

MADCAP FLARE 2024 r2

Global Project Linking

Copyright © 2024 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
1660 17th Street, Suite 201
Denver, Colorado 80202
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1

Introduction	5
--------------------	---

CHAPTER 2

Adding Project Import Files	7
How to Add a Project Import File Manually	8

CHAPTER 3

Importing From Other Projects	9
How to Import Files Using the Import Flare Project Wizard ..	9

CHAPTER 4

Reimporting Files	17
How to Reimport Files From Another Project	18

CHAPTER 5

Rules for Global Project Linking	25
--	----

APPENDIX

PDFs	29
Tutorials	29
Cheat Sheets	30

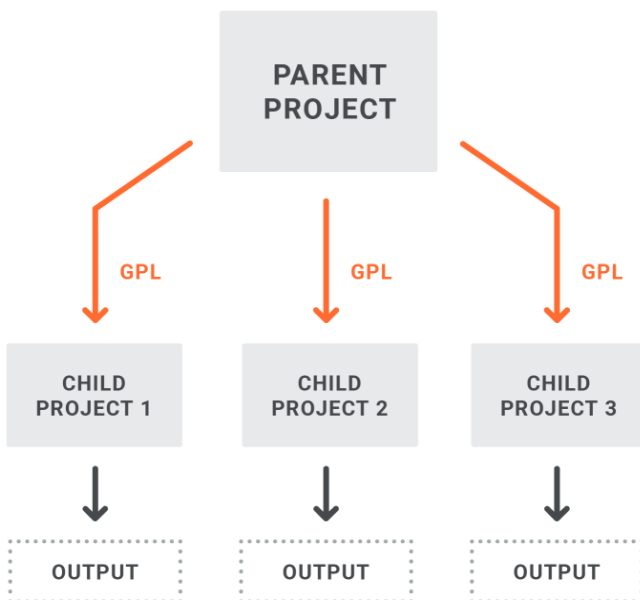
User Guides31

CHAPTER 1

Introduction

By using Global Project Linking, you can import content and project files contained in another Flare project, thus allowing you to maintain the information in one location but reuse it in any other project. With this feature, you can include or exclude particular types of files (e.g., topics, snippets, stylesheets, glossaries, targets), specific individual files, or files that have certain condition tags applied. Simply use the include/exclude methods that work best for you.

This is different than a simple import process, because in this case, the imported files remain linked to the source project. This lets you make future updates to those files in just one place—in the parent project. When you perform ongoing imports using your previous settings, Flare recognizes changes to the source files. Therefore, the new files can be brought over, replacing the outdated ones.



☆ **EXAMPLE** You are working on three different Flare projects. Within those projects, you might have 35 topics and 50 images that are identical in the three projects. In addition, you might use the same stylesheet in each project.

Rather than maintaining three different sets of identical files, you can store one set of those files and import them into the individual projects when needed. Here are a couple of options: (1) One option is that you could consider one of your three Flare projects as the "global parent" for those shared files. (2) Another option is that you could create a new Flare project (perhaps naming it "global"); this project could have no other purpose than to serve as a repository for the shared files across your projects. In other words, you would not necessarily generate any output from this parent project, but simply use it as a place to hold your shared information.





When you want to use any of the shared topic, image, or stylesheet files from the global project, you would import them into the child project. This creates a link between the imported files and those in the global project. Therefore, when you edit those files in the future, you would do so from the global project and then reimport the changes (either manually or automatically) to the other child projects.

Adding Project Import Files

The Import Flare Project Wizard guides you through the steps for importing Flare files from another project. When you use this wizard, an import file is created automatically and placed in the Imports folder in the Project Organizer.

However, as an alternative to using the Import Flare Project Wizard, you can add an import project file manually. After you create this file, it will be stored in the Imports folder in the Project Organizer. If you want to manually reimport files from the project in the future, you can open this file (with an `.flimpfl` extension). Your settings are saved in the file, and you can simply initiate a reimport.

I How to Add a Project Import File Manually


1. Do one of the following, depending on the part of the user interface you are using:
 - **Ribbon** Select **Project > New > Import File > Flare Project Import File**.
 - **Right-Click** In the Project Organizer, right-click on the **Imports** folder and from the context menu select **Add Flare Project Import File**.
2. In the **File Type** field at the top, make sure **Flare Import File** is selected.
3. In the **Source** area, choose to create the new file based on a template or an existing file.
 - **New From Template** Choose either a factory template file or one of your own custom 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 custom template file, expand the appropriate folder and click on a file. For more information about templates, see the online Help.
 - **New From Existing** 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 , use the Open File dialog to find a file, and double-click it.
4. (Optional) If you want to place the file into a subfolder previously created in the Content Explorer or Project Organizer, in the **Folder** field click  and select the subfolder. Otherwise, keep the default location.
5. In the **File Name** field, type a new name for the import file.
6. (Optional) If you want to apply condition tags to the file, expand the **Attributes** section at the bottom of the dialog. Next to the **Condition Tags** field, click  and select the conditions you want to apply. Click **OK**.
7. (Optional) If you want to apply file tags, expand the **Attributes** section at the bottom of the dialog. Next to the **File Tags** field, click  and select the file tags you want to apply. Click **OK**.
8. Click **Add**. The import file is added to the Project Organizer. The Project Import Editor opens to the right.

Importing From Other Projects

You can import files from another Flare project using Global Project Linking.

I How to Import Files Using the Import Flare Project Wizard

The Import Flare Project Wizard guides you through the steps for importing Flare files from another project. When you use this wizard, an import file is created automatically and placed in the Imports folder in the Project Organizer.

1. Select **Project > Import > Flare Project**. The Import Flare Project Wizard opens.
2. Next to the **Project file** field click .
3. In the dialog find and double-click the Flare project file (.flprj extension) from which you will be importing files.
4. (Optional) If you want future changes to the files in the source project to be reimported automatically, click the **Auto-reimport before "Generate Output"** check box. If you select this "Easy Sync" option, the changes to the source files will be reimported automatically when you generate output. If you do not select this option, you can reimport future changes from the other project manually.

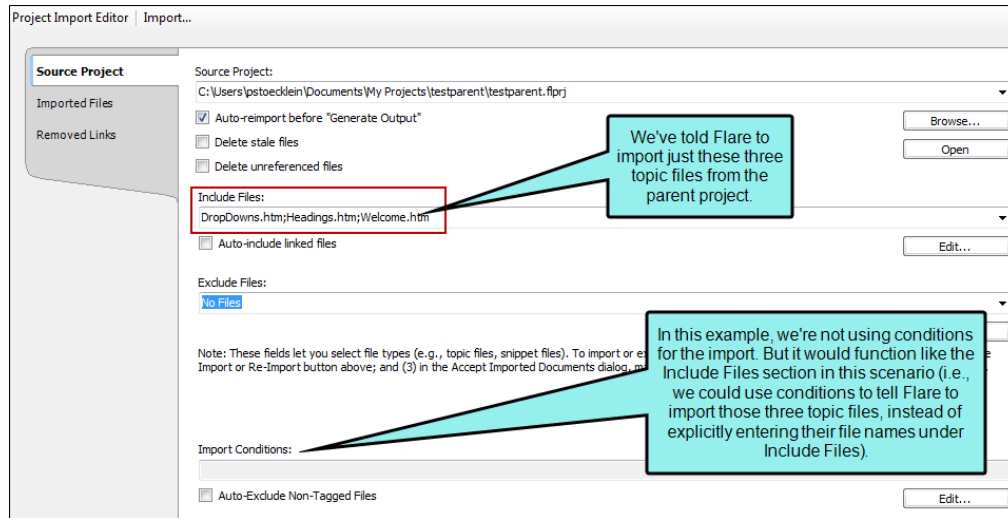
5. (Optional) If you want outdated files (e.g., files that have been deleted, renamed, or moved in the parent project) to be deleted from the child project automatically, click the **Delete stale files** check box.

☆ **EXAMPLE** You import a group of files from a parent project into a child project. Later, you delete some of the files from the parent project. Selecting this option will remove the stale files from the child project automatically when you reimport files. The stale files will be deleted when you reimport either manually (by clicking **Reimport** in the Project Import Editor) or automatically when you generate output (if you have selected **Auto-reimport before "Generate Output"** in the Project Import Editor).

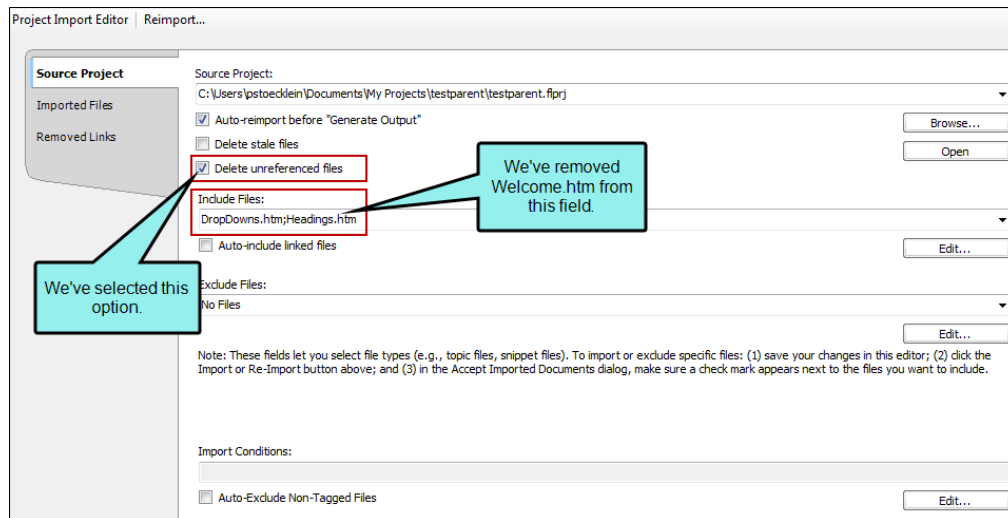
⚠ **WARNING** If you do not use this option, you must remember to delete the stale files in the child project manually. Otherwise, the files in the parent and child projects will not match.

6. (Optional) You can select **Delete unreferenced files**. This means that any files that were previously imported—but are not specified to be included (either under the Include Files or Import Conditions filters) in the reimport—will be marked for deletion in the Accept Imported Documents dialog and removed from the child project upon confirmation.

☆ **EXAMPLE** Previously you had the Project Import Editor set up like this, where you are importing three topic files from a parent project:



Later, after you've made lots of changes to the parent project, you open the child project again to import the latest files into it. However, you decide you no longer want to import the Welcome.htm topic into the child project (although it still exists in the parent project). So in the Project Import Editor, you remove it from the Include Files field. You also click **Delete unreferenced files**.



☆ When you tell Flare to reimport, the Accept Imported Documents dialog opens, showing the following:

The first two topics have been changed in the parent and are therefore newer. So they are marked to be re-imported, replacing the older versions of those topics in the child project.

But Flare sees that you've removed the reference to the Welcome.htm topic, so it is marked to be deleted.

Import	File	Status	Type
<input checked="" type="checkbox"/>	DropDowns.htm	Source is Newer	htm
<input checked="" type="checkbox"/>	Headings.htm	Source is Newer	htm
<input checked="" type="checkbox"/>	Welcome.htm	Delete unreferenced	htm

Welcome

Here is some xxgene
a topic. Replace this
own content. Here is
general text for a top
this with your own c
H...


After you click **Accept** in that dialog, the changed files are replaced and Welcome.htm is removed from the child project.

If you had not selected the "Delete unreferenced files" check box, the two changed files would still have been imported, but the old Welcome.htm topic would still remain in the child project.

This feature is slightly different from "Delete stale files." When you tell Flare to delete stale files, you automatically sync the parent project with the child project when you've removed files in the parent. So in the example above, the Welcome.htm file would automatically be removed from the child project if you had also removed it from the parent project.

7. Click **Next**.
8. Use the **Include Files** and **Exclude Files** fields to select the type of files to be included in the import or excluded from it. Click the down arrow next to the appropriate field and select the type of files. Completing the Include Files field is mandatory. Completing the Exclude Files field is optional.

If you want to import all of the files from the global project, select **All Files (*.*)**.

 **NOTE** You are not limited to importing all files of a single file type. The following steps explain how to add more than one file type to the field, as well as how to select specific files to be imported while excluding others.

9. (Optional) If you want to add more file types to one of the fields, click the **Edit** button under the appropriate field and use the following steps.
 - a. In the Import File Filter dialog click **Add**. The Import File Filter Designer dialog opens.
 - b. Select the file type that you want to add. You can also use standard wildcards (text between asterisks) to enter patterns directly into the **Pattern** field (or into the Include Files or Exclude Files fields).

☆ **EXAMPLE** You want to import only topic files that contain the word "Interface" in the file name. Rather than selecting to import all topic files and then later systematically deselecting the ones you do not really want in the import (via the Accept Imported Documents dialog), you can enter the following:

```
*Interface*.htm;*Interface*.html
```


Or maybe you want to import only files that have an extension starting with "fl," such as TOC files (.fltoc) and snippet files (.flsnp). In order to do this, you could enter *.fl* in the field, like the following:


```
*.fl*
```

- c. Click **OK**.
 - d. In the Import File Filter dialog, click **OK**.
10. (Optional) Let's say that the files you are importing contain links to other files. If you want Flare to automatically import those files as well, click **Auto-include linked files**. Not only will the immediate linked files be included, but any linked files from those files as well, and so on. It is a "domino" effect until no more linked files are found.
11. (Optional) In addition to specifying certain files or file types to include or exclude from the import, you can go a level deeper through the use of condition tags. If you have applied condition tags to files in your project, you can use the **Import Conditions** field to tell Flare to include or exclude certain files, depending on the condition tags that are applied to them at the file level. Simply click the **Edit** button to choose the conditions. In the Conditional Text dialog for this feature, you can use the Basic or Advanced mode for creating condition tag expressions. For more information about condition tags, see the online Help.
12. (Optional) If you use the "Import Conditions" option, you may want to make sure that files without condition tags in the source project are not included in the import. If that is the case, click **Auto-Exclude Non-Tagged Files**.

13. Click **Finish**. The Accept Imported Documents dialog opens. The files in the source project are listed (depending on whether the file types were included or excluded from the import in the previous steps).
14. (Optional) The Accept Imported Documents dialog provides you with one last look at the files to be imported, allowing you to make sure everything is correct and letting you change your mind on any of the files. If you recognize files in the dialog that should not be imported, you can click the check boxes next to the files you want to exclude (removing the check marks).

You can use the Select All and Clear All buttons as necessary. For example, if you only want to include a very small number of the files listed, you can click the **Clear All** button and then manually click the check boxes next to the files you want to include (this is quicker than individually deselecting each file that you want to exclude).

 **NOTE** If the current project already contains a file with the same name, the Status cell may be highlighted in green or red. Green shading indicates that the source file is newer. Red shading indicates that the local (or current) file is newer. If the file is identical in both projects, the check box is deselected by default.

 **TIP** You might find it useful to click on the column headings in the Accept Imported Documents dialog. Doing this reorganizes the contents in alphabetical order of the column that you click. For example, by clicking the **Status** column, you can easily group together all of the files that have red or green backgrounds (i.e., files that are newer in the local project or newer in the source project). This can be especially useful when reimporting project files.

15. Click **Accept**. If the current project already contains files with the same names, you may be asked if you want to replace the local copies. Select **Yes** if you do.






NOTE When you manually import files from the parent project, you will see the Accept Imported Documents dialog. This dialog lets you manually check or uncheck the files to be imported or excluded from the import. It is sort of a last chance to take a look at the files and make changes (include or exclude) before the actual import takes place. It also remembers whether you have checked or unchecked certain files during previous imports. However, when you use the "auto-reimport before generate output" feature, the import takes place automatically, so you do not see the Accept Imported Documents dialog. This means that *all files will be imported automatically based on the criteria you provide, regardless of whether you have previously deselected a file during a manual import*. This is one reason it is recommended you use conditions during Global Project Linking; with conditions, you are more assured of importing the correct files during an automatic import.


For example, let's say you have four topics—TopicA, TopicB, TopicC, and TopicD—and you have not conditioned any of them before importing. In your project import file, you specify that all topic files should be imported. You then perform the import. The Accept Imported Documents dialog opens, listing all four of your topics with check marks next to each one. However, you decide that you really do not want to import TopicD, so you remove the check mark next to it. After completing the manual import, TopicA, TopicB, and TopicC are imported, but TopicD is not. Later, let's say you have made changes to your files and need to reimport. After you click **Reimport**, you again see the Accept Imported Documents dialog. Again, check marks are automatically shown next to TopicA, TopicB, and TopicC. However, Flare remembered that you removed the check mark next to TopicD previously, so the check mark remains absent from it. You finish the reimport and everything goes according to plan.

Now let's say that after a few manual reimports, you decide to use the "auto-reimport" feature. Therefore, in your import file, you click the check box next to **Auto-reimport before "Generate Output"** and save. When you build the output, which topics are imported? The answer is TopicA, TopicB, TopicC, and TopicD.

In this example, one of the best ways to use the auto-reimport feature and be assured that the correct topics are included and excluded is to use condition tags at the file level. Therefore, in your source project, you create one condition tag (let's say you name it "YesImport") and you apply it to TopicA, TopicB, and TopicC. Then you create a second condition tag (let's say you name it "NoImport") and apply it to TopicD. Then in the child project, you open your import file. You click the **Edit** button next to **Import Conditions**, and within the dialog that opens you tell Flare to include the "YesImport" condition and exclude the "NoImport" condition. Now, the next time you build the output, which topics are imported? The answer is TopicA, TopicB, and TopicC.

 **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.

 **NOTE** The Imported Files tab in the Import Editor displays the files that have been imported previously.

 **NOTE** In the Project Import Editor, the Removed Links tab displays any files that were previously imported, but the link to the source project has since been removed. For example, let's say that you have imported several files from a source project. After awhile, you open one of those files in the project where the files were imported. You make a few changes and attempt to save it. Because Flare sees that a connection exists between the file and the source project, it prompts you with some options. One of the options is to continue to save your changes and remove the link from the source project. This means that future changes to the file need to be made in the current project, rather than in the source project. When you remove a link to a file in this way, that file is added to the Removed Links tab.

Reimporting Files


After you add a new Flare import file (with an .flimpfl extension), it is stored in the Imports folder in the Project Organizer. When you import files from another Flare project, this FLIMPFL file saves your settings. What happens if you make changes to the source project in the future?

If you selected the **Auto-reimport before "Generate Output"** check box when you first imported the files, the changes in the source project will be reimported automatically when you generate output in the current project. However, if you did not select this "Easy Sync" option, you can open the FLIMPFL file and reimport files from the source project manually. The following steps describe how to do this.


For more details about the way Global Project Linking works when different settings are in effect, see "Rules for Global Project Linking" on page 25.

I How to Reimport Files From Another Project

1. Open the Project Organizer.
2. Double-click the **Imports** folder.
3. Double-click the FLIMPFL import file that you created. The Project Import Editor opens.
4. (Optional) If necessary, change the settings on the **Source Project** tab.
 - **Source Project** If you want to change the project from which files will be imported, click **Browse**. In the dialog find and double-click the Flare project file (.flprj extension) from which you will be importing files.

 **NOTE** You can also manually enter the path in the Source Project field. However, if you enter a relative path, make sure the path is relative to the import file (located by default in the Projects/Imports subfolder), not the project file (which is located at the root level of the project files).

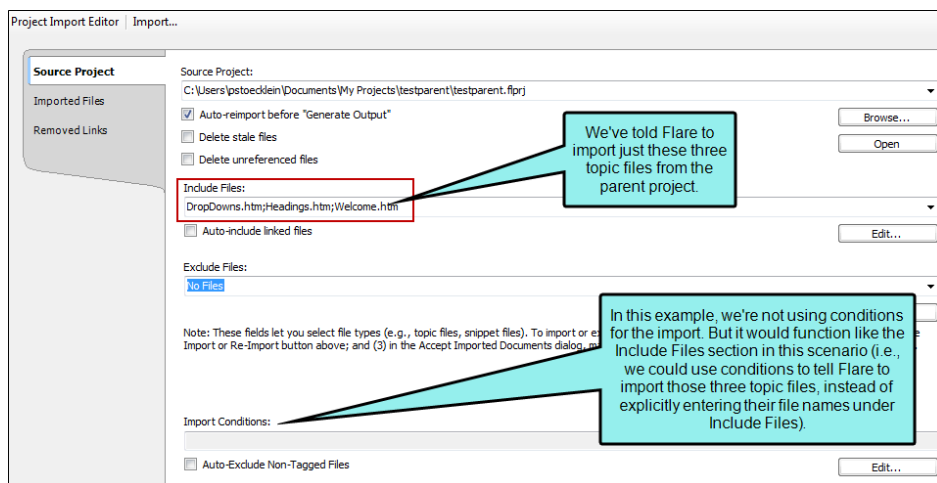
- **Auto-reimport before "Generate Output"** If you want future changes to the files in the source project to be reimported automatically, click this check box.
- **Delete stale files** If you want outdated files (e.g., files that have been deleted, renamed, or moved in the parent project) to be deleted from the child project during automatic reimports, click this check box.

 **EXAMPLE** You import a group of files from a parent project into a child project. Later, you delete some of the files from the parent project. Selecting this option will remove the stale files from the child project automatically when you reimport files. The stale files will be deleted when you reimport either manually (by clicking **Reimport** in the Project Import Editor) or automatically when you generate output (if you have selected **Auto-reimport before "Generate Output"** in the Project Import Editor).

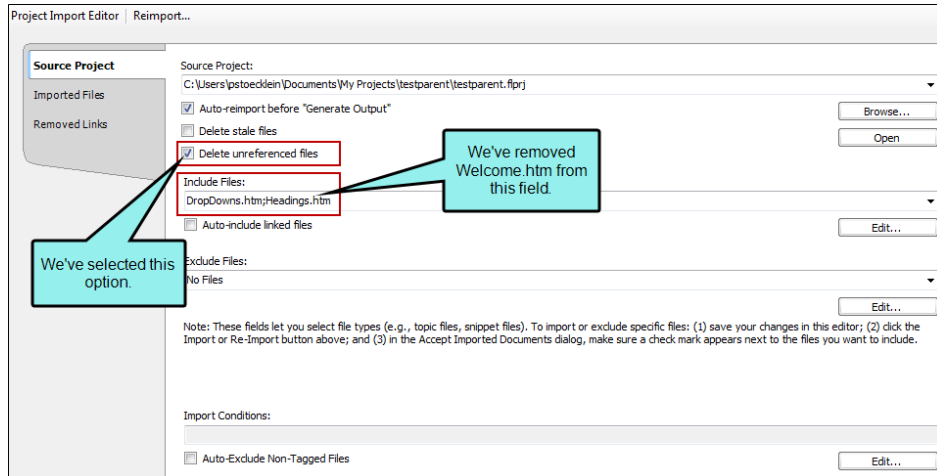
! **WARNING** If you do not use this option, you must remember to delete the stale files in the child project manually. Otherwise, the files in the parent and child projects will not match.

- **Delete unreferenced files** This means that any files that were previously imported—but are not specified to be included (either under the Include Files or Import Conditions filters) in the reimport—will be marked for deletion in the Accept Imported Documents dialog and removed from the child project upon confirmation.

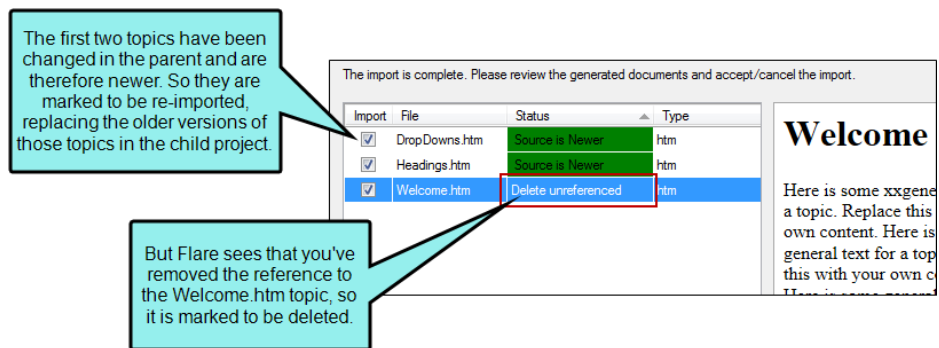
☆ **EXAMPLE** Previously you had the Project Import Editor set up like this, where you are importing three topic files from a parent project:



Later, after you've made lots of changes to the parent project, you open the child project again to import the latest files into it. However, you decide you no longer want to import the Welcome.htm topic into the child project (although it still exists in the parent project). So in the Project Import Editor, you remove it from the Include Files field. You also click **Delete unreferenced files**.



When you tell Flare to reimport, the Accept Imported Documents dialog opens, showing the following:




After you click **Accept** in that dialog, the changed files are replaced and Welcome.htm is removed from the child project.


If you had not selected the "Delete unreferenced files" check box, the two changed files would still have been imported, but the old Welcome.htm topic would still remain in the child project.

This feature is slightly different from "Delete stale files." When you tell Flare to delete stale files, you automatically sync the parent project with the child project when you've removed files in the parent. So in the example above, the Welcome.htm file would automatically be removed from the child project if you had also removed it from the parent project.

- **Include Files/Exclude Files** Use these fields to select the type of files to be included in the import or excluded from it. Click the down arrow next to the appropriate field and select the type of files. If you want to import all of the files from the global project, select **All Files (*.*)**.

 **NOTE** You are not limited to importing all files of a single file type. Items below explain how to add more than one file type to the field, as well as how to select specific files to be imported while excluding others.


- **Edit (Include Files/Exclude Files)** If you want to add more file types to one of the fields, click the **Edit** button under the appropriate field and use the following steps.
 - a. In the Import File Filter dialog click **Add**. The Import File Filter Designer dialog opens.
 - b. Select the file type that you want to add. You can also use standard wildcards (text between asterisks) to enter patterns directly into the **Pattern** field (or into the Include Files or Exclude Files fields).

 **EXAMPLE** You want to import only topic files that contain the word "Interface" in the file name. Rather than selecting to import all topic files and then later systematically deselecting the ones you do not really want in the import (via the Accept Imported Documents dialog), you can enter the following:

```
*Interface*.htm;*Interface*.html
```

Or maybe you want to import only files that have an extension starting with "fl," such as TOC files (.fltoc) and snippet files (.flsnp). In order to do this, you could enter *.fl* in the field, like the following:


```
*.fl*
```


- c. Click **OK**.
 - d. In the Import File Filter dialog, click **OK**.
- **Import Conditions** In addition to specifying certain files or file types to include or exclude from the import, you can go a level deeper through the use of condition tags. If you have applied condition tags to files in your project, you can use this field to tell Flare to include or exclude certain files, depending on the condition tags that are applied to them at the file level. Simply click the **Edit** button to choose the conditions. In the Conditional Text dialog for this feature, you can use the Basic or Advanced mode for creating condition tag expressions. For more information about condition tags, see the online Help.
 - **Edit (Import Conditions)** Clicking this button opens the Import Conditions dialog, which lets you specify which files to include or exclude in the import, based on the condition tags that are applied to them in the source project.
 - **Auto-Exclude Non-Tagged Files** If you use the "Import Conditions" option, you may want to make sure that files without condition tags in the source project are not included in the import. If that is the case, click this check box.
5. Click  to save your work.
 6. In the local toolbar of the Project Import Editor, click **Reimport**.

The Accept Imported Documents dialog opens. The files in the source project are listed (depending on whether the file types were included or excluded from the import in the previous steps).

7. (Optional) The Accept Imported Documents dialog provides you with one last look at the files to be imported, allowing you to make sure everything is correct and letting you change your mind on any of the files. If you recognize files in the dialog that should not be imported, you can click the check boxes next to the files you want to exclude (removing the check marks).

You can use the Select All and Clear All buttons as necessary. For example, if you only want to include a very small number of the files listed, you can click the **Clear All** button and then manually click the check boxes next to the files you want to include (this is quicker than individually deselecting each file that you want to exclude).



 **TIP** You might find it useful to click on the column headings in the Accept Imported Documents dialog. Doing this reorganizes the contents in alphabetical order of the column that you click. For example, by clicking the **Status** column, you can easily group together all of the files that have red or green backgrounds (i.e., files that are newer in the local project or newer in the source project). This can be especially useful when reimporting project files.


 **NOTE** If the current project already contains a file with the same name, the Status cell may be highlighted in green or red. Green shading indicates that the source file is newer. Red shading indicates that the local (or current) file is newer. If the file is identical in both projects, the check box is deselected by default.


 **NOTE** If you purposely excluded a particular file when performing a previous import, Flare remembers this and leaves the file deselected.

8. Click **Accept**.

If the current project already contains files with the same names, you may be asked if you want to replace the local copies. Select **Yes** if you do.

 **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.

 **NOTE** The Imported Files tab in the Import Editor displays the files that have been imported previously.

 **NOTE** In the Project Import Editor, the Removed Links tab displays any files that were previously imported, but the link to the source project has since been removed. For example, let's say that you have imported several files from a source project. After awhile, you open one of those files in the project where the files were imported. You make a few changes and attempt to save it. Because Flare sees that a connection exists between the file and the source project, it prompts you with some options. One of the options is to continue to save your changes and remove the link from the source project. This means that future changes to the file need to be made in the current project, rather than in the source project. When you remove a link to a file in this way, that file is added to the Removed Links tab.

Rules for Global Project Linking

If you use the Global Project Linking feature to import files from other projects, you should be aware of the various rules at play, depending on the settings you choose.

- **Include Files Required** The Include Files field is mandatory. You can use wildcards, add specific file names, and specify multiple kinds of file types to include.
- **Auto-reimport Memory** If you use the Auto-Reimport check box, Flare will remember to not import any files that you have told it to exclude (either by using the Exclude Files field or by deselecting check marks in the Accept Imported Documents dialog). Flare will only import files that you told it to include (or did not specifically tell it to exclude), thus bringing in any new files added to the source project or any changes to files that you have imported before. If a file in the current project is newer than the same file in the source project, the source file will not automatically replace the current file.
- **Exclude Files Not Required** The Exclude Files field is optional. You can use wildcards, add specific file names, and specify multiple kinds of file types to exclude.
- **Condition Tags Not Required, But Recommended** The Import Conditions section is optional. It lets you include or exclude files that have any condition tags applied to them at the file level. If you decide to take advantage of Global Project Linking, it is highly recommended that you use condition tags when doing so. By applying condition tags to your files (e.g., topics, images, stylesheets, page layouts), you can more easily control which files are imported into (or excluded from) a particular project. This is especially true if you elect to use the "auto-reimport" feature, which automatically imports files that you specify when you generate targets.

- **Condition Tags > Include Files** If you complete both the Include Files field and the Import Conditions section, the condition tags specified take precedence over what you have specified in the Include Files field.

☆ **EXAMPLE** If you tell Flare to include all topic files, but you have excluded condition tags that are applied to some of those topic files, then those topics will not be imported.

- **Exclude Files > Include Files** If you complete both the Include Files field and the Exclude Files field, the exclude files that are specified take precedence over the include files.

☆ **EXAMPLE** If you accidentally specify "All Topic Files" in both fields, then no topic files will be imported.

- **Exclude Files > Condition Tags > Include Files** If you complete the Include Files field, the Exclude Files field, and the Import Conditions section, the exclude files field takes highest precedence, then the condition tags, then the include files.

☆ **EXAMPLE** You have specified "All Topic Files" in the Include Files field. Furthermore, 30 percent of those files have an "advanced user" condition tag applied to them, and the other 70 percent have a "beginning user" condition tag applied. You have told Flare to include the files with the "advanced user" condition tags and to exclude the files with the "beginning user" condition tags. Finally, in the Exclude Files field, you have specifically typed the file name "My Big Topic.htm," which has an "advanced user" condition tag applied to it. In a case such as this, Flare imports all topic files that have the "advanced user" condition tag applied, except for the file "My Big Topic.htm," which it will not import.

- **Exclude Files > Auto-include Linked Files > Condition Tags** If you complete the Exclude Files field and the Import Conditions section, as well as select the "Auto-include linked files" check box, the exclude files takes highest precedence, then the auto-include files, and finally the condition tags.

☆ **EXAMPLE** You have specified "All Topic Files" in the Include Files field. Furthermore, 30 percent of those files have an "advanced user" condition tag applied to them, and the other 70 percent have a "beginning user" condition tag applied. You have told Flare to include the files with the "advanced user" condition tags and to exclude the files with the "beginning user" condition tags. You have also told Flare to automatically include files that are linked from the files that you are specifically importing. Finally, in the Exclude Files field, you have specifically typed the file name "My Big Topic.htm," which has a "beginning user" condition tag applied to it, but there is an "advanced user" topic that links to that file. In a case such as this, Flare will import all topic files that have the "advanced user" condition tag applied. It will also import any "beginning user" files that are linked from any "advanced user" files. However, because you have specifically told Flare to exclude the file "My Big Topic.htm," it will not be included in the import.

- **"Domino" Effect for Auto-include Linked Files** If you select the "Auto-include linked files" check box, Flare will import any linked files it finds, even beyond the first level. Not only will the immediate linked files be included, but any linked files from those files as well, and so on. It is a "domino" effect until no more linked files are found.

☆ **EXAMPLE** You have told Flare to import just one topic called "My Big Topic.htm," but you also tell it to automatically include any linked files. If "My Big Topic.htm" contains a cross-reference to a topic called "My Little Topic.htm" and it also contains a picture called "My Image.png," Flare will import all three of those files. Furthermore, let's say that "My Little Topic.htm" contains a hyperlink to yet another topic called "My Medium Topic.htm" and a link to a snippet called "My First Snippet.flshp." Therefore, Flare will also import those two files, and it will scan those files for further links that may need to be imported.

- **Files Without Condition Tags Included** If you use the Import Conditions section to include and exclude files based on condition tags applied to them, but a few files do not have condition tags, those files will be included in the import. However, you have the option of removing the check mark from those files in the Accept Imported Documents dialog to exclude them from the import.
- **Memory in Accept Imported Documents Dialog** After you perform an import the first time, all of the files in the Accept Imported Documents dialog will have check marks next to them, meaning they will be imported. However, you can deselect any specific files in that dialog if you decide not to import them. The next time you perform a reimport, those files will still show up in the Accept Imported Documents dialog, but they will remain unchecked because Flare remembers that you wanted to leave them out of a previous import. If you click the check boxes next to them to add check marks, those files will be imported.
- **New Files Contain Check Marks** After you have performed an import, you might add new files to the source project. If you then perform a reimport, those new files will automatically have check marks next to them in the Accept Imported Documents dialog (unless you specify beforehand that they should be excluded).
- **Green=Source Newer, Red=Local Newer** Let's say that you have imported several files from a source project, and then later you perform a reimport. If Flare recognizes that the same files in the two projects are not identical, it color-codes them in the Accept Imported Documents dialog. If the source file is newer than the file in the current (local) project, the file is shaded in green and a check mark appears next to it (indicating that it will be imported and therefore replace the local file). If the local file is newer than the file in the source project, the file is shaded in red in the Accept Imported Documents dialog, and a check mark does not appear next to it (indicating that the file will not be imported, replacing the local file, unless you manually add a check mark).

APPENDIX

PDFs

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

I Tutorials

Autonumbers Tutorial

Back-to-Top Button Tutorial

Context-Sensitive Help Tutorial

Custom Toolbar Tutorial

eLearning Tutorial—Basic

eLearning Tutorial—Advanced

Getting Started Tutorial

Image Tooltips Tutorial

Lists Tutorial

Meta Tags Tutorial

Micro Content Tutorial—Basic

Micro Content Tutorial—Advanced

Responsive Output Tutorial

Single-Sourcing Tutorial

Snippet Conditions Tutorial

Styles Tutorials

Tables Tutorial

Word Import Tutorial

| Cheat Sheets

Context-Sensitive Help Cheat Sheet

Folders and Files Cheat Sheet

Learning & Development Cheat Sheet

Lists Cheat Sheet

Micro Content Cheat Sheet

Print-Based Output Cheat Sheet

Search Cheat Sheet

Shortcuts Cheat Sheet

Structure Bars Cheat Sheet

Styles Cheat Sheet

I User Guides

Accessibility Guide

Analysis and Reports Guide

Architecture Guide

Autonumbers Guide

Branding Guide

Condition Tags Guide

Context-Sensitive Help Guide

Eclipse Help Guide

eLearning Guide

Getting Started Guide

Global Project Linking Guide

HTML5 Guide

Images Guide

Import Guide

Indexing Guide

Key Features Guide

Lists Guide

*MadCap Central Integration
Guide*

Meta Tags Guide

Micro Content Guide

Navigation Links Guide

Plug-In API Guide

Print-Based Output Guide

Project Creation Guide

QR Codes Guide

*Reviews & Contributions With
Contributor Guide*

Scripting Guide

Search Guide

SharePoint Guide

Skins Guide

Snippets Guide

Source Control Guide: Git

*Source Control Guide:
Perforce Helix Core*

*Source Control Guide:
Subversion*

*Source Control Guide: Team
Foundation Server*

Styles Guide

Tables Guide

Tables of Contents Guide

Targets Guide

Template Pages Guide

Templates Guide

Topics Guide

Touring the Workspace Guide

*Transition From FrameMaker
Guide*

*Translation and Localization
Guide*

Variables Guide

Videos Guide

What's New Guide