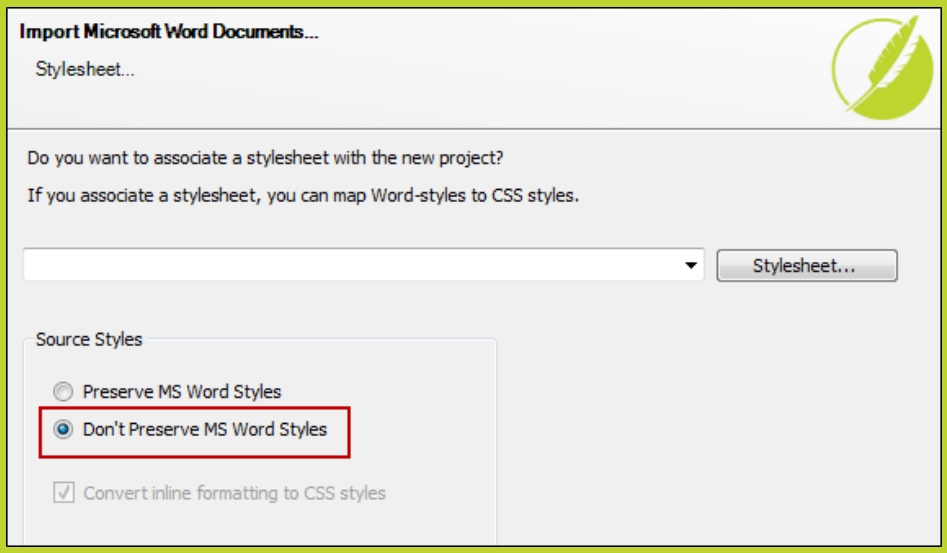
If you make any future changes in Flare to the h1 style, they will trickle down to the h1.Heading 1 child (unless the child style has explicit settings that conflict with the parent). You can also apply style properties directly to the h1.Heading 1 child. So while it is generally a good idea to use standard CSS parent styles (such as h1) when possible in Flare, the mapping performed in this import process—and the subsequent creation of a child style—lets you keep the Arial 14 pt red look.

EXAMPLE—DO NOT PRESERVE STYLES

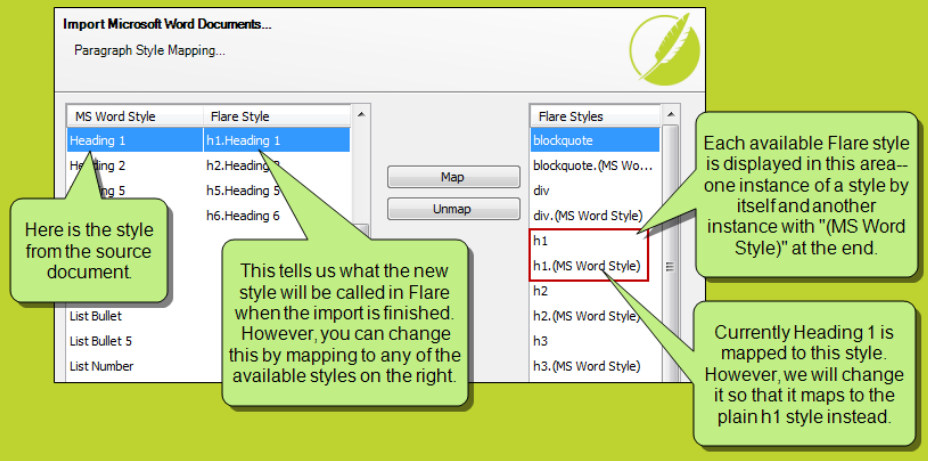
Let's say you have a style in your Word source document called "Heading 1" that is using Arial 14 pt and is red, like this.

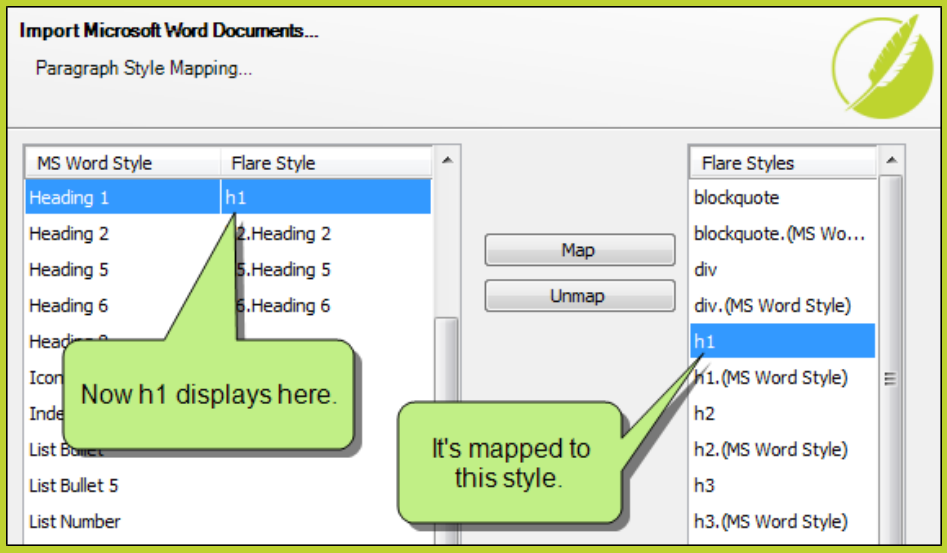
Hello

During the process of importing your Word document using the Import Microsoft Word Wizard, you select the option to *not preserve* your Word styles.



The next page of the wizard looks something like this.





When you finish importing, the content that was associated with Heading 1 in the source document is no longer using Arial 14 pt, red. Instead, it looks something like this.

Hello

Also, the style is now called "h1." (Keep in mind that, even if you had mapped the style to "h1.(Word Style) in this case, the formatting would still be removed.)

So although the formatting was not retained, you were able to map to the standard CSS style for first-level headings—h1 .

14. Select the **Character Styles** tab.
15. Use this page to map character styles from the source documents to Flare's character styles, including those from the stylesheet you may have selected. Your Word style will adopt the name of that style. This works the same as the feature for mapping paragraph styles, except it has to do with character-level styles. To map a style, click the style in the **MS Word Style** column, click a style in the **Flare Styles** section, and then click the **Map** button.

The style is added to the Flare Styles column. When you are finished importing the documents and the new Flare project is loaded, the content that had been associated with the style in the Word document will now be associated with a new style that you mapped it to.

16. In the local toolbar click **Import** (if this is the first time importing files) or **Reimport** (if files have been imported previously). The Accept Imported Documents dialog opens. The files that will be created as a result of the import are listed on the left. A preview of each file can be seen to the right when you click the file.
17. When you are finished previewing the files to be created, click **Accept**.

Note: The Imported Files tab displays the files that have been imported previously.

Note: Flare supports Microsoft Word 2003 and newer versions.

Note: When you import Word files, a stylesheet (at least a generic one) will always be created and initially be applied at the project level, whether you choose to preserve Word styles or not. When a master stylesheet is applied at either the target or project level, you cannot associate stylesheets at the topic level (unless you remove the master stylesheet association at the project and/or target level).

Importing from Source Control

Following are the steps for importing a project from source control.

HOW TO IMPORT A PROJECT FROM SOURCE CONTROL

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>New Project>Import Project**.
   - **Menu** Select **File>Import Project>From Source Control**.

   The Import Project From Source Control Wizard opens.
2. Follow one of these sets of steps, depending on the source control application being used.

   IF USING TEAM FOUNDATION SERVER

   a. From the drop-down, select **Microsoft Team Foundation Server**.

   b. In the **Server** field, enter the name of the computer or the IP address of the server. You can also click Browse to select a "Team Project Collection." If you click this button, the Select Team Foundation Server Project Collection dialog opens, and you can do the following.

      i. To add a server, click . The Add Team Foundation Server dialog opens.

      ii. Enter the name or URL of the server.

      iii. Enter the path and port number.

      iv. Select the protocol (HTTP or HTTPS).

      Note: You may need to obtain this information from your system administrator. Click **OK** in the dialogs until you return to the main wizard page.

      Note: If you run into issues in this dialog, chances are your user permissions do not have access to all of the collections on the TFS server. Please try to enter in the server/collection path manually.

   c. Next to the **Team Project** field, click **Browse**. If the Log In dialog opens, complete the **User name** and **Password** fields.

   d. Click on the Team Foundation project.

   IF USING VISUAL SOURCESAFE

   a. From the drop-down, select **Microsoft Visual SourceSafe**.

b. Next to the Database field, click **Browse**.

c. Find the appropriate SourceSafe database and double-click the **INI** file.

IF USING SUBVERSION

a. From the drop-down, select **Subversion**.

b. In the **Server** field, enter the name of the computer or server IP address.

IF USING PERFORCE

a. From the drop-down, select **Perforce**.

b. In the **Server** field, enter the server address.

c. Next to the **User** field, click **Browse**. The Select User dialog opens.

d. Select a user and click **OK**. (Users are set up separately in Perforce.)

e. Next to the **Stream** field, click **Browse**. The Browse Streams dialog opens.

f. Select a stream (i.e., the name of your Flare project).

g. Click **OK**.

IF USING ANOTHER SOURCE CONTROL APPLICATION

a. From the drop-down, select **Third Party Plug-in**.

b. In the **Provider** field, select the appropriate source control program from the list.

c. (Optional) You might be able to click the Advanced button to enter more options. This depends on whether the provider has enabled advanced options; if not, the button is disabled. The advanced options available to you are different for each provider.

3. Click **Next**.
4. Next to the **Project file** field, click **Browse**. The Browse Source Control Files dialog opens.
5. Find and click on the Flare project file (FLPRJ) that you want to import. (You may need to log in with your user name and password.)
6. Click **OK**.
7. Click **Next**.
8. In the **Project name** field, the name of the project being imported is displayed. It is recommended that you leave the name as it is, especially if you are working with other authors on the project. However, you can enter a different project name if you want.
9. In the **Project folder** field, either accept the default location for the new project or click ... to browse for and select a folder.
10. Click **Finish**. The project is imported and loaded into Flare.

CHAPTER 2

Merging Projects

This feature is supported in all online outputs (DotNet Help, Eclipse Help, Microsoft HTML Help, HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus).

In Flare you can merge projects in many ways.

This chapter discusses the following:

Linking to CHM Files 116
Merging Flare Projects Using Targets 117
Merging Output from DNH Projects 121
Merging Output from Flare WebHelp 128
Merging Output from HTML Help Projects (CHM Files) 135
Merging Output at Runtime Using HTML5 139
Merging Output at Runtime Using WebHelp Plus 147
Displaying Merged Navigation in HTML Help Child Outputs 155

Linking to CHM Files

You can include a CHM file in the TOC in your project so that it can be opened in your output.

If you do this when generating Microsoft HTML Help, the linked CHM will be merged with the TOC in the project.

If you do this when generating DotNet Help, HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, or WebHelp Plus, the user will be able to download the CHM.

Regardless of the type of online output you are generating, you can find the steps for linking to a CHM file in the following topic: "Merging Output from HTML Help Projects (CHM Files)" on page 135.

Note: Links to CHM files can be configured in two ways. First, links can point to the CHM file in general. Second, they can point to a *specific topic* within the CHM. The first method is supported in all of the online outputs mentioned above. The second method is supported only in Microsoft HTML Help output.

Merging Flare Projects Using Targets

Use this merging method if you have access to all of the Flare project files to be merged. You can merge projects together by linking targets with the same output type (e.g., linking DotNet Help to DotNet Help, linking Microsoft HTML Help to Microsoft HTML Help, linking WebHelp to WebHelp).

HOW TO MERGE MULTIPLE FLARE PROJECTS ON YOUR COMPUTER

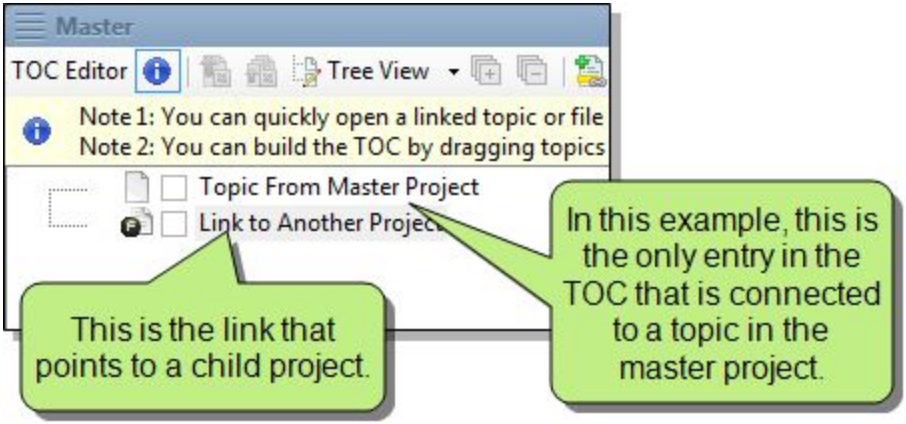
1. Open the TOC in the Flare project that will serve as the parent project.

   Note: You can also merge projects by linking them from a browse sequence, as well as from a TOC. If you want to do this, simply follow these same steps after opening your browse sequence.

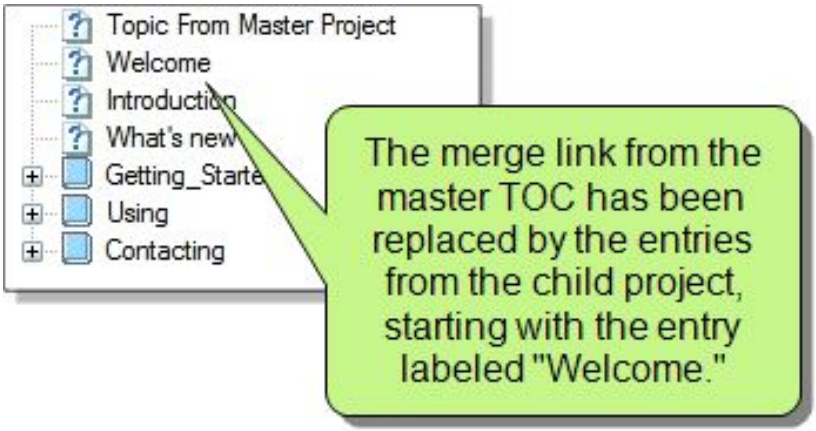
2. In the TOC Editor, select the TOC entry or book where you want to link the output from another Flare project. (The TOC of the linked project will be inserted at that spot in the "master" TOC.)
3. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
4. Select the **General** tab.
5. Click **Select Flare Project and Target**. The Link to Flare Project and Target dialog opens.
6. Do one of the following:
   - Click **Browse for Project**. Then click **Project File** and find/select a Flare project to which you want to link.

     OR
   - Click **Select Recent Project**. Then select a recently opened Flare project from the list on the right.
7. From the **Select Target** field, choose a specific target in the Flare project. Make sure you select a target that uses the same output type (e.g., DotNet Help, Microsoft HTML Help, WebHelp) as the target that you will build in the current project.
8. Click **OK**.
9. (Optional) If you want the merged TOC to replace the entry that you selected, do the following.
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

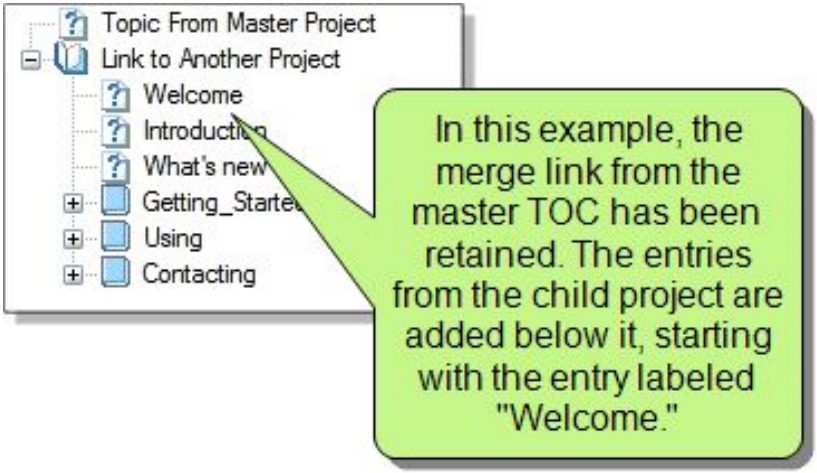
For example, you might have a TOC in your parent project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



10. In the Properties dialog, click **OK**.
11. Build the output for the target in your parent project. The output files for the projects are merged. When you open the TOC in the output, you will see the integrated TOC from the linked project.

Note: There are occasions when projects cannot be merged because the file name is the same in two or more of the linked projects (e.g., multiple DotNet Help projects all named "MyDotNetProject.mchelp," multiple WebHelp projects all named "MyWebProject.mcwebhelp," or multiple Microsoft HTML Help projects all named "MyHtmlHelp.chm"). The way to solve this dilemma is to make sure each linked project has a different output file name. To do this: (1) open the target, (2) select the **General** tab in the Target Editor, and (3) enter a unique name in the **Output File** field.

Note: If you are working with HTML Help and import a RoboHelp project that has been merged with other CHM files, the linked CHM files are placed in a special subfolder in the Content Explorer (Resources\CHMSup-port).

Note: If you have previously merged Flare projects with WebHelp outputs and then decide to generate WebHelp Plus output from the master project instead, you must make sure that the child projects linked to the master contain at least one WebHelp Plus target each. Even if you keep the link from the master project to a WebHelp target in the child, the child project must also have a WebHelp Plus target.

Note: If you merge projects, synonym files will remain separate in each project. For example, if you create synonyms in Project A but not in project B, only the topics from Project A will use the synonyms when users perform searches in the output.

Note: You can merge HTML Help so that the navigation (table of contents, index, search) for each CHM file is displayed, regardless of which CHM file you open (parent or child). This can be done in a couple of ways, depending on whether you have access to all of the Flare projects or to the CHM files only.

Merging Output from DNH Projects

Use this merging method if you are developing DotNet Help and you want to merge the output files from your parent project with the DotNet Help output files from an external project. This method is useful, for example, if another author is working on the external DotNet Help project to which you are linking and you only have access to the output files (not the project files).

Note: There are occasions when projects cannot be merged because the file name is the same in two or more of the linked projects (e.g., multiple DotNet Help projects all named "MyDotNetProject.mchelp," multiple WebHelp projects all named "MyWebProject.mcwebhelp," or multiple Microsoft HTML Help projects all named "MyHtmlHelp.chm"). The way to solve this dilemma is to make sure each linked project has a different output file name. To do this: (1) open the target, (2) select the **General** tab in the Target Editor, and (3) enter a unique name in the **Output File** field.

Before your master project can be merged with an external Help system, the output folders and files for the external (or "child") project need to be copied to a folder called "Subsystems" in the same location where the master project is being published. The child's target folder containing these output files also needs to be included at this location.

E X A M P L E

Let's say the master project's output files are being published to the following:

C:\MyHelpSystem\Master\Master.mchelp

In that case, the output folders and files for the child project should be copied to the following folder:

C:\MyHelpSystem\Master\Subsystems

For example: C:\MyHelpSystem\Master\Subsystems\AnotherHelpSystem\AnotherHelpSystem.mchelp

In this example, "AnotherHelpSystem" is the target folder that you generated, which holds the output folders and files, and "AnotherHelpSystem.mchelp" is the main output file for the child project that was created. Not only does the main .mchelp output file need to be included in this folder, but also the ancillary output folders and files (e.g., Content, Data).

You can merge DotNet Help outputs using an automatic or manual method.

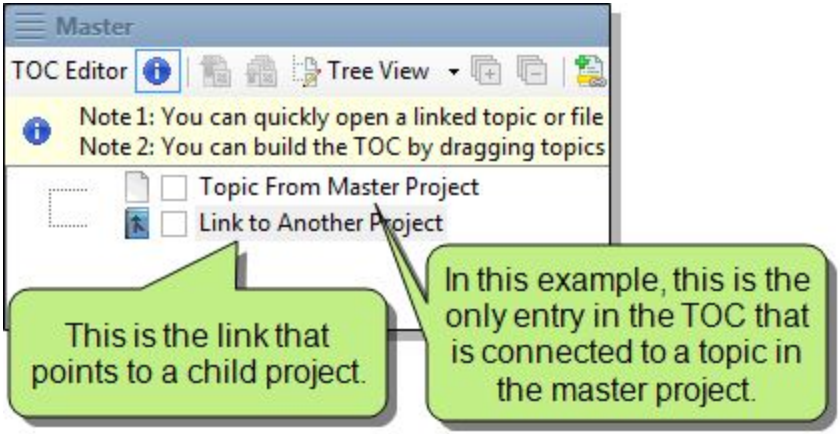
HOW TO MERGE OUTPUT FROM DOTNET HELP PROJECTS (AUTOMATIC METHOD)

1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.

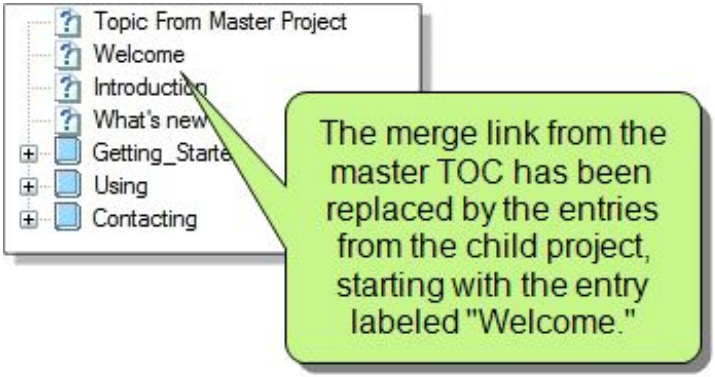
> **Note:** You can also merge projects by linking them from a browse sequence, as well as from a TOC. If you want to do this, simply follow these same steps after opening your browse sequence.

3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child DotNet Help project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Click **Browse**.
7. In the **Files of type** field, click the down arrow and select **MadCap DotNet Help Files (*.mchelp)**. Then find and select the MCHELP output file to which you want to link.
8. Click **OK**. The full path to the file is displayed in the field.
9. (Optional) If you want the merged TOC to replace the entry that you selected, do the following.
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

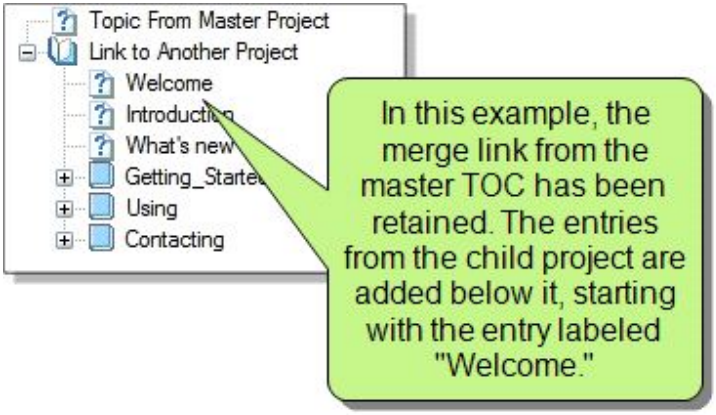
For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



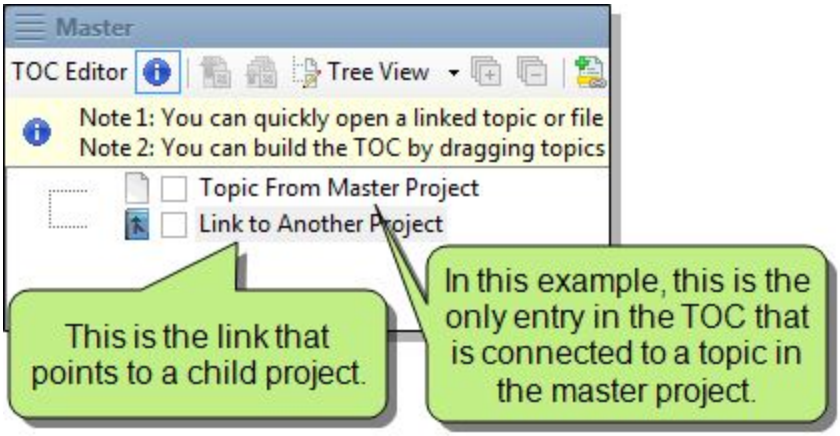
10. In the Properties dialog, click **OK**.
11. Build the output for the DotNet Help target in your "master" project.

    Flare automatically creates the Subsystems folder for you. It also copies the target folder and .mchelp file (along with the ancillary output files and folders) to the appropriate destination. When you open the TOC in the output, you will see the integrated TOC from the linked project.

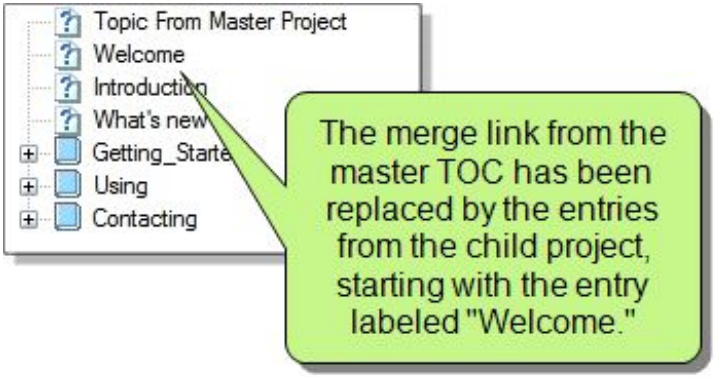
HOW TO MERGE OUTPUT FROM DOTNET HELP PROJECTS (MANUAL METHOD)

1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.
3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child DotNet Help project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Type the name of the child target and main output file (e.g., AnotherHelpSystem/AnotherHelpSystem.mchelp).
7. Click **OK**.
8. (Optional) If you want the merged TOC to replace the entry that you selected, do the following.
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

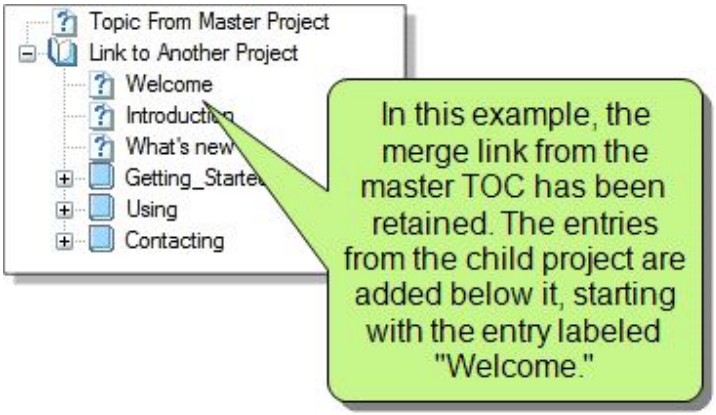
   For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



9. In the Properties dialog, click **OK**.
10. Build the output for the DotNet Help target in your "master" project.
11. In the master Flare project, right-click on the target and select **Open Output Folder**. Within that folder, manually create another folder and name it Subsystems.

12. In the child Flare project, select **Project>Open Output Folder** (if using the ribbon view) or **Build>Open Output Folder** (if using the menu view).
13. Copy the target folder to be merged with the master project.
14. Paste the target folder within the Subsystems folder that you created in the master output folder.

Note: If you merge projects, synonym files will remain separate in each project. For example, if you create synonyms in Project A but not in project B, only the topics from Project A will use the synonyms when users perform searches in the output.

Merging Output from Flare WebHelp

This feature is supported in all WebHelp outputs (HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus).

Use this merging method if you are developing one of the WebHelp outputs (WebHelp, HTML5, WebHelp AIR, WebHelp Mobile, WebHelp Plus) and you want to merge the output files from your parent project with the WebHelp output files from an external Flare project. This method is useful, for example, if another author is working on the external WebHelp project to which you are linking and you only have access to the output files (not the project files).

Before your master project can be merged with an external Help system, the output folders and files for the external (or "child") project need to be copied to a folder called "Subsystems" in the same location where the master project is being published. The child's target folder containing these output files also needs to be included at this location.

E X A M P L E

For example, if the master project's output files are being published to C:\MyHelpSystem\Master\Master.mcwebhelp, the output folders and files for the child project should be copied to the folder C:\MyHelpSystem\Master\webSubsystems (e.g., C:\MyHelpSystem\Master\Subsystems\AnotherHelpSystem\AnotherHelpSystem.mchelp). In this example, "AnotherHelpSystem" is the target folder that you generated, which holds the output folders and files, and "AnotherHelpSystem.mcwebhelp" is the main output file for the child project that was created. Not only does the main .mcwebhelp output file need to be included in this folder, but also the ancillary output folders and files (e.g., Content, Data, Skin).

You can merge WebHelp outputs using an automatic or manual method.

HOW TO MERGE OUTPUT FROM WEBHELP PROJECTS (AUTOMATIC METHOD)

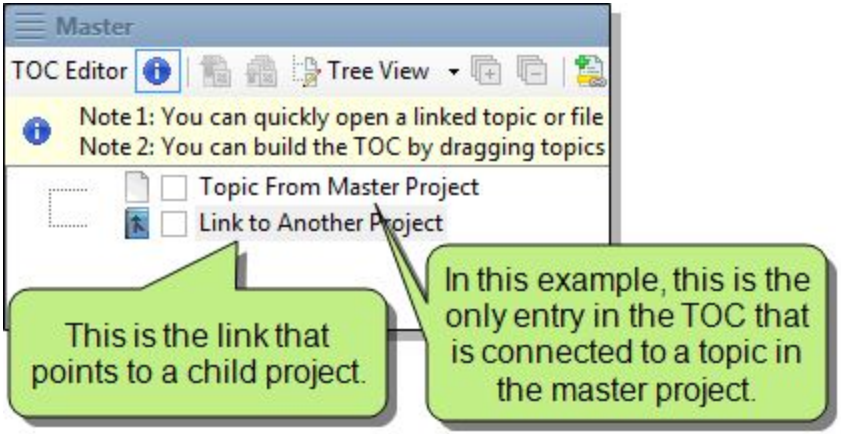
1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.

> **Note:** You can also merge projects by linking them from a browse sequence, as well as from a TOC. If you want to do this, simply follow these same steps after opening your browse sequence.

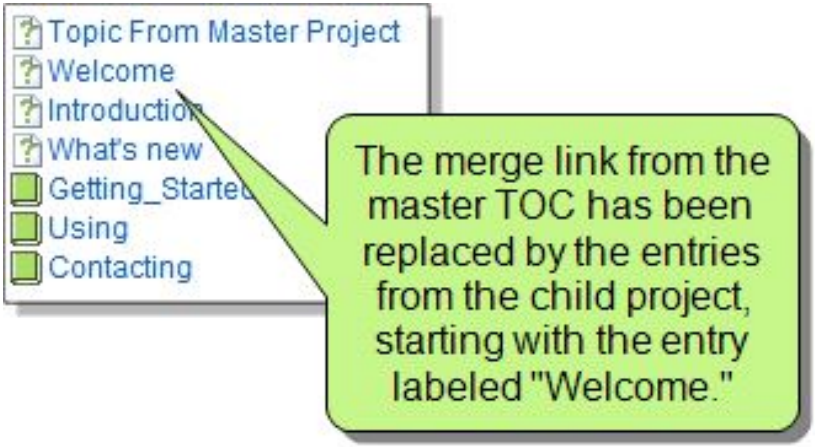
3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child WebHelp project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Click **Browse**.
7. In the **Files of type** field, click the down arrow and select **MadCap Web Help Files (*.mcwebhelp)**. Then find and select the MCWEBHELP output file to which you want to link.
8. Click **OK**.

   The full path to the file is displayed in the field.
9. (Optional) If you want the merged TOC to replace the entry that you selected, do the following:
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

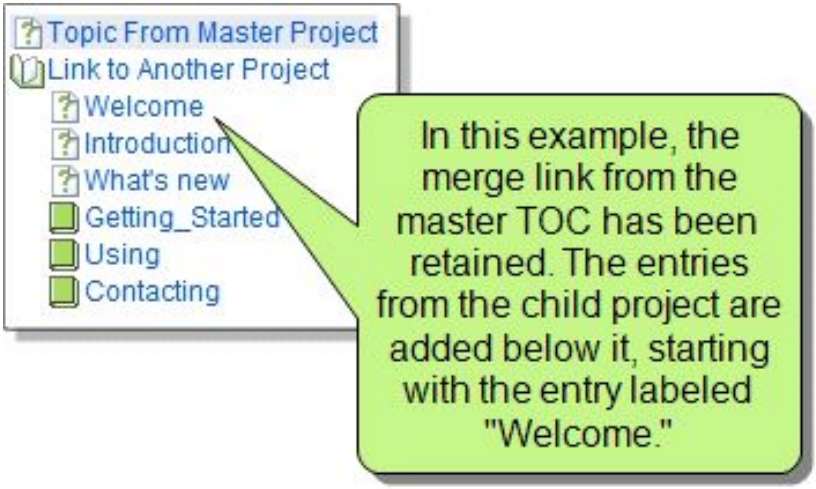
For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



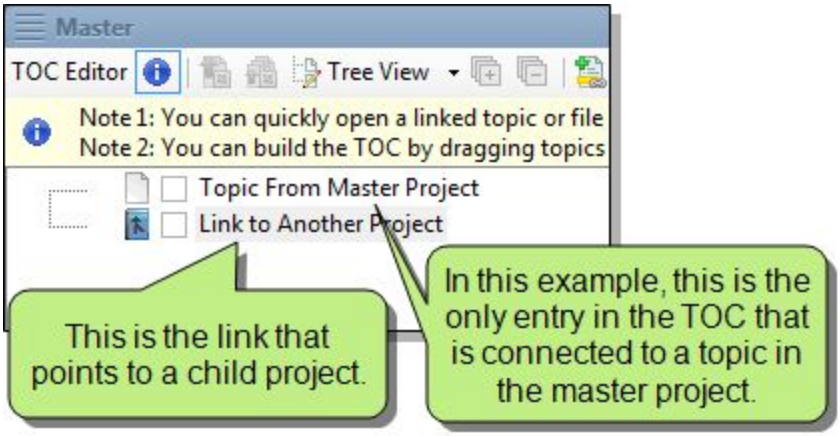
10. In the Properties dialog, click **OK**.
11. Build the output for the WebHelp target in your "master" project.

    Flare automatically creates the Subsystems folder for you. It also copies the target folder and .mcwebhelp file (along with the ancillary output files and folders) to the appropriate destination. When you open the TOC in the output, you will see the integrated TOC from the linked project.

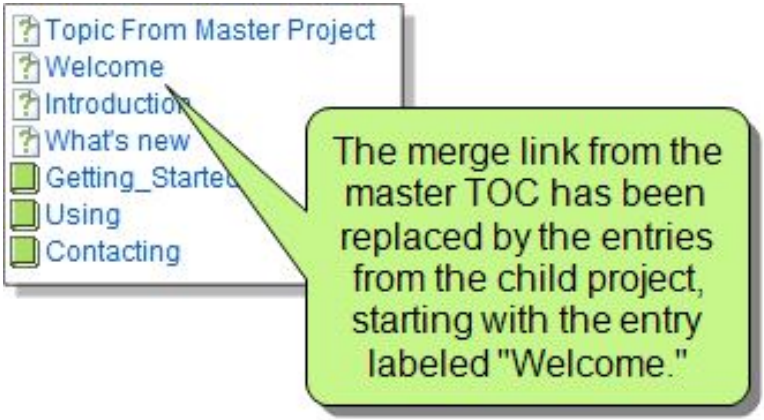
HOW TO MERGE OUTPUT FROM WEBHELP PROJECTS (MANUAL METHOD)

1. Generate the output for the child project.
2. Open the TOC in the Flare project that will serve as the parent project.
3. In the TOC Editor, select the TOC entry or book where you want to link the output from the child WebHelp project. (The TOC of the linked project will be inserted at that spot in the parent TOC.)
4. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
5. Click **Enter External Help System**. The Link to Help System dialog opens.
6. Type the name of the child target and main output file (e.g., Another-HelpSystem/AnotherHelpSystem.mcwebhelp).
7. Click **OK**.
8. (Optional) If you want the merged TOC to replace the entry that you selected, do the following:
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

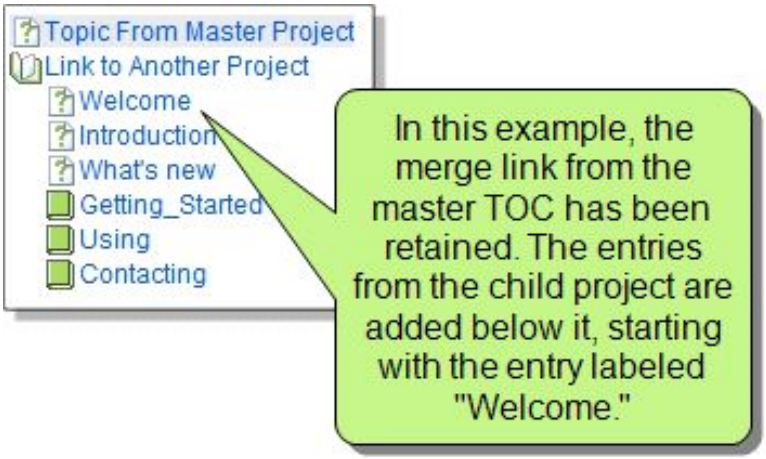
   For example, you might have a TOC in your master project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



9. In the Properties dialog, click **OK**.
10. Build the output for the WebHelp target in your "master" project.
11. In the master Flare project, right-click on the target and select **Open Output Folder**. Within that folder, manually create another folder and name it Subsystems.
12. In the child Flare project, select **Project>Open Output Folder** (if using the ribbon view) or **Build>Open Output Folder** (if using the menu view).
13. Copy the target folder to be merged with the master project.
14. Paste the target folder within the Subsystems folder that you created in the master output folder.

Note: There are occasions when projects cannot be merged because the file name is the same in two or more of the linked projects (e.g., multiple DotNet Help projects all named "MyDotNetProject.mchelp," multiple WebHelp projects all named "MyWebProject.mcwebhelp," or multiple Microsoft HTML Help projects all named "MyHtmlHelp.chm"). The way to solve this dilemma is to make sure each linked project has a different output file name. To do this: (1) open the target, (2) select the **General** tab in the Target Editor, and (3) enter a unique name in the **Output File** field.

Note: If you plan to use Mark of the Web (MOTW) and are merging output from multiple Flare WebHelp projects, make sure you enable MOTW in all of the subprojects to be included in the merge. Otherwise, the output will not load smoothly into the browser window for the user. For more information about MOTW, see Microsoft's MSDN website.

Note: If you have previously merged Flare projects with WebHelp outputs and then decide to generate WebHelp Plus output from the master project instead, you must make sure that the child projects linked to the master contain at least one WebHelp Plus target each. Even if you keep the link from the master project to a WebHelp target in the child, the child project must also have a WebHelp Plus target.

Note: If you merge projects, synonym files will remain separate in each project. For example, if you create synonyms in Project A but not in project B, only the topics from Project A will use the synonyms when users perform searches in the output.

Merging Output from HTML Help Projects (CHM Files)

Use this merging method if you are developing Microsoft HTML Help (a CHM file) and you want to merge your output with another CHM file. This method is useful, for example, if another author is working on the external Microsoft HTML Help project to which you are linking and you only have access to the other CHM file (not the project files).

You can merge your output with another CHM file that you have already brought into your project (perhaps via the external resources feature), or you can select a CHM file located elsewhere, in which case a copy of it is added to your project.

HOW TO MERGE OUTPUT FROM HTML PROJECTS

1. Open the TOC in the Flare project that will serve as the parent project.

   Note: You can also merge projects by linking them from a browse sequence, as well as from a TOC. If you want to do this, simply follow these same steps after opening your browse sequence.

2. Do one of the following:
   - In the TOC Editor, select the TOC entry or book where you want to link the output from the child HTML Help project. (The TOC of the linked CHM file will be inserted at that spot in the parent TOC.)

     OR
   - If the CHM file already exists somewhere in your project, you can open the Content Explorer and drag the CHM file to the location in the "master" TOC where you want to place it. If you use this method, the link will not point to any particular topic in the CHM file. Also, if you use this method, you do not need to complete the rest of the steps below.
3. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
4. On the **General** tab, click **Select HTML Help**. The Link to HTML Help dialog opens.
5. Do one of the following:
   - **If the child CHM file is already in your project** Select **Project Files** and then use the area below to navigate to the file that you want to link to and select it. By using the buttons in the local toolbar, you can view all files in a list, view files in their folder structure, and use other options.

	Shows or hides the folders that the files are stored in.
	Shows or hides the files. If you click this button when the Show Folders button is selected, the area splits into two halves. The folder is shown on the left side, and the files and subfolders within it are shown on the right.

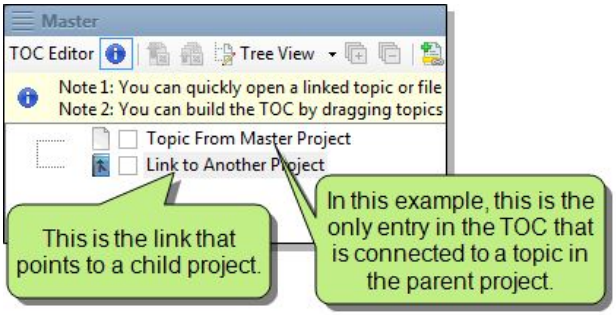
» **If the child CHM file is not yet in your project** Select **Import Existing**. In the dialog that opens, find and double-click the child CHM file. Then from the drop-down—which displays "(root folder)" by default, you can select a specific content folder in your project to place the child CHM file.

6. (Optional) If you want to point to a specific topic in the child CHM file, in the **Topic** field click ... . In the dialog that opens, find and double-click the topic that you want to link to.

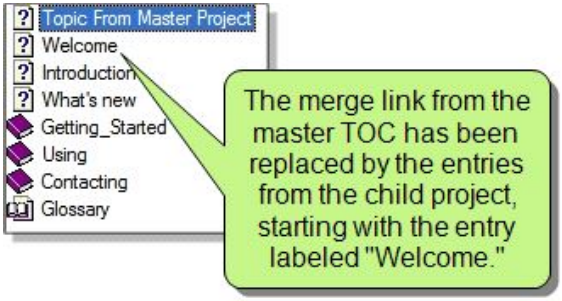
   If you use this option, only the specified topic will be included in the TOC, as opposed to the entire TOC of the child CHM. However, even though just one topic will be included in the merged TOC, the other topics in it will still be included in the output and you can get to them through other means (e.g., clicking hyperlinks).

7. Click **OK** in the Link to HTML Help dialog.
8. If you imported a child CHM file, the Copy to Project dialog opens. You can select **Keep file synchronized (create mapping)**. This creates a link (map) between the original file and the copy being added to your project. Then click **OK** in the Copy to Project dialog.
9. (Optional) If you want the merged TOC to replace the entry that you selected, do the following.
   a. Select the **Advanced** tab.
   b. Click the check box labeled **When merging, replace node with merged TOC**.

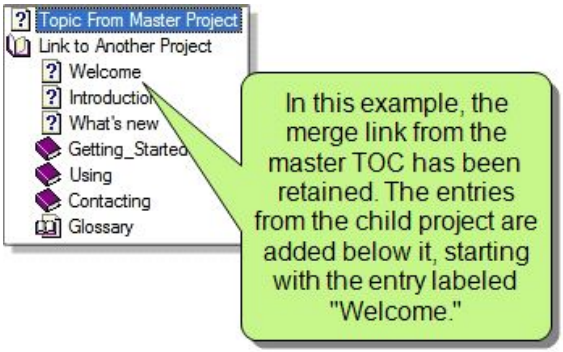
   For example, you might have a TOC in your parent project that looks like this:



If you select the option in this step to replace the node, the merged TOC would look like this in the output:



And if you do not select this option, your merged TOC would look like this:



10. In the Properties dialog, click **OK**.

Note: There are occasions when projects cannot be merged because the file name is the same in two or more of the linked projects (e.g., multiple DotNet Help projects all named "MyDotNetProject.mchelp," multiple WebHelp projects all named "MyWebProject.mcwebhelp," or multiple Microsoft HTML Help projects all named "MyHtmlHelp.chm"). The way to solve this dilemma is to make sure each linked project has a different output file name. To do this: (1) open the target, (2) select the **General** tab in the Target Editor, and (3) enter a unique name in the **Output File** field.

Note: If you are working with HTML Help and import a RoboHelp project that has been merged with other CHM files, the linked CHM files are placed in a special subfolder in the Content Explorer (Resources\CHMSup-port).

Note: You can also link to CHM files from the TOC in DotNet Help and all WebHelp outputs (HTML5, WebHelp, WebHelp AIR, WebHelp Mobile, and WebHelp Plus). However, linking to a *specific topic* within the CHM is not supported in these outputs.

Note: You can merge HTML Help so that the navigation (table of contents, index, search) for each CHM file is displayed, regardless of which CHM file you open (parent or child). This can be done in a couple of ways, depending on whether you have access to all of the Flare projects or to the CHM files only.

Merging Output at Runtime Using HTML5

This is an easy way to merge the output from multiple HTML5 server-based Flare targets into one Help system. These targets can be originated from the same Flare project or from different Flare projects. You simply place the output files in the correct location on the server (i.e., within your master project's AutoMerge folder). Flare then automatically merges the output from all of the targets when users access the Help. From the end user's perspective, the results are seamless, appearing as one large Help system. All of the TOCs, browse sequences, indexes, glossaries, and search capabilities for the projects are merged.

HOW TO MERGE OUTPUT AT RUNTIME USING HTML5

1. **Enable HTML5 Server-Based Output** If you want to take advantage of the advanced server-side features of HTML5 (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable HTML5server-based output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the HTML5 target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling HTML5 search. For more information see the online Help.
2. **Determine Master Output** Decide which of your project outputs will serve as the "master." This is the main output that users will open. All other outputs will be accessed from that output, although it will appear as one large Help system to end users.
3. **(Optional) Specify TOC and/or Browse Sequence Locations for Automerge** By default, the TOCs and browse sequences from the secondary outputs will be appended at the end of the master output's TOC and browse sequence. However, if you want them to be appended somewhere within the master output's TOC or browse sequence, you can specify the exact location. See "Specifying the Automerge Location in a TOC for HTML5" on page 143 and "Specifying the Automerge Location in a Browse Sequence for HTML5" on page 141.
4. **(Optional) Specify Order of Merged Outputs** When you automerge HTML5 outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Specifying the Order of Automerged HTML5 Outputs" on page 145.
5. **Generate Outputs** Build the output for each of the targets to be included in the automerge. You must use HTML5 for all of the targets.
6. **Publish Output to Server** Publish the output for the master project to the web server running Microsoft IIS. The output files for the secondary targets need to be published to the "AutoMerge" subfolder located in the master target's main output folder (e.g., http://www.xyzincorporated.com/help systems/MyTargetName/AutoMerge).

   When readers open the online output, they will see all of the HTML5 outputs merged into one Help system. They can easily navigate and use the index and search across all of the output files.

Note: If you are testing HTML5 server-based output on your local machine, you need view the output at least one time. When you view HTML5 output on your local computer, you need to create a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder. Place a copy of your HTML5 output files in it. This enables you to test the advanced features of HTML5 on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.) When testing the automerge feature on your computer, you need to place the secondary ("child") outputs in the AutoMerge sub-folder at this location (as opposed to the output folder that was generated where your Flare project is located).

Note: If you are testing HTML5 server-based output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

Note: If you want to test HTML5 server-based output on your local computer,the advanced search features of HTML5 are not operable.

Specifying the Automerge Location in a Browse Sequence for HTML5

You can determine where other Flare project outputs are merged relative to your "master" project's browse sequence if you are generating HTML5 server-based output and you are publishing the files to a web server running Microsoft IIS.

By default, the other HTML5 server-based outputs will be merged at the end of your master project's browse sequence. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A BROWSE SEQUENCE

1. Open the browse sequence in the Flare project that will serve as the "master" project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the browse sequence (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new browse sequence item. To do this:
     a. Place your cursor in the browse sequence where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the browse sequence.
3. In the local toolbar of the Browse Sequence Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected browse sequence entry.
   - **After** The automerge will occur immediately after the selected browse sequence entry.
   - **First Child** The automerge will occur at the first location directly after the selected browse sequence book (i.e., before any other entries within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.

- **Last Child** The automerge will occur at the last location after the selected browse sequence book (after the last entry within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.
- **Replace** The automerge will occur at the location of the browse sequence entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master browse sequence so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Automerge Location in a TOC for HTML5

You can determine where other Flare project outputs are merged relative to your "master" project's TOC if you are generating HTML5 server-based output and you are publishing the files to a web server running Microsoft IIS.

By default, the other HTML5 server-based outputs will be merged at the end of your master project's TOC. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A TOC

1. Open the TOC in the Flare project that will serve as the parent project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the TOC (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new TOC item. To do this:
     a. Place your cursor in the TOC where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the TOC.
3. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected TOC entry.
   - **After** The automerge will occur immediately after the selected TOC entry.
   - **First Child** The automerge will occur at the first location directly after the selected TOC book (i.e., before any other entries within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.
   - **Last Child** The automerge will occur at the last location after the selected TOC book (after the last entry within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.

» **Replace** The automerge will occur at the location of the TOC entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master TOC so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Order of Automerged HTML5 Outputs

When you automerge HTML5 outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Merging Output at Runtime Using HTML5" on page 139.

EXAMPLE

Let's say you have a master project with a target named "Main" and three smaller projects with targets named "First Child," "Second Child," and "Third Child," respectively. If you place the three smaller project outputs in the AutoMerge subfolder where "Main" is published, their TOCs will be appended to the master project's TOC in the following order:

1. First Child
2. Second Child
3. Third Child

But what if you want "Third Child" to be appended first? You can use the steps in this topic to position it above the other targets. Therefore, the result would be:

1. Third Child
2. First Child
3. Second Child

Now let's say that you add two more targets (called "Final Child" and "Another Child") to the AutoMerge folder. If you do not adjust your custom order, Flare will automerge these outputs in alphabetical order at the end of your custom order. Therefore, the final result would be:

1. Third Child
2. First Child
3. Second Child
4. Another Child
5. Final Child

HOW TO SPECIFY THE ORDER OF AUTOMERGED OUTPUTS

1. Create an XML file and name it SortOrder.xml. You can do this by opening an editor such as Notepad. When you save the file, type the xml file extension at the end of the name.
2. Enter the following code into the blank file.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>ProjectA</Item>

<Item>ProjectB</Item>

<Item>ProjectC</Item>

</SortOrder>
```

3. Replace "ProjectA," "ProjectB," and "ProjectC" with the names of your own child targets. You can add or remove line items as necessary.

   If you were to use the example at the top of this topic, you would enter the following.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>Third Child</Item>

<Item>First Child</Item>

<Item>Second Child</Item>

</SortOrder>
```

4. Save the file.
5. Copy and paste the SortOrder.xml file into the AutoMerge directory of your published HTML5 output. This file sits beside the child output folders that you are using for the automerge.

Merging Output at Runtime Using WebHelp Plus

This is an easy way to merge the output from multiple WebHelp Plus Flare targets into one Help system. These targets can be originated from the same Flare project or from different Flare projects. You simply place the output files in the correct location on the server (i.e., within your master project's AutoMerge folder). Flare then automatically merges the output from all of the targets when users access the Help. From the end user's perspective, the results are seamless, appearing as one large Help system. All of the TOCs, browse sequences, indexes, glossaries, and search capabilities for the projects are merged.

HOW TO MERGE OUTPUT AT RUNTIME USING WEBHELP PLUS

1. **Enable WebHelp Plus Output** If you want to take advantage of the advanced server-side features of WebHelp Plus (i.e., automatic runtime project merging, faster search, searching of non-XHTML files), you must enable WebHelp Plus output. This includes performing the following tasks: (1) installing Microsoft Internet Information Services (IIS) and ASP.NET, (2) setting up the WebHelp Plus target and generating/publishing, (3) configuring IIS on the production server, (4) starting Microsoft Indexing Service or Microsoft Windows Search (depending on the operating system), and (5) enabling WebHelp Plus search. For more information see the online Help.
2. **Determine Master Output** Decide which of your project outputs will serve as the "master." This is the main output that users will open. All other outputs will be accessed from that output, although it will appear as one large Help system to end users.
3. **(Optional) Specify TOC and/or Browse Sequence Locations for Automerge** By default, the TOCs and browse sequences from the secondary outputs will be appended at the end of the master output's TOC and browse sequence. However, if you want them to be appended somewhere within the master output's TOC or browse sequence, you can specify the exact location. See "Specifying the Automerge Location in a TOC for WebHelp Plus" on page 151 and "Specifying the Automerge Location in a Browse Sequence for WebHelp Plus" on page 149.
4. **(Optional) Specify Order of Merged Outputs** When you automerge WebHelp Plus outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Specifying the Order of Automerged WebHelp Plus Outputs" on page 153.
5. **Generate Outputs** Build the output for each of the targets to be included in the automerge. You must use WebHelp Plus for all of the targets.
6. **Publish Output to Server** Publish the output for the master project to the web server running Microsoft IIS. The output files for the secondary targets need to be published to the "AutoMerge" subfolder located in the master target's main output folder (e.g., http://www.xyzincorporated.com/help systems/MyTargetName/AutoMerge).

   When readers open the online output, they will see all of the WebHelp Plus outputs merged into one Help system. They can easily navigate and use the index and search across all of the output files.

Note: If you are testing WebHelp Plus output on your local machine, you need view the output at least one time. When you view WebHelp Plus output on your local computer, Flare creates a special folder called "MCPreview" within your "C:\Inetpub\wwwroot" folder, and it places a copy of your WebHelp Plus output files in it. This enables you to test the advanced features of WebHelp Plus on your local machine. (For Windows Vista users testing locally, the physical location is C:\MadCap\[Name of Project]\[Name of Main Entry File].htm.) When testing the automerge feature on your computer, you need to place the secondary ("child") outputs in the AutoMerge subfolder at this location (as opposed to the output folder that was generated where your Flare project is located).

Note: If you are testing WebHelp Plus output on your local machine, you may need to wait a few minutes after viewing the output for the Indexing Service to fully scan your files. Otherwise, you may not immediately see the effects of the scan (e.g., searches of non-XHTML files, incorporation of merged output files) in the output. If you avoid performing other tasks during this period, the scanning of the files will be completed more quickly.

Note: If you want to test WebHelp Plus on your local computer,the advanced search features of WebHelp Plus are not operable.

Specifying the Automerge Location in a Browse Sequence for WebHelp Plus

You can determine where other Flare project outputs are merged relative to your "master" project's browse sequence if you are generating WebHelp Plus output and you are publishing the files to a web server running Microsoft IIS.

By default, the other WebHelp Plus outputs will be merged at the end of your master project's browse sequence. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A BROWSE SEQUENCE

1. Open the browse sequence in the Flare project that will serve as the "master" project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the browse sequence (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new browse sequence item. To do this:
     a. Place your cursor in the browse sequence where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the browse sequence.
3. In the local toolbar of the Browse Sequence Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected browse sequence entry.
   - **After** The automerge will occur immediately after the selected browse sequence entry.
   - **First Child** The automerge will occur at the first location directly after the selected browse sequence book (i.e., before any other entries within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.

- **Last Child** The automerge will occur at the last location after the selected browse sequence book (after the last entry within the book). If you use this option on a simple browse sequence entry instead of a book, the entry will automatically become a book once the outputs are automerged.
- **Replace** The automerge will occur at the location of the browse sequence entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master browse sequence so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Automerge Location in a TOC for WebHelp Plus

You can determine where other Flare project outputs are merged relative to your "master" project's TOC if you are generating WebHelp Plus output and you are publishing the files to a web server running Microsoft IIS.

By default, the other WebHelp Plus outputs will be merged at the end of your master project's TOC. However, you can use the following steps to select one of the available options to override this placement.

HOW TO SPECIFY THE AUTOMERGE LOCATION IN A TOC

1. Open the TOC in the Flare project that will serve as the parent project.
2. Do one of the following:
   - If you want to merge the other outputs in relation to one of the existing entries in the TOC (e.g., before it, after it), select that entry (whether it is an individual item or a book).

     OR
   - If you want to merge the other outputs at the location of an entry that is not linked to any other file, and you want to provide a label to indicate the location of the merge, create a new TOC item. To do this:
     a. Place your cursor in the TOC where you want to add the new item.
     b. Click .
     c. Press **F2**.
     d. Replace the default text with new text.
     e. Press **Enter**.
     f. If necessary, use the arrow buttons in the local toolbar to position the new entry in the TOC.
3. In the local toolbar of the TOC Editor, click . The Properties dialog opens.
4. Select the **Advanced** tab.
5. In the **Server-based Automerge field**, select one of the following:
   - **Before** The automerge will occur immediately before the selected TOC entry.
   - **After** The automerge will occur immediately after the selected TOC entry.
   - **First Child** The automerge will occur at the first location directly after the selected TOC book (i.e., before any other entries within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.
   - **Last Child** The automerge will occur at the last location after the selected TOC book (after the last entry within the book). If you use this option on a simple TOC entry instead of a book, the entry will automatically become a book once the outputs are automerged.

» **Replace** The automerge will occur at the location of the TOC entry where you have specified this option. It would replace any links that might otherwise be applied to that entry. You might use this option, for example, if you want to create a new entry in the master TOC so that you can add a label at the point where the automerge occurs.

6. In the Properties dialog, click **OK**.
7. Click to save your work.

Specifying the Order of Automerged WebHelp Plus Outputs

When you automerge WebHelp Plus outputs, the secondary outputs are merged to the master project's table of contents (TOC) and/or browse sequence in alphabetical order. However, you can override this default configuration and merge the secondary outputs in any order that you like. See "Merging Output at Runtime Using WebHelp Plus" on page 147.

EXAMPLE

Let's say you have a master project with a target named "Main" and three smaller projects with targets named "First Child," "Second Child," and "Third Child," respectively. If you place the three smaller project outputs in the AutoMerge subfolder where "Main" is published, their TOCs will be appended to the master project's TOC in the following order:

1. First Child
2. Second Child
3. Third Child

But what if you want "Third Child" to be appended first? You can use the steps in this topic to position it above the other targets. Therefore, the result would be:

1. Third Child
2. First Child
3. Second Child

Now let's say that you add two more targets (called "Final Child" and "Another Child") to the AutoMerge folder. If you do not adjust your custom order, Flare will automerge these outputs in alphabetical order at the end of your custom order. Therefore, the final result would be:

1. Third Child
2. First Child
3. Second Child
4. Another Child
5. Final Child

HOW TO SPECIFY THE ORDER OF AUTOMERGED OUTPUTS

1. Create an XML file and name it SortOrder.xml. You can do this by opening an editor such as Notepad. When you save the file, type the xml file extension at the end of the name.
2. Enter the following code into the blank file.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>ProjectA</Item>

<Item>ProjectB</Item>

<Item>ProjectC</Item>

</SortOrder>
```

3. Replace "ProjectA," "ProjectB," and "ProjectC" with the names of your own child targets. You can add or remove line items as necessary.

   If you were to use the example at the top of this topic, you would enter the following.

```
<?xml version="1.0" encoding="utf-8"?>

<SortOrder>

<Item>Third Child</Item>

<Item>First Child</Item>

<Item>Second Child</Item>

</SortOrder>
```

4. Save the file.
5. Copy and paste the SortOrder.xml file into the AutoMerge directory of your published WebHelp Plus output. This file sits beside the child output folders that you are using for the automerge.

Displaying Merged Navigation in HTML Help Child Outputs

You can merge HTML Help so that the navigation (table of contents, index, search) for each CHM file is displayed, regardless of which CHM file you open (parent or child). This can be done in a couple of ways, depending on whether you have access to all of the Flare projects or to the CHM files only.

HOW TO DISPLAY MERGED NAVIGATION IN HTML HELP—ACCESS TO ALL FLARE PROJECTS

1. Open the target to be used as the parent CHM.
2. On the **Advanced** tab of the Target Editor, select **Display merged navigation in HTML Help**.
3. Click to save your work.
4. Follow the steps for merging based on HTML Help targets. See "Merging Flare Projects Using Targets" on page 117.
5. Generate the parent HTML Help target.

   The parent CHM file and all child CHM files are located in the same output folder, and when you open any of those CHM files, you can see the navigation for any of the other outputs.

HOW TO DISPLAY MERGED NAVIGATION IN HTML HELP—ACCESS TO CHM FILES ONLY

Use this method if you *do not* have access to all of the Flare projects, but you *do* know the names of the other CHM files that will be created. This method is more of a manual setup and is useful if you are part of a team where different authors have access to different Flare projects that need to be merged. This method also lets you add or remove child CHM files without needing to recompile the parent project.

1. The author for each project (parent *and* child) places a simple TXT file named *merge.txt* in the same folder where the Flare project file (.flprj) is located. This TXT file can be created with Notepad and should simply list the names of all CHM files to be included in the merged output (one on each line), with the name of the parent CHM appearing first.

   E X A M P L E

   MyParent.chm

   MyFirstChild.chm

   MySecondChild.chm

   MyThirdChild.chm

2. The author for each child project generates his output and provides the author of the parent project with the CHM file(s).

3. The author for the parent project brings the child CHM files into the parent project and follows the steps for merging based on CHM files. See "Merging Output from HTML Help Projects (CHM Files)" on page 135.
4. The author for the parent project generates the main HTML Help target.

   The parent CHM file and all child CHM files are located in the same output folder, and when you open any of those CHM files, you can see the navigation for any of the other outputs.

CHAPTER 3

Exporting Projects

You can export projects, thus creating copies of them.

This chapter discusses the following:

Adding an Export Project File 158
Exporting Projects 159
Exporting Projects Using the Command Line 184

Adding an Export Project File

The following steps show you how to add an export project file, which is used to export projects.

HOW TO ADD AN EXPORT PROJECT FILE

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Project** ribbon. In the **Content** section select **New>Add Export Project File**.
   - **Menu** Select **Project>Add Export Project File**.

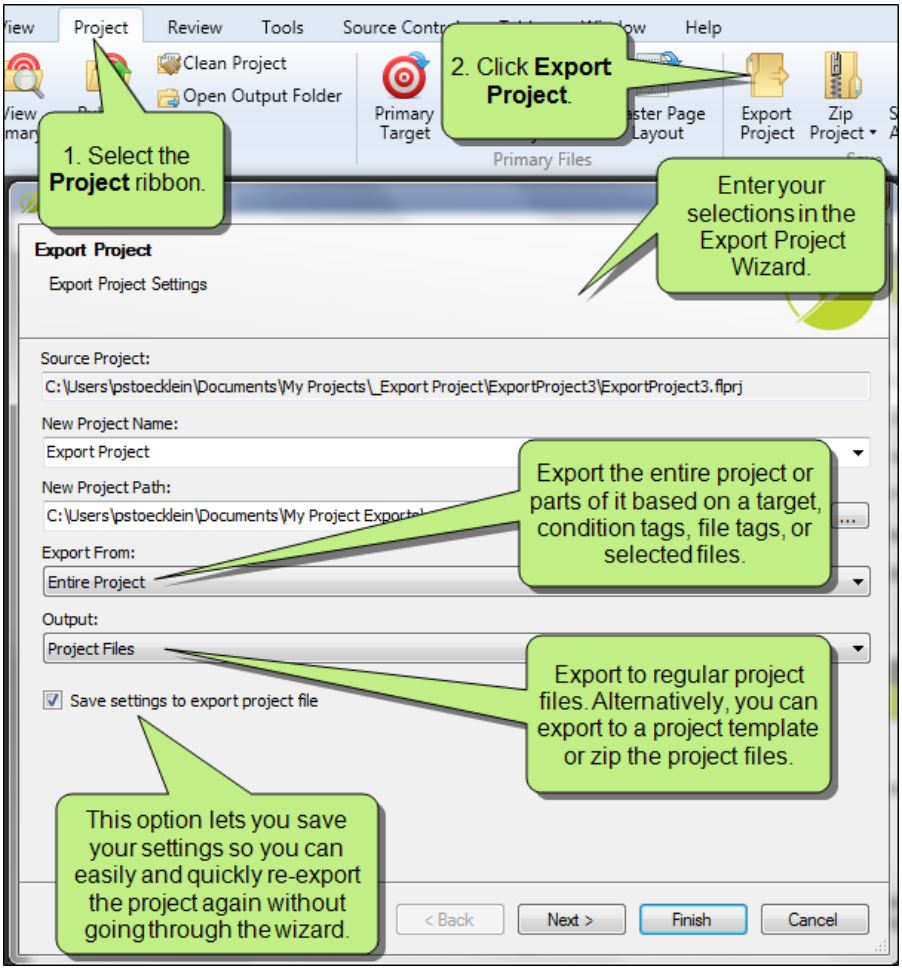
   The Add File dialog opens.
2. In the **File Type** field at the top, make sure **Export Project File** is selected.
3. In the **Source** area select one of the following:
   - **New from template** This lets you choose either the factory template file or one of your own customized template files as a starting point. The new file will take on all of the settings contained in the template. If you want to use the factory template provided by Flare, expand the **Factory Templates** folder and click on a template file. If you want to use your own customized template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.

     Note: In some dialogs and wizards you can click the **Manage Templates** button if you want to open the Template Manager. This lets you manage any of your template files (e.g., add new templates, enter descriptions for templates). For more information see the online Help.
   - **New from existing** This lets you choose an existing file of the same type as a starting point for your new file. As with template files, your new file will take on all of the settings contained in the file you select. To use this option, click the browse button ..., use the Open File dialog to find a file, and double-click it.
4. In the **File Name** field, type a new name for the export file.
5. Click **Add**. The export file is added to the Exports folder in the Project Organizer. The Export Project File Editor opens to the right.

After you add the new export file, you can export the project or part of it. See "Exporting Projects" on the following page.

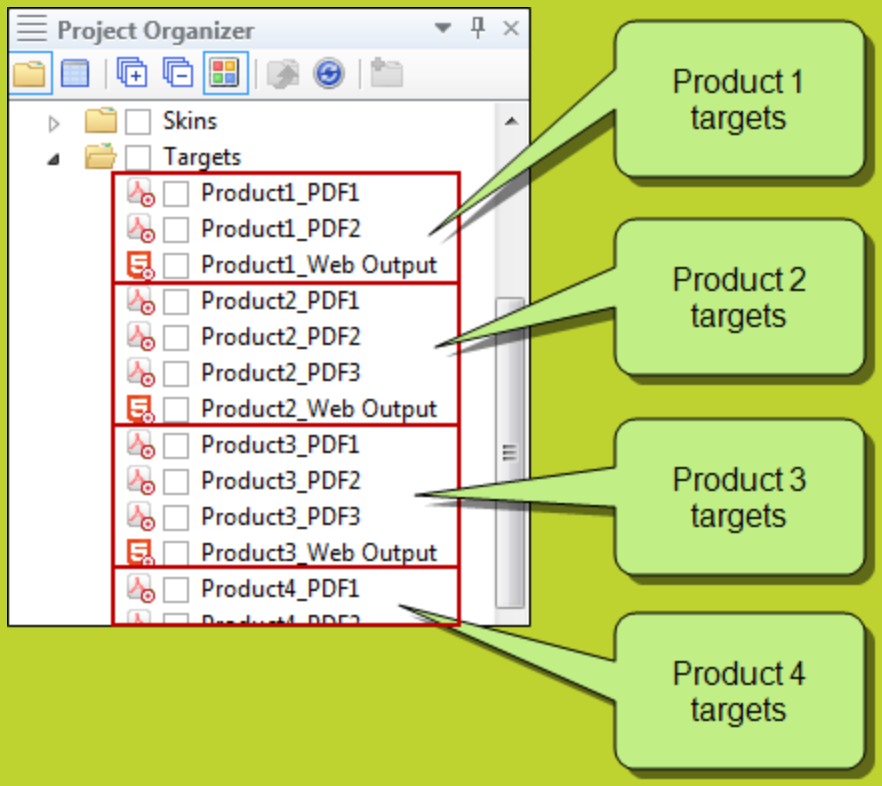
Exporting Projects

You can export an entire Flare project, or parts of one, to another location. One reason you might want to use this feature is to quickly and easily archive projects, especially if you have an extremely large Flare project and need to archive only parts of it. Another use for this feature is translation. If you only need a portion of a master project to be translated, you don't want to send the translator all of the files, but rather a smaller version of the project containing only the files requiring translation.

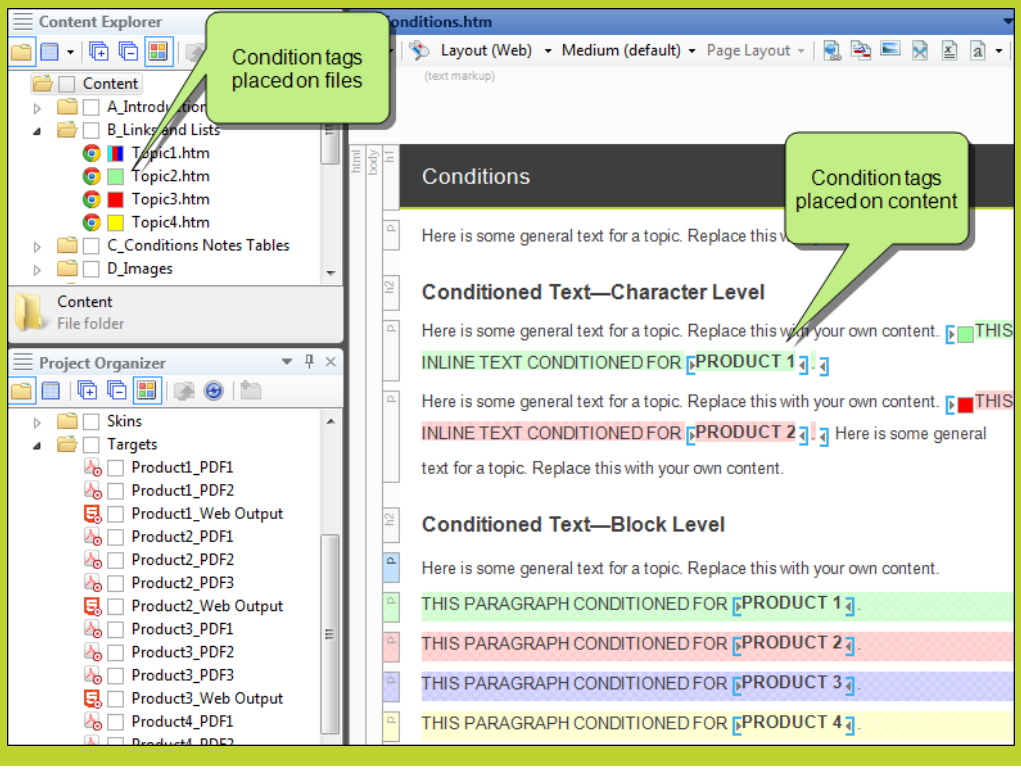


EXAMPLE—ARCHIVE

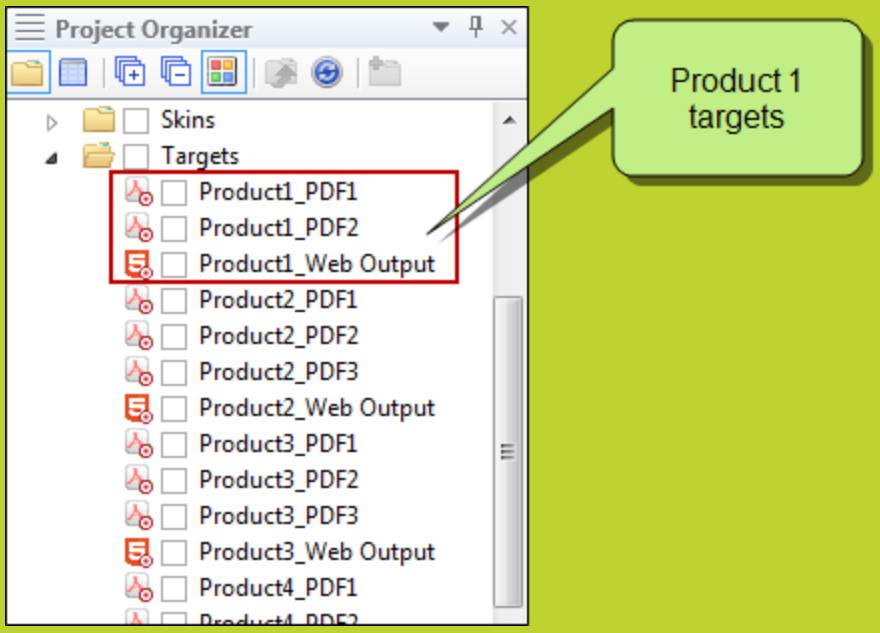
Let's say you have a large Flare project, from which you create single-sourced output for four products. Furthermore, let's say that the documentation for each of those products consists of several targets in the Flare project (e.g., some online, some print-based outputs).



In order to keep the output separate and organized throughout the Flare project, you use condition tags, both on content and on the files themselves (e.g., topics, images, stylesheet).



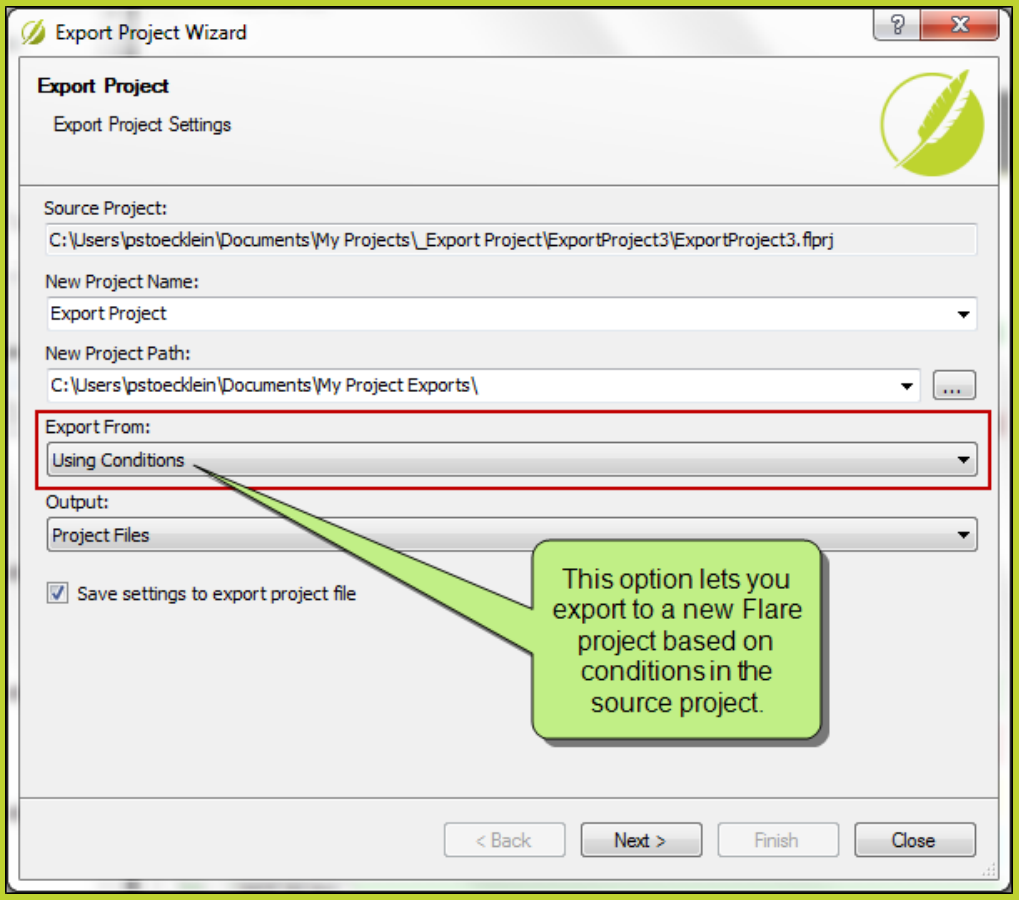
Your manager says that each time you finish documentation for a particular product, you must save just those files in a repository. So let's say one day you finish the documentation for Product 1, which you've designed to produce output from three targets (one output for the Web, and two outputs for PDFs).



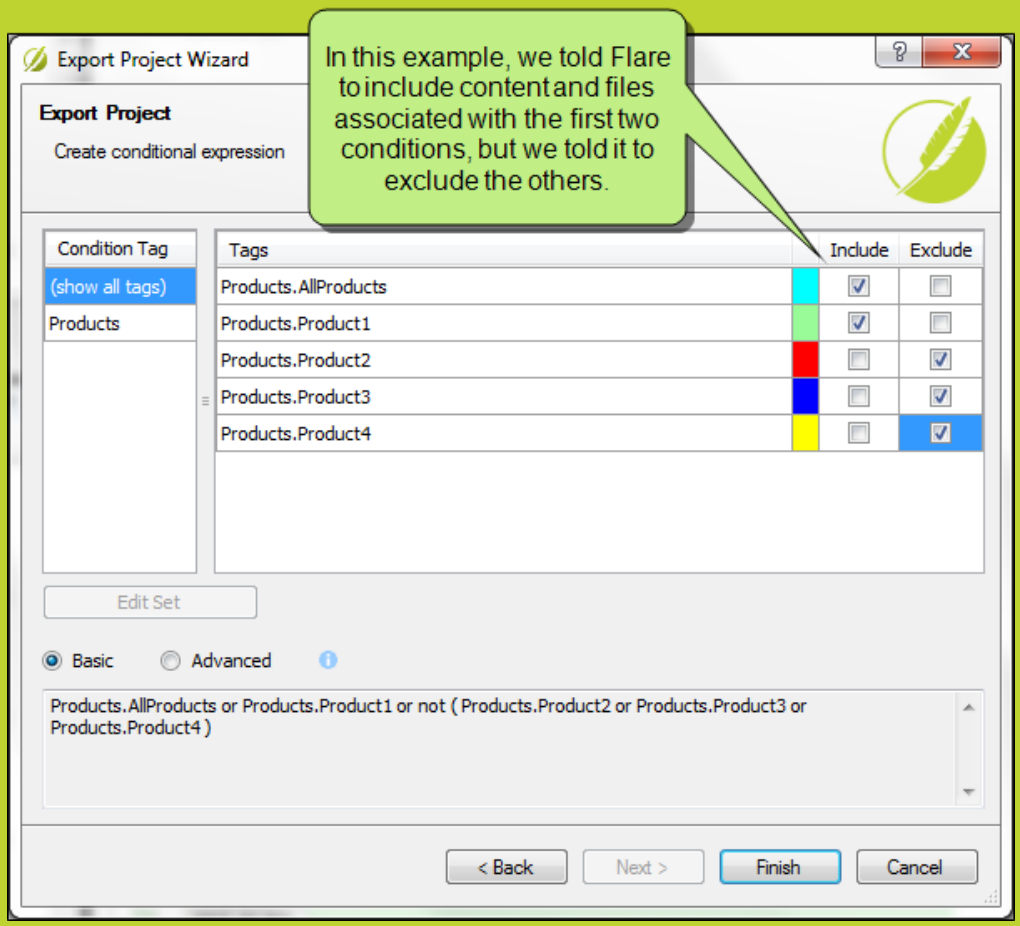
Your large Flare project holds the finished documentation for Product 1, but it also holds the documentation for Products 2, 3, and 4, which are in different stages of progress. You want to archive only the documentation that is specific to Product 1, so you decide to export the relevant files from the large Flare project to a smaller one.

But how do you do this? One option is to export to a smaller project based on condition tags, which you have used to organize all of the content and files in the project.

So from the **Project** ribbon you click **Export Project**. On the first page of the wizard, you click in the **Export From** drop-down and select **Using Conditions**.



You can then tell Flare which conditions to include and which to exclude. In this case, you tell Flare to include all of the condition tags that have to do with Product 1 and its targets, and you exclude all of the condition tags that have to do exclusively with the other products and their targets.

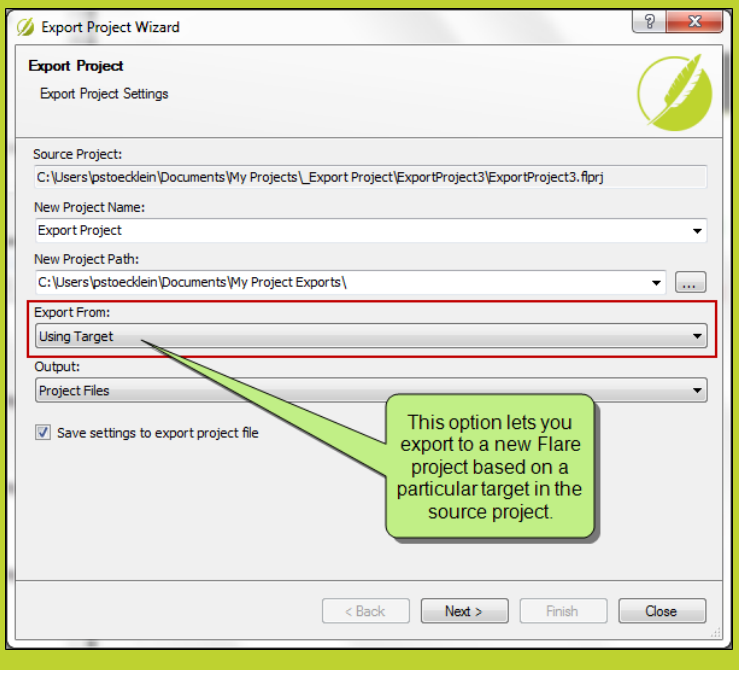


After clicking **Finish**, the relevant files and content are exported to a new, smaller Flare project.

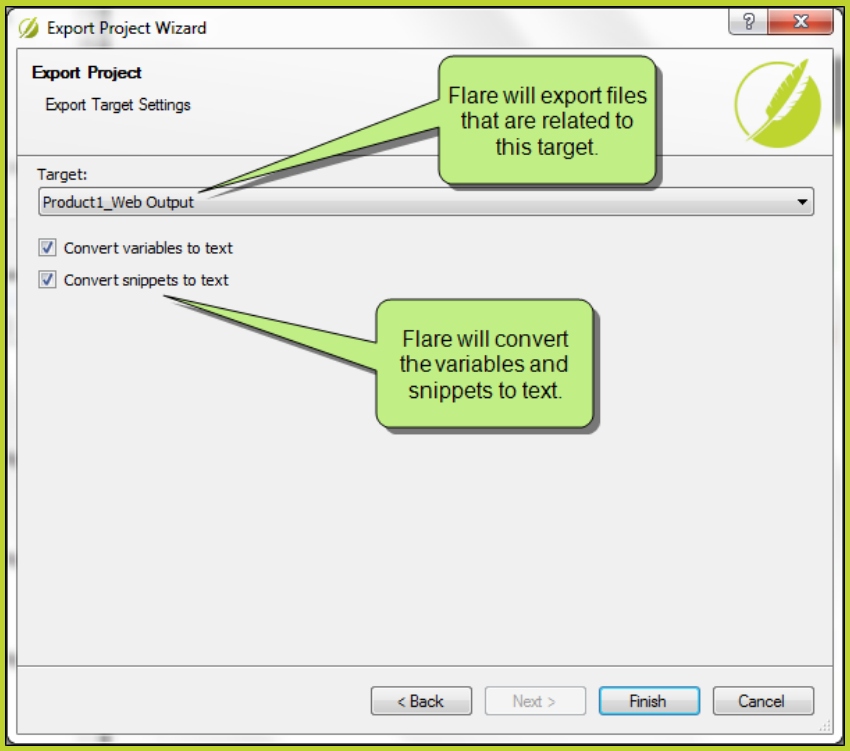
EXAMPLE—TRANSLATION

Let's say you have a Flare project with seven targets you need to translate the content associated one of those targets from English to French. You could send your entire Flare project to the translator, but that would mean the translator would be getting files associated with all seven targets, not just the one requiring translation. So you decide to export only the portion of the Flare project that needs to be translated.

First, from the **Project** ribbon you click **Export Project**. On the first page of the wizard, you click in the **Export From** drop-down and select **Using Target**.



On the next page of the wizard, you select the target whose files you want to export. In this case, let's say the target in question is named "Product1_Web Output." In addition, you tell Flare to convert your variables and snippets to text so that they become part of the topics, rather than separate files.



After clicking **Finish**, the relevant files and content are exported to a new, smaller Flare project. Only the files and content necessary to produce the Product1_Web Output target are included in the export. Therefore, the translator receives only the files requiring translation.

You can export a project using the Export Project Wizard to guide you. Alternatively, you can add an export file to your project and use the Export Project File Editor.

HOW TO EXPORT A PROJECT USING THE EXPORT PROJECT WIZARD

1. Open a project.
2. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Project** ribbon. In the **Save** section select **Export Project**.
   - **Menu** Select **File>Export Project**.

   The Export Project Wizard opens.
3. In the **New Project Name** field, enter a name for the exported project.
4. (Optional) The New Project Path field is automatically populated with the default location (Documents\My Project Exports). If you want to export the project to a different location, click ... and select the folder you want.

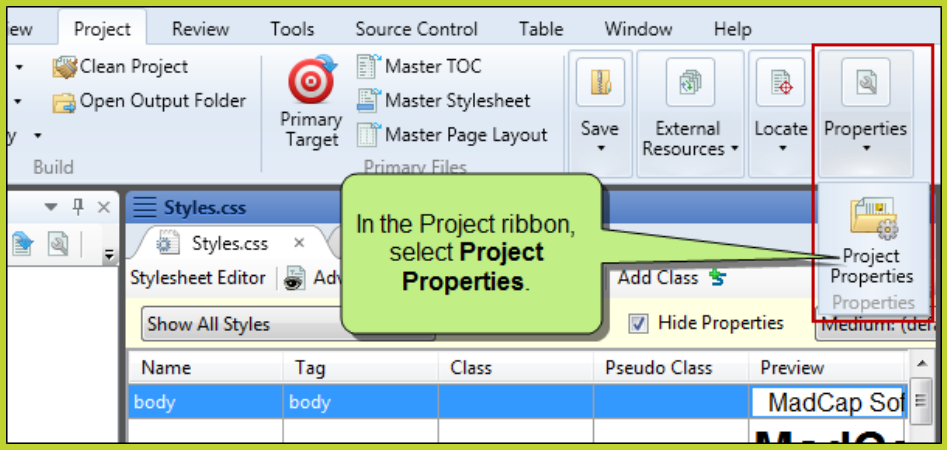
   Note: In some operating systems, the folder is called "My Documents" instead of "Documents."

   Note: If you select Project Template below in step 6, this area is grayed out because the project files will automatically be placed in your templates folder.
5. In the **Export From** drop-down, select one of the following:
   - **Entire Project** This option makes a copy of the entire project, including all folders, subfolders, and files.
   - **Using Target** This option uses the same workflow as that used for generating a target. When you select a particular target, the same files and content that would be included in generated output are included in the exported project.
   - **Using Conditions** This option exports only files and content affected by condition tags that you tell Flare to include or exclude.
   - **Using File Tags** This option exports only files affected by file tags that you tell Flare to include or exclude. This method can be especially useful for translation purposes, exporting only files that are marked with a certain file tag status (e.g., Ready for Translation).
   - **Select Files** This option exports only specific files that you select. You can choose any files stored in the source project's Content Explorer and Project Organizer.

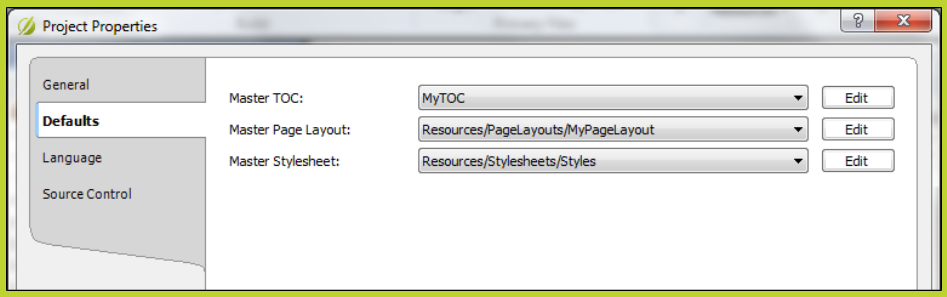
   If you have a master TOC, page layout, or stylesheet set in the Project Properties dialog and include those files in an export, the settings in the Project Properties dialog will be preserved. If you do not include those files in an export, default files are added to the exported project to allow it to work, and the settings in the Project Properties dialog are set to default.

EXAMPLE

You can open the Project Properties dialog from the Project ribbon.

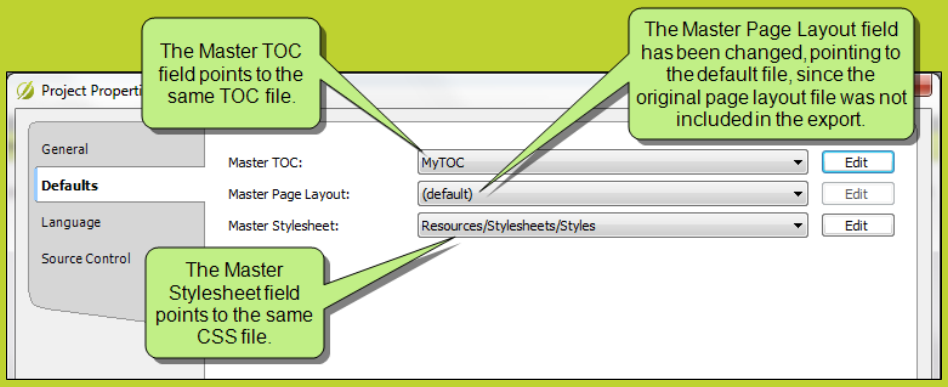


Let's say the Defaults tab looks like this:



Notice that the fields point to specific files.

Suppose you export this project and include the TOC and styles files in the export, but you do not include the page layout file. In that case, the Project Properties dialog will look like this in the exported project:



Note: If Flare detects any missing files (e.g., stylesheet, master page, target) that are necessary in order to properly open the exported project, default files are added. For example, if no target file is added directly as a result of your selection, an HTML5 target is added by default.

6. In the **Output** drop-down, select one of the following:
   - **Project Files** This option simply exports the project files to a location that you select. The default location is Documents\My Project Exports.
   - **Project Template** This option exports the project files to your templates folder (e.g., Documents\My Templates\Projects). By being placed in this location, the project files become available as a template selection when you create a new project.
   - **Zip Files** This option packages the project files into a single zip file with an .flprjzip extension and places it in a location that you select. The default location is Documents\My Project Exports.
7. (Optional) Select **Save settings to export project file** if you want to retain your settings for future re-exports. The export project file is stored in the Exports folder in the Project Organizer. When you open this export file, the Export Project File Editor opens. This editor lets you change any of your settings (if necessary) and export the project again (see the set of steps below for exporting a project using the Export Project File Editor).
8. Click **Next**.

9. If you are exporting based on a target, click in the **Target** drop-down field and select the appropriate target from the source project.
10. (Optional) You can select one or both of the following:
    - **Convert variables to text** Select this option if you want all relevant variables to be converted to text.
    - **Convert snippets to text** Select this option if you want all relevant snippets to be converted to text.
11. Do the following, depending on the type of export you selected in step 5:

    ENTIRE PROJECT

    Skip to step 12.

    USING TARGET

    Skip to step 12.

    USING CONDITIONS

    a. Click **Next**.

    b. Use either the Basic or Advanced section to tell Flare which condition tags to include and which to exclude from the export.

    By default when you open this page, the **Basic** option is selected. You can click **Advanced** to switch to that mode if you are experienced at writing expressions by hand. While the Basic section is easier for most people to use, it is also more limited in the type of expressions it can create.

    With the Basic method, all of your work is done on the top half of the user interface, by just clicking check boxes. On the other hand, with the Advanced method, most of your work is done on the bottom portion of the user interface.

    For details and examples on using each of these methods, see the online Help.

    Note: In order for this method to work, make sure you have already applied the appropriate condition tags to files and content in the project.

    Note: If you are an experienced user and want to use the Advanced mode, you might find it helpful to begin with the Basic mode to create the initial expression and then switch to Advanced when you are ready.

> **Note:** The Finish button is not enabled until you create at least one condition expression on this page.

USING FILE TAGS

a. Click **Next**.

b. In the **Tag Type** field, you can select a specific file tag set or choose to display all file tags to the right.

c. Use either the **Basic** or **Advanced** section to tell Flare which file tags to include (e.g., Ready for Translation) and which to exclude from the export.

   By default when you open this page, the **Basic** option is selected. You can click **Advanced** to switch to that mode if you are experienced at writing expressions by hand. While the Basic section is easier for most people to use, it is also more limited in the type of expressions it can create.

   With the Basic method, all of your work is done on the top half of the user interface, by just clicking check boxes. On the other hand, with the Advanced method, most of your work is done on the bottom portion of the user interface.

   The process for creating file tag expressions is essentially the same as that for condition tag expressions.

> **Note:** In order for this method to work, make sure you have already associated the appropriate file tags with files in the project.

> **Note:** If you are an experienced user and want to use the Advanced mode, you might find it helpful to begin with the Basic mode to create the initial expression and then switch to Advanced when you are ready.

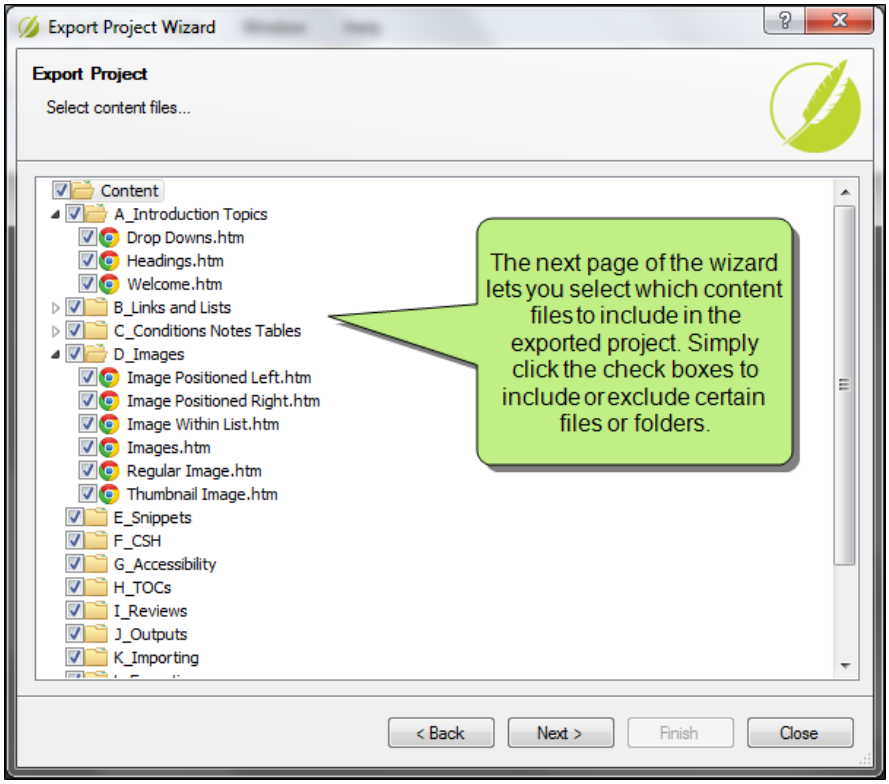
> **Note:** The Finish button is not enabled until you create at least one file tag expression on this page.

SELECT FILES

a. Click **Next**.

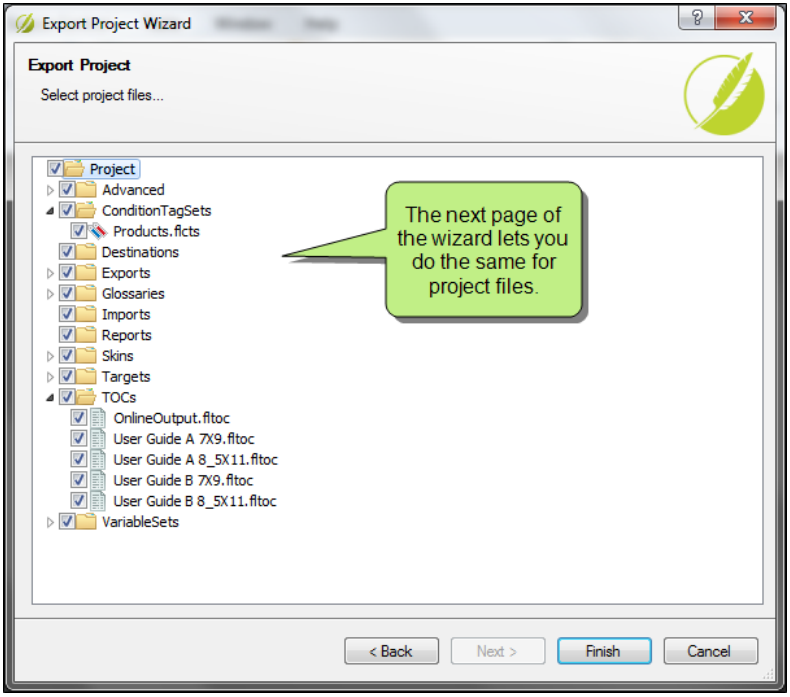
b. Make sure a check mark is next to each content folder or file that you want to include in the exported project.

By default, all of the boxes are checked, but you can click in any of the boxes to remove the check marks if necessary. You can expand folders and subfolders to see files within them.



c. Click **Next**.

d. Make sure a check mark is next to each project folder or file that you want to include in the exported project. By default, all of the boxes are checked, but you can click in any of the boxes to remove the check marks if necessary. You can expand folders and subfolders to see files within them.



12. Complete one of the following sets of steps, depending on the type of output method you selected in step 6:

PROJECT FILES

» Click **Finish**. The project is exported, and Windows opens to the location, showing you the files.

PROJECT TEMPLATE

a. Click **Next**.

b. On the next page of the template, select the template folder where you want to export the files. You can click **Manage Templates** if you need to add or change template folders.

c. In the **Template Name** field, enter a new name for the template.

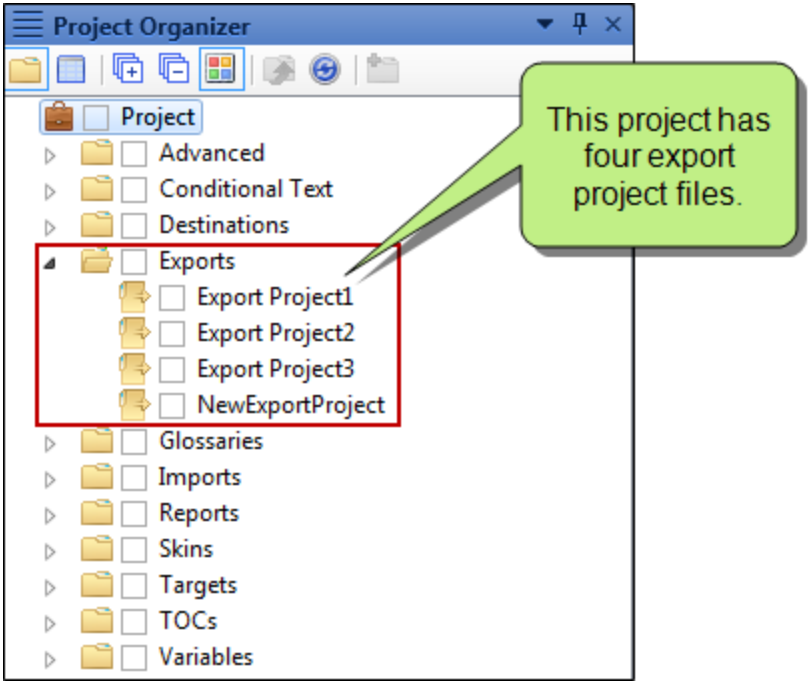
d. Click **Finish**. The project is exported, and Windows opens to the location, showing you the files.

ZIP FILES

» Click **Finish**. The project is exported, and Windows opens to the location, showing you the zipped file.

HOW TO EXPORT A PROJECT USING THE EXPORT PROJECT FILE EDITOR

1. Add an export file to your project via the previous set of steps using the Export Project Wizard. You can also do this manually (see "Adding an Export Project File" on page 158).
2. Make sure the Project Organizer is open.
3. Double-click the **Exports** subfolder.
4. Double-click the appropriate export file.

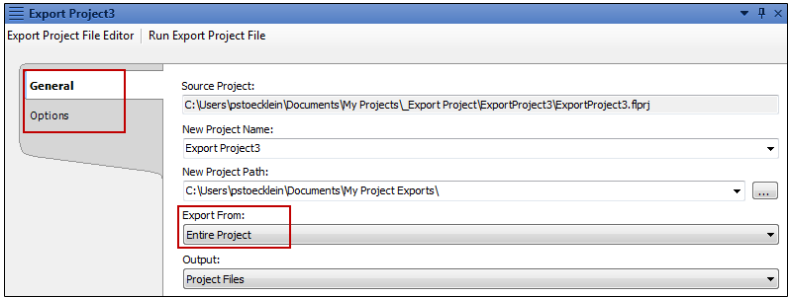


The Export Project File Editor opens. At minimum a General tab is shown, with basic export settings.

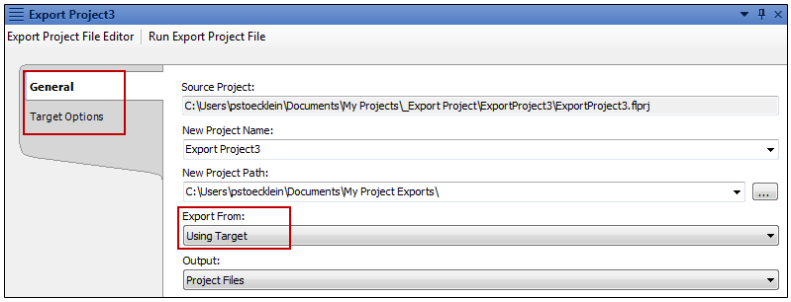
TABS IN THE EXPORT PROJECT FILE EDITOR

Depending on your selection in the **Export From** or **Output** field, the editor changes to show additional tabs when needed.

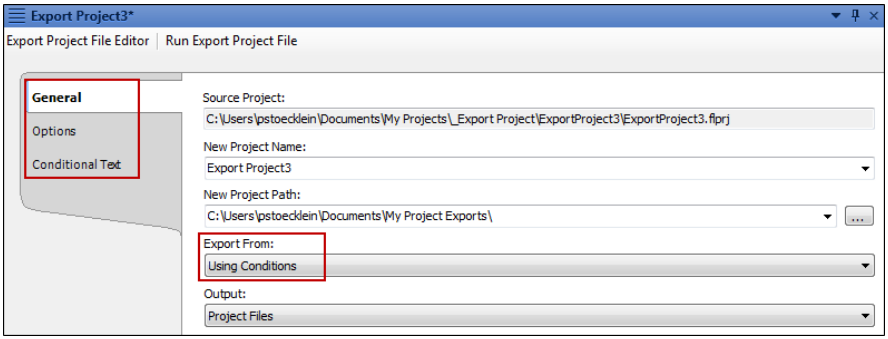
- If you choose **Entire Project**, the Start tab is shown, as well as an Options tab for converting variables and snippets to text.



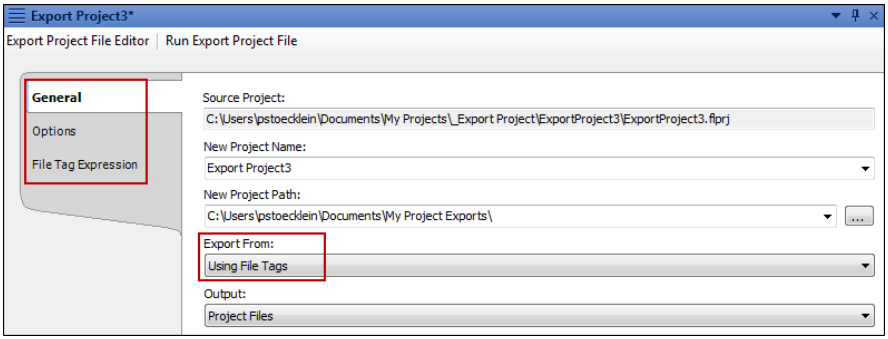
- If you choose **Using Target**, the Start tab is shown, as well as a Target Options tab so that you can adjust those settings, as well as convert variables and snippets to text.



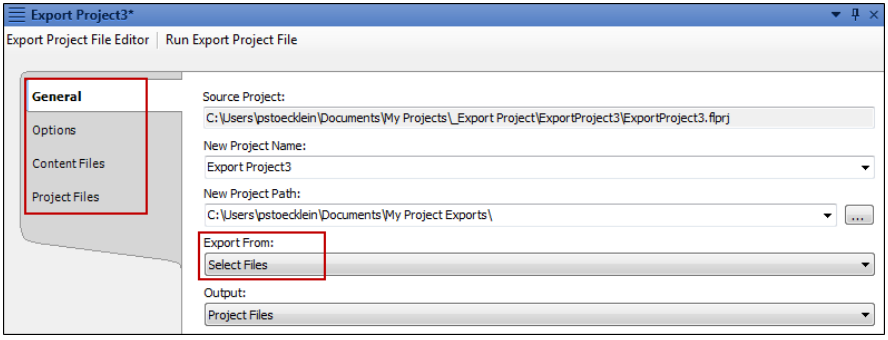
» If you choose **Using Conditions**, the Start tab is shown, as well as an Options tab for converting variables and snippets to text and a Conditional Text tab so that you can include and exclude conditions.



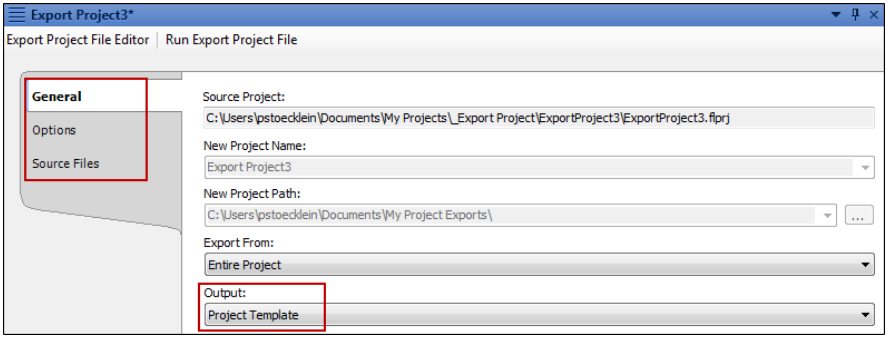
» If you choose **Using File Tags**, the Start tab is shown, as well as an Options tab for converting variables and snippets to text and a File Tag Expression tab is shown so that you can include and exclude file tags.



» If you choose **Select Files**, the Start tab is shown, as well as an Options tab for converting variables and snippets to text. In addition, a Content Files tab is shown so that you can adjust content file selections, and a Project Files tab is shown so that you can adjust project file selections.



» If you choose **Project Template**, a Template Path tab is also shown so that you can select a template folder to store the exported project. You can also use this tab to manage your templates.



5. (Optional) Adjust the settings on any of the relevant tabs:

START TAB

- **Source Project** This field displays the path and name of the current project. You cannot edit this field.
- **New Project Name** Enter a name for the exported project.
- **New Project Path** The New Project Path field is automatically populated with the default location (Documents\My Project Exports). If you want to export the project to a different location, click [...] and select the folder you want.

> **Note:** In some operating systems, the folder is called "My Documents" instead of "Documents."

> **Note:** If you select Project Template in the Export From field, this area is grayed out because the project files will automatically be placed in your templates folder.

- **Export From** Select one of the following:
  - **Entire Project** This option makes a copy of the entire project, including all folders, subfolders, and files.
  - **Using Target** This option uses the same workflow as that used for generating a target. When you select a particular target, the same files and content that would be included in generated output are included in the exported project.
  - **Using Conditions** This option exports only files and content affected by condition tags that you tell Flare to include or exclude.
  - **Using File Tags** This option exports only files affected by file tags that you tell Flare to include or exclude. This method can be especially useful for translation purposes, exporting only files that are marked with a certain file tag status (e.g., Ready for Translation).
  - **Select Files** This option exports only specific files that you select. You can choose any files stored in the source project's Content Explorer and Project Organizer.

> **Note:** If Flare detects any missing files (e.g., stylesheet, master page, target) that are necessary in order to properly open the exported project, default files are added. For example, if no target file is added directly as a result of your selection, an HTML5 target is added by default.

- **Output** Select one of the following:
  - **Project Files** This option simply exports the project files to a location that you select. The default location is Documents\My Project Exports.
  - **Project Template** This option exports the project files to your templates folder (e.g., Documents\My Templates\Projects). By being placed in this location, the project files become available as a template selection when you create a new project.
  - **Zip Files** This option packages the project files into a single zip file with an .flprjzip extension and places it in a location that you select. The default location is Documents\My Project Exports.

OPTIONS TAB

- **Convert variables to text** Select this option if you want all relevant variables to be converted to text.
- **Convert snippets to text** Select this option if you want all relevant snippets to be converted to text.

TARGET OPTIONS TAB

- **Target** Select the appropriate target from the source project.
- **Convert variables to text** Select this option if you want all relevant variables to be converted to text.
- **Convert snippets to text** Select this option if you want all relevant snippets to be converted to text.

CONDITIONAL TEXT TAB

Use either the Basic or Advanced section to tell Flare which condition tags to include and which to exclude from the export.

By default when you open this page, the **Basic** option is selected. You can click **Advanced** to switch to that mode if you are experienced at writing expressions by hand.While the Basic section is easier for most people to use, it is also more limited in the type of expressions it can create.

With the Basic method, all of your work is done on the top half of the user interface, by just clicking check boxes. On the other hand, with the Advanced method, most of your work is done on the bottom portion of the user interface.

For details and examples on using each of these methods, see the online Help.

Note: In order for this method to work, make sure you have already applied the appropriate condition tags to files and content in the project.

Note: If you are an experienced user and want to use the Advanced mode, you might find it helpful to begin with the Basic mode to create the initial expression and then switch to Advanced when you are ready.

FILE TAG EXPRESSION TAB

In the **Tag Type** field, you can select a specific file tag set or choose to display all file tags to the right.

Use either the **Basic** or **Advanced** section to tell Flare which file tags to include (e.g., Ready for Translation) and which to exclude from the export.

By default when you open this page, the **Basic** option is selected. You can click **Advanced** to switch to that mode if you are experienced at writing expressions by hand. While the Basic section is easier for most people to use, it is also more limited in the type of expressions it can create.

With the Basic method, all of your work is done on the top half of the user interface, by just clicking check boxes. On the other hand, with the Advanced method, most of your work is done on the bottom portion of the user interface.

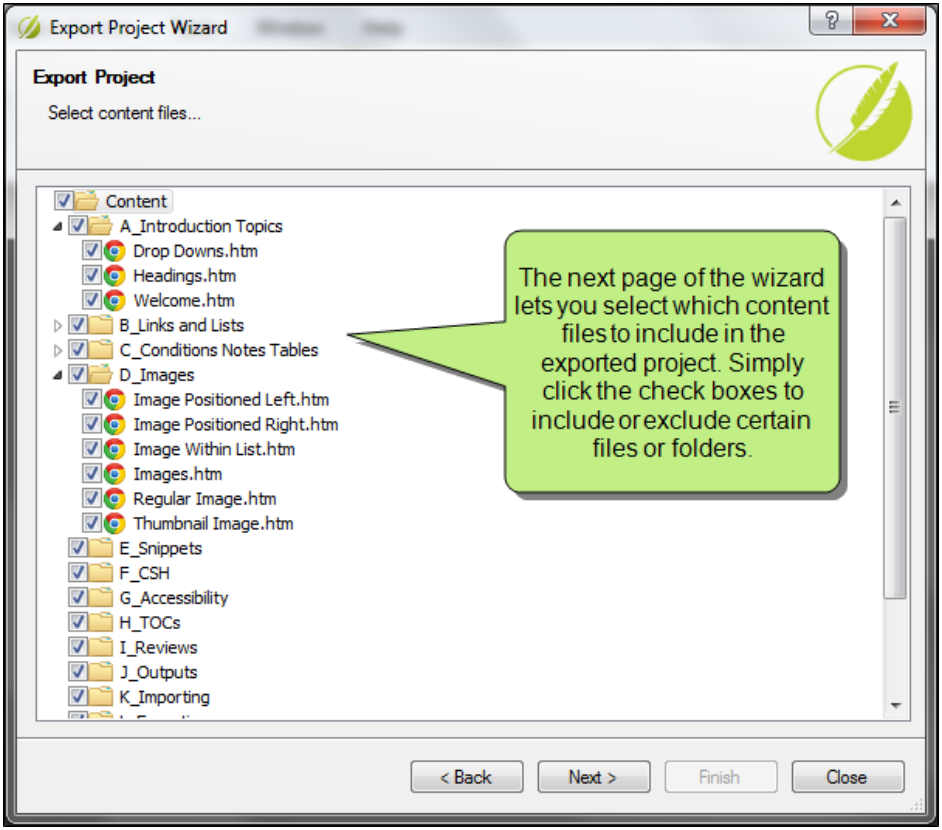
The process for creating file tag expressions is essentially the same as that for condition tag expressions.

Note: In order for this method to work, make sure you have already associated the appropriate file tags with files in the project.

Note: If you are an experienced user and want to use the Advanced mode, you might find it helpful to begin with the Basic mode to create the initial expression and then switch to Advanced when you are ready.

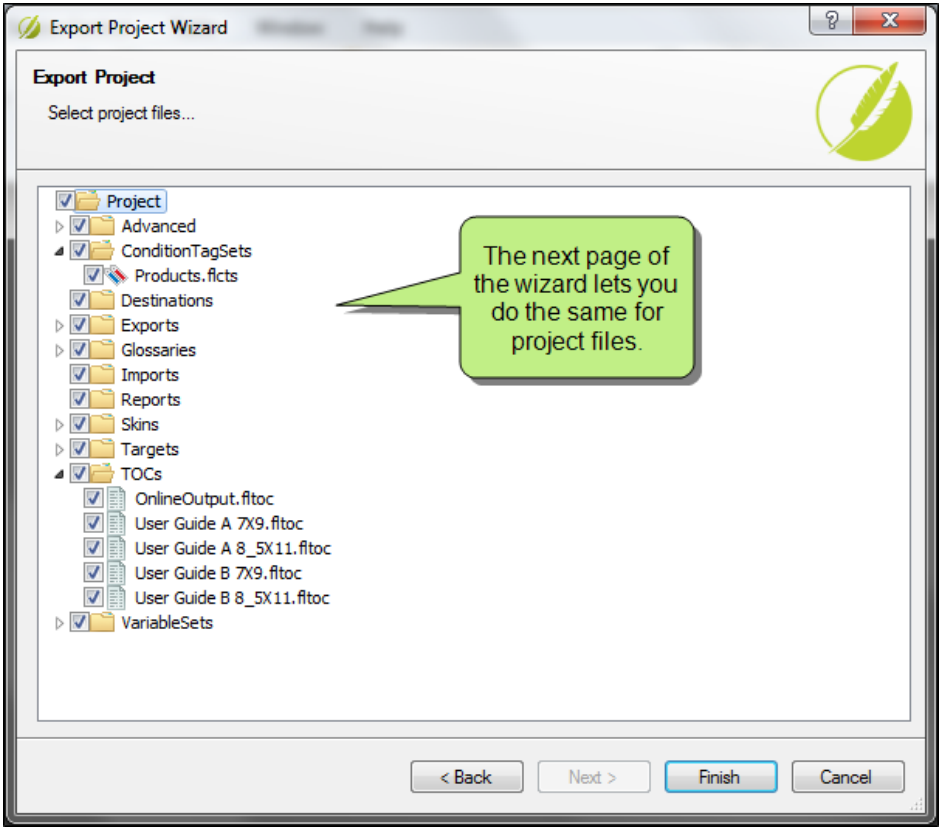
CONTENT FILES TAB

Make sure a check mark is next to each content folder or file that you want to include in the exported project. By default, all of the boxes are checked, but you can click in any of the boxes to remove the check marks if necessary. You can expand folders and subfolders to see files within them.



PROJECT FILES TAB

Make sure a check mark is next to each project folder or file that you want to include in the exported project. By default, all of the boxes are checked, but you can click in any of the boxes to remove the check marks if necessary. You can expand folders and subfolders to see files within them.



TEMPLATE PATH TAB

- **Template Folder** Select the template folder where you want to export the files.
- **Manage Templates** Select this option if you want to add or change template folders.
- **Template Name** Enter a new name for the template.

6. Click to save your work.
7. At the top of the Export Project File Editor, click **Run Export Project File**. The project is exported, and Windows opens to the location, showing you the files.

Note: As an alternative to exporting projects in the Flare user interface, you can use madbuild to export projects from the command line. Using this method, you do not need to have Flare open, and you can schedule exports for specific days and times. See "Exporting Projects Using the Command Line" on the next page.

Note: There is also a less robust, but quicker feature for zipping projects. See "Zipping Projects" on page 193.

Exporting Projects Using the Command Line

As an alternative to exporting projects in the Flare user interface, you can use madbuild to export projects from the command line. Using this method, you do not need to have Flare open, and you can schedule exports for specific days and times.

HOW TO USE THE COMMAND LINE TO EXPORT A PROJECT

1. Add an export project file in the Project Organizer. It will be placed in an Exports folder. See "Adding an Export Project File" on page 158.
2. Complete the fields in the Export Project File Editor. You need to do this so that Flare knows where to export the project, which parts of it to export, and so on. These fields need to be completed only once. After that, you can export the project anytime you want and these same rules will be used. For more information on each field and option in the Export Project File Editor, see "Exporting Projects" on page 159.
3. Do one of the following, depending on whether you want to create a batch file for later use or to export immediately using the command line.

   » **To Create a Batch File (Recommended)** Open Notepad. For example, in Windows 7, you can open it by clicking the **Start** button and selecting **All Programs>Accessories>Notepad**.

   OR

   » **To Generate Immediately From the Command Line** Open your command prompt. For example, in Windows 7, you can open it by clicking the **Start** button and then selecting **All Programs>Accessories>Command Prompt**.

4. Type the path to the Flare.app folder where you installed Flare, and press **Enter**.

   E X A M P L E

   If you installed Flare at C:\Program Files **(this is just an example; your files might be installed in a different folder, such as Program Files x86)**, you might type the following and press **Enter** on your keyboard:

   ```
   cd\Program Files\MadCap Software\MadCap Flare V10\Flare.app
   ```

5. Type the following and press **Enter** on your keyboard (entering the path to your project, and the export file within it, in place of the section in angle brackets):

   ```
   madbuild -export <project path> -settings <export project file name>
   ```

   Make sure you add the .flexp file extension at the end of the export project file name.

Note: If there are spaces anywhere in your path or project name, you need to use quotation marks around it.

Tip: If you are using the command prompt (as opposed to a Notepad file) and you have an operating system that supports dragging, you can drag a file from the Windows location to the Command Prompt window. This will add the path of the file within quotation marks for you.

EXAMPLE—NO SPACES

Let's say you have a Flare project called "FictionSoftPro" that you have stored in a folder of the same name at the root level of the C: drive. And within that project you've created an export file named "MyExportFile." In that case, you can type this:

```
madbuild -export c:\FictionSoftPro\FictionSoftPro.flprj -settings MyExportFile.flexp
```

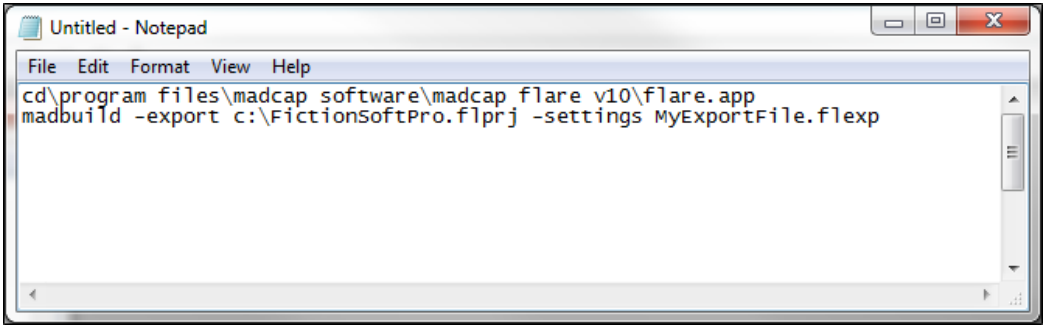
EXAMPLE—SPACES

Let's say you have a project called "FictionSoftPro" that you have stored in a folder named "My Projects" (with a space) at the root level of the C: drive. And within that project you've created an export file named "My Export File" (with spaces). In that case, you can type this:

```
madbuild -export "c:\My Projects\FictionSoftPro\FictionSoftPro.flprj" -settings "My Export File.flexp"
```

If you are using the command prompt window, the project is exported immediately.

If you are creating a batch file in Notepad, it might look something like this:



6. If you are creating a batch file, save the Notepad file to any location you like on your computer. When you do this, type `.bat` as the extension at the end of the file name (e.g., MyBatchFile.bat). At any time you like, you can export the project simply by double-clicking the batch file. You can also use a tool to schedule the batch file to run.

   For example, complete the following steps if using the Windows 7 Task Scheduler utility.

   a. In Windows, click the **Start** button.
   b. Select **All Programs>Accessories>System Tools>Task Scheduler**.
   c. Click **Action>Create Basic Task**.
   d. In the Create Basic Task Wizard, give the task a name and click **Next**.
   e. Using the next couple of pages of the wizard, choose when you wan the batch to run (e.g., daily, starting at 2 a.m.). Click **Next** until you get to the Action page.
   f. Click **Start a program**, and click **Next**.
   g. Click **Browse**. Then find and double-click the batch file you created.
   h. Click **Next**.
   i. Click **Finish**.

ERROR CODES

If you export a project using the command line, the following error codes may be seen if problems occur during an export process:

- **3001** The entered export project file does not exist.
- **3002** There was an error parsing the entered export project file.
- **3003** Export failed.

- **3004** Export destination name not set in the export file
- **3005** Export destination path not set in the export file
- **3006** Template name not set in the export file
- **3007** Template path not set in the export file
- **3008** Target path not set in export file (when exporting using target)
- **3009** Target path not found (when exporting using target)
- **3010** Conditional expression not set (when exporting using conditional expression)
- **3011** File Tag expression not set (when exporting using file tag expression)
- **3012** No files selected (when exporting using manual selection)

CHAPTER 4

Other Project Tasks

In addition to creating and merging projects, there are several other project-related tasks that you might perform.

This chapter discusses the following:

Opening a Project 190
Closing Projects 191
Deleting Projects 192
Zipping Projects 193

Opening a Project

There are two ways to open an existing project. You can use interface options or drag files from Windows Explorer.

HOW TO OPEN A PROJECT USING INTERFACE OPTIONS

Do one of the following, depending on the part of the user interface you are using:

- **Start Page** On the left side of the Start Page, click **Open**. You can also click a recently opened project. If you do not see the Start Page, select **View>Start Page**.
- **Ribbon** Select **File>Open**. The Open dialog displays. The Flare project file extension (.flprj) appears in the File Type field. In the Open dialog navigate to the project you want, select it, and click **Open**.

  You can use the Options dialog to switch between ribbons and the classic tool strip layout. For more information see the online Help.

  Keep in mind that the smaller the application window becomes, the more the options in a ribbon shrink. Therefore, you might only see a small icon instead of text, or you might see only a section name displayed with a down arrow to access the options in it. You can hover over small icons to see tooltips that describe them. You can also enlarge the application window or click one of the section drop-downs in the ribbon to locate a hidden feature.
- **Menu** Select **File>Open**. The Open dialog displays. The Flare project file extension (.flprj) appears in the File Type field. In the Open dialog navigate to the project you want, select it, and click **Open**.
- **Standard Toolbar** Click . The Open dialog displays. The Flare project file extension (.flprj) appears in the File Type field. In the Open dialog navigate to the project you want, select it, and click **Open**.

  To open the Standard toolbar from the menu view, you can select **View>Toolbars>Standard**.
- **Keyboard Shortcut** Press **CTRL+O**. The Open dialog displays. The Flare project file extension (.flprj) appears in the File Type field. In the Open dialog navigate to the project you want, select it, and click **Open**.

HOW TO OPEN A PROJECT BY DRAGGING A FILE FROM WINDOWS EXPLORER

1. Open Windows and navigate to a folder containing a project file.
2. Launch Flare.
3. Drag the project file from Windows to the application window and drop it on the title bar in Flare.

Closing Projects

When you are finished working on a project, you can either close the project or close Flare altogether. If you close just the project, Flare remains open, which lets you start another project if you want.

HOW TO CLOSE A PROJECT

Do one of the following, depending on the part of the user interface you are using:

- **Ribbon** Select **File>Close>Close Project**.
- **Menu** Select **File>Close Project**.

Deleting Projects

To delete a project completely, you must use Windows Explorer. This operation cannot be done from Flare.

You can also remove projects from displaying as recent projects in Flare. This does not delete the project, but simply removes it from those shortcut locations.

HOW TO DELETE A PROJECT THROUGH WINDOWS

1. In Windows Explorer, browse to the location where you have stored the project (e.g., C:\Documents and settings\Projects\My Project).
2. Delete the folder holding the project files.

HOW TO REMOVE A PROJECT FROM RECENT PROJECTS IN FLARE

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select **File>Manage Recent Projects**.
   - **Menu** Select **File>Recent Projects>Manage Recent Projects**.

   The Recent Projects dialog opens.
2. Highlight the project and click **Remove**.
3. Click **OK**.

Zipping Projects

If you want to send your project to another location or person, you can package (zip) the project into a single file that is much smaller than the entire project in its normal state. The zipped project can then be easily unzipped by anyone who also uses Flare, retuning the project to its normal state so that it can be edited. Zipping a project is also a great way to create and store a backup of your project.

There are two methods you can use—(1) zipping only or (2) zipping and emailing. The first method lets you zip the project and place the compressed file where you want on your computer. The second method lets you zip the project and immediately attach it to an email so that you can send it to someone. This topic deals with the first method. Also see "Ribbon Select the Project ribbon. In the Save section select Zip Project>Zip and Email Project. " below.

HOW TO ZIP A PROJECT

1. Open a project.
2. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Project** ribbon. In the **Save** section select **Zip Project**.
   - **Menu** Select **File>Zip>Zip Project**.

   The Create Package dialog opens.
3. (Optional) You can click in the **Package File** field and rename the zipped file at the end of the path. By default, the zipped file is named after your project.
4. (Optional) You can click the **Browse** button and navigate to another folder to store the zipped file. By default, Flare specifies that the zipped file should be placed in your "Documents" folder, but you can use the Browse button to change that.
5. Click **Create**. Flare compresses the project and places it into a single file with an .flprjzip extension.

HOW TO ZIP AND EMAIL A PROJECT

1. Open a project.
2. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Project** ribbon. In the **Save** section select **Zip Project>Zip and Email Project**.
   - **Menu** Select **File>Zip>Zip and Email Project**.

   Flare compresses the project and places it into a single file with an.flprjzip extension. The file is attached to an email message, ready for you to send to others.

   When individuals receive the email, the can unzip the file by double-clicking it (if they also have Flare installed).

Note: Alternatively, you can use a more robust export feature, with additional options, to zip a project (or parts of one). See "Exporting Projects" on page 159.

APPENDIX

PDF Guides

The following PDF guides are available for download from the online Help.

Accessibility Guide
Analyzer Guide
Autonumbers Guide
Condition Tags Guide
Context-sensitive Help Guide
DotNet Help Guide
Eclipse Help Guide
Getting Started Guide
Global Project Linking Guide
HTML Help Guide
HTML5 Guide
Images Guide
Importing Guide
Index Guide
Key Features Guide
Language Support Guide
Movies Guide
Navigation Links Guide
Print-based Output Guide
Project Creation Guide
Pulse Guide
QR Codes Guide
Reports Guide
Reviews & Contributions Guide
Search Guide
SharePoint Guide
Shortcuts Guide
Skins Guide

Snippets Guide

Source Control Guide

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

Transition From FrameMaker Guide

Variables Guide

WebHelp Outputs Guide

What's New Guide