MadCap Software

Source Control 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 Introduction 5

What is Source Control/Multi-Authoring? 6
Is the Source Control Application Inside Flare? 7
Which Source Control Programs are Integrated with the Application? 8
Who is Involved with Source Control? 9
Which Files Can Be Included in the Integrated Source Control? 10
Common Source Control Terms 12
Source Control Icons 13
General Workflow (Multi-Author Team) 14
General Workflow (Single Author) 16
Additional Source Control Tasks and Features 17

CHAPTER 2 Binding a Project to Source Control 21

CHAPTER 3 Importing from Source Control 25

CHAPTER 4 Getting the Latest Version 29

Automatic Get 30
Manual Get 31

CHAPTER 5 Checking Out Source Control Files 33

CHAPTER 6 Checking In Source Control Files 37

CHAPTER 7 Merging Source Control Files 39

Icons for File Differences 48

CHAPTER 8 Adding Files to Source Control 51
CHAPTER 9 Requesting a User to Check In Files 53
CHAPTER 10 Requesting a User to View Files 55
CHAPTER 11 Additional Source Control Tasks 57
Disabling the Get Latest Prompt for Source Control 58
Enabling Source Control Status Checks 59
Rolling Back to an Earlier Version of a File 60
Setting Color Options for Project File Differences 62
Unbinding a Project from Source Control 63
Undeleting Source Control Files 64
Undoing a Check-Out of Source Control Files 66
Using an API to Integrate Source Control 67
Viewing Differences in Source Control Files 68
Viewing Files That are Checked Out 74
Viewing the History of Source Control Files 78
APPENDIX PDF Guides 79

CHAPTER 1

Introduction

Flare includes support for every source control tool on the market by virtue of the fact that Flare uses a wide open architecture. Instead of using proprietary files and databases, all content and project files in Flare are stored as independent XML files. This means that Flare projects are compatible with all source control systems. In addition to this native support for all source control applications, Flare provides integrated support for some source control applications as well.

This chapter discusses the following:

What is Source Control/Multi-Authoring? 6
Is the Source Control Application Inside Flare? 7
Which Source Control Programs are Integrated with the Application? 8
Who is Involved with Source Control? 9
Which Files Can Be Included in the Integrated Source Control? 10
Common Source Control Terms 12
Source Control Icons 13
General Workflow (Multi-Author Team) 14
General Workflow (Single Author) 16
Additional Source Control Tasks and Features 17

What is Source Control/Multi-Authoring?

Source control is a process that allows you to store your project files in a central location and determine which individuals have access to those files. By using a system where individuals must "check in" and "check out" files to edit them, you can ensure that there are no file conflicts when many individuals are working on the same project (multi-authoring teams). You can also revert back to earlier versions of source control files if necessary. Source control is even useful for single authors because it provides a means for maintaining a constant backup of all project files.

Is the Source Control Application Inside Flare?

No. Flare does not use a proprietary source control system within the Flare application. Instead, Flare allows you to connect (or "bind") your project to an existing third-party source control application that is separate from Flare.

Which Source Control Programs are Integrated with the Application?

Built-in support is available for Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, and Apache Subversion. In addition, an API allows you to configure your project for integration with other source control tools. You also have the option of manually adding your project files to any source control tool, rather than using 's built-in integration.

Who is Involved with Source Control?

This depends on how you work in your company. Someone (perhaps a network administrator) needs to set up your source control program (such as Microsoft Visual SourceSafe) and create the necessary database(s) in it. This individual may also set up the "rights" to all persons who have access to the necessary databases and files. If you are working on a multi-author team, each writer on the team needs to have a local copy of the Flare project and will be responsible for checking in and checking out files as necessary.

Which Files Can Be Included in the Integrated Source Control?

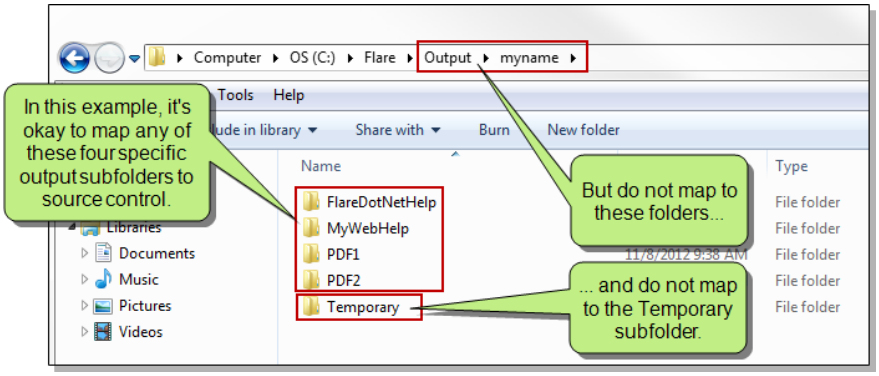
The files that can be included in source control depends on whether you are adding them manually or using the built-in integration.

If Including Files Manually in Your Source Control Client

If you are manually adding files to a source control application (rather than using the built-in Flare integration), you can transfer any appropriate files in a Flare project. Appropriate files include the following.

- The main FLPRJ file
- All files from the Content folder
- All files from the Project folder, except those in the Project/Users subfolder
- Files from a specific Output subfolder

  You might transfer output files to source control, for example, if you are integrating your Help into a desktop application, and programmers need access to your output files to build the application. However, if you do this, make sure you map the source control to the appropriate output subfolder, and not to the Temporary subfolder or to the main Output folder, which contains the Temporary subfolder.



If Using the Source Control Integration in the Flare Interface

If you are using Flare's built-in source control integration, you can transfer the following types of files.

- The main FLPRJ file
- All files from the Content folder
- All files from the Project folder, except those in the Project/Users subfolder

With the built-in integration, you cannot transfer the output files. In an ideal workflow, you would only need to transfer the content and project files mentioned, and when you are ready to generate output, you would publish the output files to another location where users can access them (such as a website or network drive).

Common Source Control Terms

Following are definitions for some of the common phrases used in Flare's built-in source control system.

- **Bind** This means to connect your project to an external source control application (such as Microsoft Visual SourceSafe). After doing this, you can then take advantage of all the automated source control tasks (such as check in, check out, instant messaging, and so on).
- **Check In** This means to copy your local Flare files to the source control application, overwriting the copies stored in the source control tool. This enables other authors on your team to access and edit the latest files.
- **Check Out** This means to copy the latest source control files to your local Flare project and remove the "Read Only" designation from them so that you can edit the files.
- **Get Latest Version** This means to copy the latest source control files to your local project without necessarily checking out the files. This means that the "Read Only" designation will remain associated with the files until you check them out.

Source Control Icons

Following are descriptions for the primary icons that you may see next to files when using source control.

Checked Out
This indicates that the file has been checked out from source control. You can check in the file when you are ready (if you are the user who has checked it out).

New File (Pending Add)

This indicates that you have a file in your project but have not yet added it to the integrated source control application. This might occur, for example, if you create a new topic and do not add the file to source control during the topic creation process. To resolve this, simply right-click on the file and select **Source Control>Add**.

Not Checked Out

This indicates that the file is added to source control but is not currently checked out, which means that it contains a "Read Only" designation in its properties. In order to edit the file, you must check it out from source control.

Checked Out to Other User
This indicates that the file is currently checked out to another user. You can send a request to the user, asking that person to check in the file so that you can check it out.

Out of Date
This indicates that the file is not current (i.e., the local copy of the file is older than the source control copy). This might happen, for example, if another user checks out the file, makes changes to it, and checks it back in to source control. If this occurs, you can check the file out or get the latest version of the file from source control.

Deleted
This indicates that the file is deleted from source control, but you still have a copy of the file on your local machine. If necessary, you can add the file to source control.

General Workflow (Multi-Author Team)

Following are the basic steps for using Flare's source control integration for a multi-author team.

1. **Set Up Source Control Application** Install the source control application and set up the database and any folders where you want to place the project. Also, assign permissions to the proper individuals. This task may be performed by someone else, such as a network administrator.
2. In Flare, do one of the following, depending on whether you are creating a new Flare project or using an existing one.
   - **Create and Bind New Project** Create a new Flare project and simultaneously bind it to an integrated source control application. For more information see the online Help.

     OR
   - **Bind Existing Project** If you have an existing Flare project on your local machine, bind the project to the source control application. See "Binding a Project to Source Control" on page 21.

   Note: Only one of the authors needs to perform this task. The other authors on the team can follow Step 3 below.
3. **Import Project** Other users on the team can easily import the project from source control. This creates a copy of the Flare project on the author's computer. See "Importing from Source Control" on page 25.
4. **Get Latest Version** After you bind a Flare project to a source control application, you can get the latest version of any of the source control files. When you do this, you are copying the most current files stored in the source control application to your local Flare project without necessarily checking out the files. This means that the "Read Only" designation will remain associated with the files until you check them out. When working with other authors, it is a good idea to do this frequently (e.g., at the beginning of each day) in order to ensure you have the latest changes that those authors have made, and that they have the latest changes you've made. See "Getting the Latest Version" on page 29.
5. **Check Out Files** When you need to work on any of the Flare project files, you can check them out. Checking out files means to copy the latest source control files to your local Flare project and remove the "Read Only" designation from them so that you can edit the files. A red check mark is displayed next to each file that is checked out. See "Checking Out Source Control Files" on page 33.

6. **Check In Files** When you are finished editing files, you can check them in to source control. Checking in a file overwrites the old copy of the file in the source control database with the new one from your local machine. So even if others will not be working on that file, it is a good idea to periodically check in files so that you have a backup in source control. See "Checking In Source Control Files" on page 37.

   Merging If you are set up in a source control application to allow multiple authors to have the same files checked out, there may be times when you need to merge changes from different authors when checking in a file. The merge occurs automatically if there are no conflicting changes (i.e., changes do not occur in the same location). If there are conflicting changes, a dialog opens, allowing you to determine how changes are merged. See "Merging Source Control Files" on page 39.

7. **Add Files** When working in a project that is connected to a source control application, there may be occasions when you have files in your local copy of the project that are not yet part of the source control copy. For example, when you add a new topic in your local copy of the project, that file will not be included in the source control copy of the project until you add it. See "Adding Files to Source Control" on page 51.
8. **Request to Check In Files** If you discover that another user already has that file checked out, you can automatically send a request to that user to check in the file. This is a completely automated process involving instant messaging or email. Not only are the messages sent to the appropriate individuals automatically, but the mechanisms for checking in and checking out the necessary files are built in to those messages. This means that you and the other authors on your team do not need to hunt down the correct files manually. See "Requesting a User to Check In Files" on page 53.
9. **Request to View Files** If you want another author to review a file and provide comments, this can be done by performing a request to view files. Similar to the process for requesting check-ins, this feature uses instant messaging to streamline the workflow. See "Requesting a User to View Files" on page 55.

General Workflow (Single Author)

Following are the basic steps for using Flare's source control integration for a single author.

1. **Set Up Source Control Application** Install the source control application and set up the database and any folders where you want to place the project. Also, assign permissions to the proper individuals. This task may be performed by someone else, such as a network administrator.
2. In Flare, do one of the following, depending on whether you are creating a new Flare project or using an existing one.
   - **Create and Bind New Project** Create a new Flare project and simultaneously bind it to an integrated source control application. For more information see the online Help.

     OR
   - **Bind Existing Project** If you have an existing Flare project on your local machine, bind the project to the source control application. See "Binding a Project to Source Control" on page 21.
3. **Check Out Files** When you need to work on any of the Flare project files, you can check them out. Checking out files means to copy the latest source control files to your local Flare project and remove the "Read Only" designation from them so that you can edit the files. A red check mark is displayed next to each file that is checked out. See "Checking Out Source Control Files" on page 33.
4. **Check in Files** When you are finished editing files, you can check them in to source control. Checking in a file overwrites the old copy of the file in the source control database with the new one from your local machine. So even if others will not be working on that file, it is a good idea to periodically check in files so that you have a backup in source control. See "Checking In Source Control Files" on page 37.
5. **Add Files** When working in a project that is connected to a source control application, there may be occasions when you have files in your local copy of the project that are not yet part of the source control copy. For example, when you add a new topic in your local copy of the project, that file will not be included in the source control copy of the project until you add it. See "Adding Files to Source Control" on page 51.

Additional Source Control Tasks and Features

Following are some additional tasks and features related to source control.

- **History** You can view the history of a particular source control file. See "Viewing the History of Source Control Files" on page 78.
  - **Version** Displays a number for each version of the file. The lower the number, the older the version. The higher the number, the more recent the version.
  - **Users** Displays the name of the user who has been working on the file.
  - **Date** Displays the date and time when the action has occurred.
  - **Action** Displays the action that has taken place for the file (e.g., check in).
  - **Comment** Displays the comment (if any) associated with the file. A comment can be added to a file when you check that file in to source control. This enables you to maintain an audit trail for the file's history.
  - **Get Selected Version** You can perform a "get" of a particular version of a file, thus rolling back to that version of the file. See "Rolling Back to an Earlier Version of a File" on page 60 and "Getting the Latest Version" on page 29.
  - **Show Differences** You can view the differences between two versions of a file. If you select one row in the History dialog and view the differences, you will see the content differences between the version that you selected and the version of the file in your local copy of the Flare project. If you select two files in the History dialog (by holding down the CTRL key) and view the differences, you will see the content differences between those two versions of the file.

- **Differences** You can view differences between files in the following ways. For more information see "Viewing Differences in Source Control Files" on page 68.
  - **Two Versions of Same Source Control File (History/Roll Back)** You can view code and content differences between two source control versions of the same file. This is useful if you need to roll back to an earlier version of a file. See "Rolling Back to an Earlier Version of a File" on page 60 and "Getting the Latest Version" on page 29.

    EXAMPLE

    Let's say that you have been working on a particular topic for a few days. Each day you check out the latest version of that topic file from source control, make your changes, and check the file back in to source control at the end of the day. At a certain point, you determine that you need to "roll back" to an earlier version of the file, using it to replace the latest version. Therefore, you use this feature to view the highlighted differences between the current version and an older version of the file. Once you have identified the older version that you want to use, you can perform a "get" of that version.

  - **Two Different Source Control Files** You can view code and content differences between the source control version of two different files.

    EXAMPLE

    Let's say that you have two topics that are quite similar in some aspects, but different in others. Rather than trying to scan the two topics with your eyes to determine the differences, you can select the two files in the Flare interface and use this feature. Flare will retrieve the source control versions of the two topic files and display the content and code of each side-by-side, highlighting the areas where differences occur.

- **Local Versus Source Control Version of a File** You can view code and content differences between the local version of a file and the source control version of that file.

  EXAMPLE

  Let's say that you check out a procedure topic from source control and then add some lines of text to your local copy of that topic file. You save your changes. Later that day, you want to revisit the new content that you wrote, but you cannot remember exactly which lines of text you added. Therefore, you use this feature to highlight the text differences between your local checked-out version of the file and the version stored in the source control application. The new lines of text are highlighted on the side displaying the local version of the file.

- **Local Versus Source Control Version of All Files in a Folder** You can view file differences between the local version of the files in a folder and the source control version. Most likely, you will use this to view all of the difference between the local files and source control files in either your Content Explorer or Project Organizer.

  EXAMPLE

  Let's say that you have checked out all of the files for a Flare project. During the course of the day, you add several new topics to the project, but you forget to add them to source control during the creation process. At the end of the day, you check in all of the files in the project to source control. Afterwards, you realize that you forgot to add the new files to source control. Therefore, you use this feature to see the file-level differences between your local copy of the Content Explorer root folder and the source control copy. The differences are color coded, so you can easily identify the files in question. (By default, the files that are included only in your local copy are green, and the files that are included only in source control are red.) You can then right-click on the files that were added only to the local copy, and you can select to add them to source control.

- **Locks** If you are using Microsoft Team Foundation Server,Apache Subversion, or Perforce, you can lock files so that other users cannot check in those same files when you have them checked out. You might use this feature to prevent conflicting changes from occurring and having to merge files as a result. You can set files to be locked automatically when you check out files. This can be done from the Options dialog (Source Control tab). Another option is to manually set files to be locked whenever you check out files. This can be done from the Check Out dialog. See "Checking Out Source Control Files" on page 33.
- **Unbind** You can unbind ("disconnect") a project from source control. See "Unbinding a Project from Source Control" on page 63.

- **View Checked-Out Files** You can use the Pending Check-Ins window pane to view all of the files that you have checked out and need to be checked in, as well as files that others have checked out. You can use the File List window pane to view files that are checked out by other users. See "Viewing Files That are Checked Out" on page 74.
- **Automatic Status Checks** If you are using source control integration in Flare, you can check for frequent status changes automatically. You can specify the number of minutes and seconds when you want Flare to ping the source control repository and get status changes for files that have been checked out, checked in, moved, deleted, etc. The upside of this feature is that you can ensure that the source control status information is always up to date. The downside is that you may experience slower performance due to this constant communication over the network. See "Enabling Source Control Status Checks" on page 59.
- **Disable Prompt for Get Latest** By default, when you open a project that is bound to source control, a message automatically asks if you wanted to get the latest version of files. However, you can disable this prompt in the Options dialog. See "Disabling the Get Latest Prompt for Source Control" on page 58.
- **API** If you are working with a source control application other than those directly supported in Flare (i.e., Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, Apache Subversion), you can use the Microsoft Source Code Control API (SCC API) to integrate that application with your Flare project. See "Using an API to Integrate Source Control" on page 67.

Note: It is not mandatory that you use the SCC API to use a source control application for your Flare project files. It is only necessary if you want to take advantage of the source control integration features inside Flare. Otherwise, you can manually add your Flare project files to any source control application.

CHAPTER 2

Binding a Project to Source Control

Use the following steps if you have already created a Flare project and want to bind ("connect") it to a source control application (Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, Apache Subversion, or another program via an API).

HOW TO BIND A PROJECT TO SOURCE CONTROL

1. Open the 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 **Properties** section select **Project Properties**.
   - **Menu** Select **Project>Project Properties**.
   - **Project Toolbar** Click Project Properties...

     To open the Project toolbar from the menu view, select **View>Toolbars>Project**.

   The Project Properties dialog opens.
3. Select the **Source Control** tab.
4. Click **Bind Project**. The Bind Project dialog opens.

5. 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 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**.

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**.

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**.

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**.

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**.

Note: In order to use this option, you must first employ the Microsoft SCC API. See "Using an API to Integrate Source Control" on page 67.

6. In the Bind Project dialog, click **OK**.
7. If the Log In dialog opens, complete the **User name** and **Password** fields and click **OK**.Copies of the Flare files are created and added to the folder you specified.
8. In the Project Properties dialog, click **OK**. The project is connected to the source control application, and you can now check files in and out as necessary.

Note: You can also bind a new Flare project to source control while creating it.

CHAPTER 3

Importing from Source Control

This chapter focuses on importing a Flare project from source control. You might use this method, for example, if you are working on a multi-author project and another member of the team has placed the Flare project in a source control tool such as Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, or Apache Subversion. For more information see "Introduction" on page 5.

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 4

Getting the Latest Version

After you bind a Flare project to a source control application, you can get the latest version of any of the source control files. When you do this, you are copying the most current files stored in the source control application to your local Flare project without necessarily checking out the files. This means that the "Read Only" designation will remain associated with the files until you check them out.

Following are steps for getting the latest version of all files in a project automatically, as well as steps for getting the latest version of files manually.

This chapter discusses the following:

Automatic Get 30

Manual Get 31

Automatic Get

You might use this option if you are working with a team of authors and want to make sure that you include the latest changes from other writers in the output (without having to manually get those files).

With this option:

- You will not be prompted before the "get" is performed.
- Flare will not get the latest copy of the files in the Targets folder, because that would conflict with the generation of the output.
- Conflicts with files will not cause local files to be overwritten. Therefore, if your local files are writable or already checked out, those files will be kept, rather than overwritten with the source control files.

HOW TO GET THE LATEST VERSION OF SOURCE CONTROL FILES AUTOMATICALLY

1. Open the target.
2. In the Target Editor, select the **General** tab.
3. Select the check box labeled **Automatically get latest version of all files before generating the target**.
4. Click to save your work.
5. Build the target.

Note: The "automatic get" feature is not supported if you are building output using the command line, as opposed to the Flare interface.

Manual Get

You can manually get the latest version of all of the files in the Flare project or specific files only.

HOW TO GET THE LATEST VERSION OF SOURCE CONTROL FILES MANUALLY

1. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files in the window pane that you want to check out, right-click, and select **Source Control>Get Latest Version** (for selected files) or **Source Control>Project>Get Latest Version All** (for all files in the project).
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Get Latest Version** (for selected files). Alternatively, in the **Project** section you can click **Get Latest Version All** (for all files in the project).
   - **Menu** Select **File>Source Control>Get Latest Version** (for selected files) or **File>Source Control>Project>Get Latest Version All** (for all files in the project).
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Get Latest Version** (for selected files) or **Project>Get Latest Version All** (for all files in the project).

     To open the Standard toolbar from the menu view, you can select **View>Toolbars>Standard**.
2. If the local and server files are the same, a message tells you so. Click **OK**.

   If the local file is different form the file on the server, the Resolve Conflicts dialog opens. If you want to accept all of the differences between the local and server files, thus merging them, click **Auto Merge All**. If you want to review the differences in the files side by side and resolve each conflict, click **Resolve**. For more information about merging files and resolving conflicts, see "Merging Source Control Files" on page 39.

Note: By default, when you open a project that is bound to source control, a message automatically asks if you wanted to get the latest version of files. However, you can disable this prompt in the Options dialog. See "Additional Source Control Tasks" on page 57.

CHAPTER 5

Checking Out Source Control Files

When you need to work on any of the Flare project files, you can check them out. Checking out files means to copy the latest source control files to your local Flare project and remove the "Read Only" designation from them so that you can edit the files. A red check mark is displayed next to each file that is checked out.

You can check files out automatically or manually.

HOW TO CHECK OUT FILES FROM SOURCE CONTROL AUTOMATICALLY

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

   The Options dialog opens.
2. Select the **Source Control** tab.

   Note: This tab will not be visible if your project is not yet bound to source control. See "Binding a Project to Source Control" on page 21.
3. Click the check box **Automatically check file out from source control when saving file changes**. A check mark in the box indicates that the feature is enabled.
4. (Optional) If you are using Microsoft Team Foundation Server, Apache Subversion, or Perforce and want files that are automatically checked out to be locked as well, select **Lock files when automatically checking out**.

   Locking files means that other users cannot check in those same files when you have them checked out. You might use this feature to prevent conflicting changes from occurring and having to merge files as a result.
5. Click **OK**.

   Whenever you open a file that is not checked out, make changes, and save, not only will your changes be saved but the file will automatically be checked out to you.

HOW TO CHECK OUT FILES FROM SOURCE CONTROL MANUALLY

1. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files in the window pane that you want to check out, right-click, and select **Source Control>Check Out** (for selected files) or **Source Control>Project>Check Out All** (for all files in the project).
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Check Out** (for selected files). Alternatively, in the **Project** section you can click **Check Out All** (for all files in the project).
   - **Menu** Select **File>Source Control>Check Out** (for selected files) or **File>Source Control>Project>Check Out All** (for all files in the project).
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Check Out** (for selected files) or **Project>Check Out All** (for all files in the project).

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

   The Check Out dialog opens. The selected files are listed with check boxes next to them.
2. Make sure to click the check box next to each file you want to check out so that it contains a check mark.
3. (Optional) If you are using Microsoft Team Foundation Server or Apache Subversion and want the files to be locked as well, select **Lock files**.

   Locking files means that other users cannot check in those same files when you have them checked out. You might use this feature to prevent conflicting changes from occurring and having to merge files as a result.
4. Click **Check Out**.

CHAPTER 6

Checking In Source Control Files

When you are finished editing files, you can check them in to source control. Checking in a file overwrites the old copy of the file in the source control database with the new one from your local machine. So even if others will not be working on that file, it is a good idea to periodically check in files so that you have a backup in source control.

HOW TO CHECK IN FILES TO SOURCE CONTROL

1. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files in the window pane that you want to check in, and in the local toolbar click Check In...
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Check In** (for selected files). Alternatively, in the **Project** section you can click **Check In All** (for all files in the project).
   - **Menu** Select **File>Source Control>Check In** (for selected files) or **File>Source Control>Project>Check In All** (for all files in the project).
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Check In** (for selected files) or **Project>Check In All** (for all files in the project).

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

   The Check In dialog opens. The selected files are listed with check boxes next to them.
2. (Optional) Enter an optional comment tied to the check-in. This enables you to keep an audit trail for a file. The comment can then be viewed from the History dialog, which can be accessed from the Source Control ribbon, the File menu, or the Source Control button.
3. (Optional) If you want to see all files with pending changes (rather than only those you selected), click.
4. Make sure to click the check box next to each file you want to check in so that it contains a check mark.
5. If you want to keep the files checked out of source control, select **Keep Checked Out**. Doing this will overwrite the source control copies of the files so that they have the latest changes, but it allows you to continue working on the files. This is a useful option if you are the only author working on the files in question.
6. Click **Check In**.

   If no other users have also made changes to the file and checked it in while you were working on it, your version of the file is checked in.

   However, if that is the case, the Resolve Version Conflict dialog opens to let you know that another user has already checked in the file with changes. You can merge the files automatically if there are no conflicting changes (i.e., changes do not occur in the same location in the file). If there are conflicting changes, you can use the Merge Changes dialog to determine how changes are merged. See "Merging Source Control Files" on page 39.

CHAPTER 7

Merging Source Control Files

If you are set up in a source control application to allow multiple authors to have the same files checked out, there may be times when you need to merge changes from different authors when checking in a file. The merge occurs automatically if there are no conflicting changes (i.e., changes do not occur in the same location in the file). If there are conflicting changes, a dialog opens, allowing you to determine how changes are merged.

HOW TO MERGE SOURCE CONTROL FILES

1. Go through the process of getting the latest version of files from source control or checking in files. See "Getting the Latest Version" on page 29 and "Checking In Source Control Files" on page 37. If your local copy of the file is different from the server copy (e.g., another author has already checked in the same file), the Resolve Conflicts dialog opens.
2. Click **Auto Merge All**. If changes from the other author do not conflict with your changes, this will merge all changes. A message lets you know that a backup of your local copy has been made. This allows you to restore that file if you do not want to keep the merged version. You do not need to complete the rest of the steps below.

   However, if your changes conflict with those from another author, a message displays to tell you. In this case, continue with the next step.
3. Click **OK** on the conflict message.
4. In the Resolve Conflicts dialog click **Resolve**. The Resolve Version Conflict dialog opens. From this dialog, you can choose from the following options.
   - **Merge changes for me** Automatically merges changes within the same file that are not part of the same element. If changes have been made to the same element (for example, the same <p> tag or <h1> tag), Flare will display a prompt to merge the changes using the merge tool.
   - **Merge changes in merge tool** Opens a merging interface, which lets you see exactly what changes were made and choose which to keep.
   - **Undo my local changes** Automatically removes your changes and keeps changes from other authors.
   - **Discard external changes** Automatically removes changes from other authors and keeps your changes.
5. If you selected the first option, the Merge Changes dialog opens. Use this dialog to view and select changes. You can take actions in the following ways.
   - **Right-click icon** You can right-click on the icon next to a change on either the server or local side. This displays a context menu, which lets you either keep or ignore a particular change.
   - **Left-click icon** You can left-click on the icon next to a change on either the local or server side. When you do this, the change from one side is kept and the change from the other side is ignored. If you left-click the icon again, the results are toggled; the first change is ignored and the other is kept.
   - **Type content** If you want to use your changes as well as those from another author, and even tweak the paragraph a bit more, you can click in the area at the bottom of the dialog and simply type content.
   - **Previous/next conflict** When you are finished resolving the first conflict, you can use the "Previous Conflict" and "Next Conflict" buttons at the bottom of the dialog to work on other conflicts in the file.

Note: If you selected "Merge as Text" in the local toolbar and are working in the code, you can click on text with a hatched background to keep the change in it. After you click on text with a hatched background, the hatched lines are removed, leaving a solid color.

6. After all conflicts have been resolved, a message lets you know that a backup of your local copy has been made. This allows you to restore that file if you do not want to keep the merged version. Click **OK**.

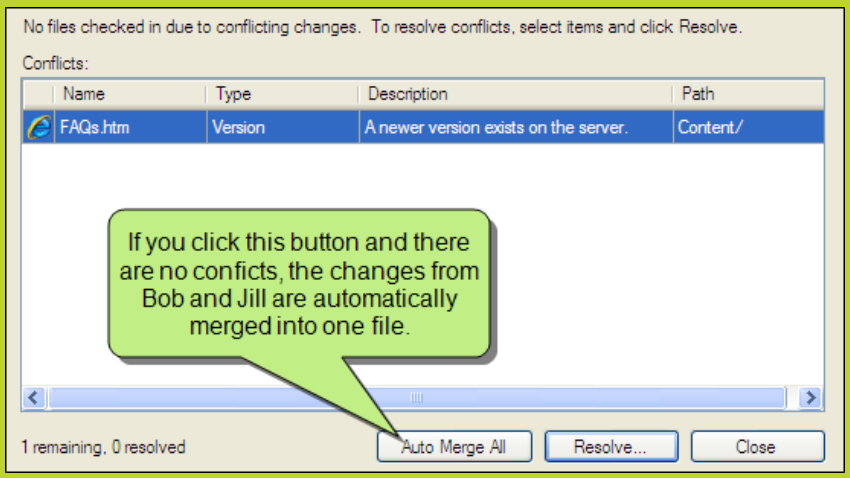
EXAMPLE—NO CONFLICTING CHANGES

Let's say two authors—Bob and Jill—are working on the same project, using source control to manage the files.

Bob checks out the topic "FAQs" and starts making changes to it.

Jill also checks out the "FAQs" topic and makes some changes to it. Jill's changes are in a different location in the topic than Bob's changes; there are no conflicts. She finishes before Bob and checks the file in to source control.

Bob finishes his changes and tries to check in the file. Before the file can be checked in, Bob is prompted with a dialog, indicating that changes from another author have already been checked in.



Bob selects **Auto Merge All**. The changes from Bob and Jill are now both shown in the merged topic.

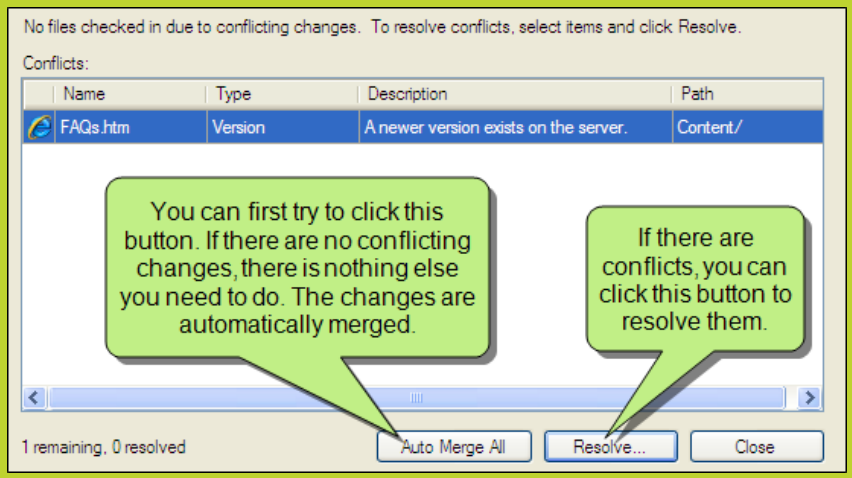
EXAMPLE—CONFLICTING CHANGES

Let's say two authors—Bob and Jill—are working on the same project, using source control to manage the files.

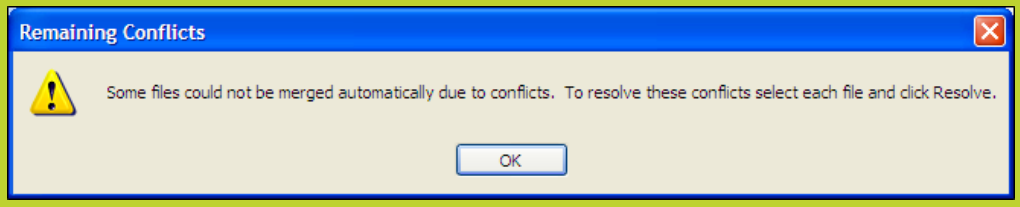
Bob checks out the topic "FAQs" and starts making changes to it.

Jill also checks out the "FAQs" topic and makes some changes to it. Jill's changes are in the same paragraph in the topic as Bob's changes; thus, there is a conflict. She finishes before Bob and checks the file in to source control.

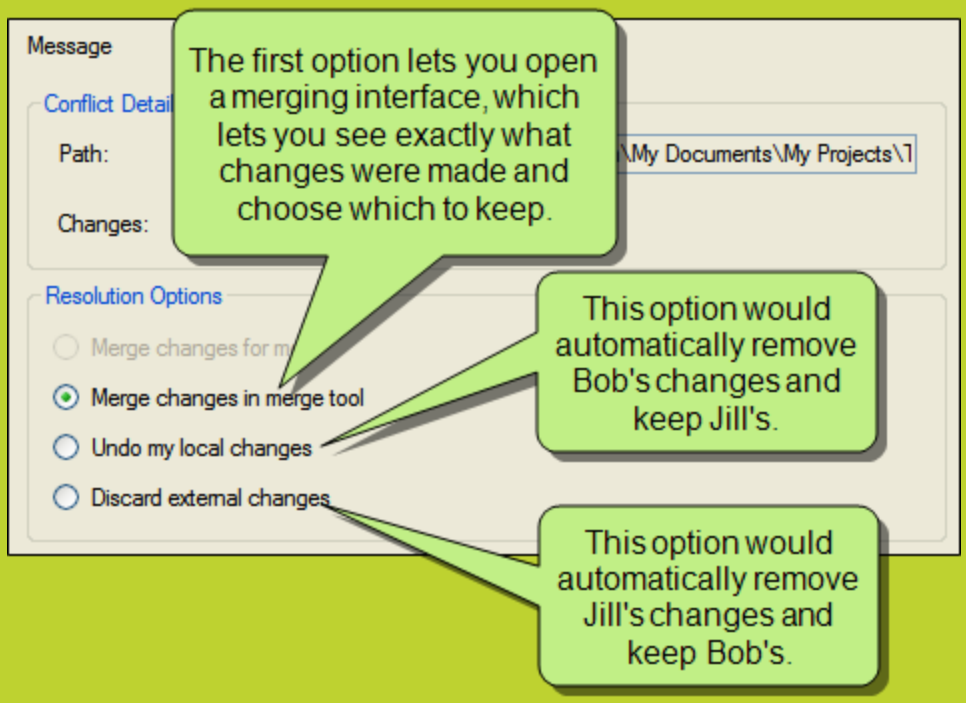
Bob finishes his changes and tries to check in the file. Before the file can be checked in, Bob is prompted with a dialog, indicating that changes from another author have already been checked in.



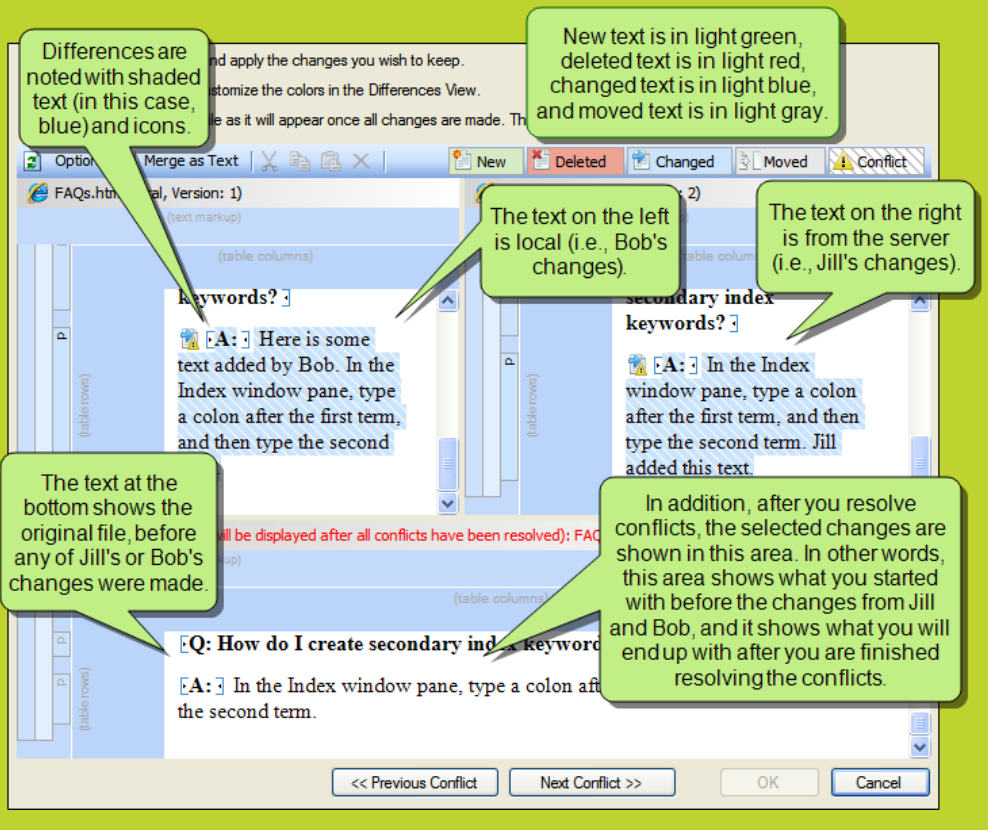
Bob selects **Auto Merge All**. However, he receives another message, stating that his changes conflict with those of Jill.



Bob clicks **OK**. Then in the Resolve Conflicts dialog he clicks **Resolve**. This opens the Resolve Version Conflict dialog. From this dialog, Bob can choose from a few different options.

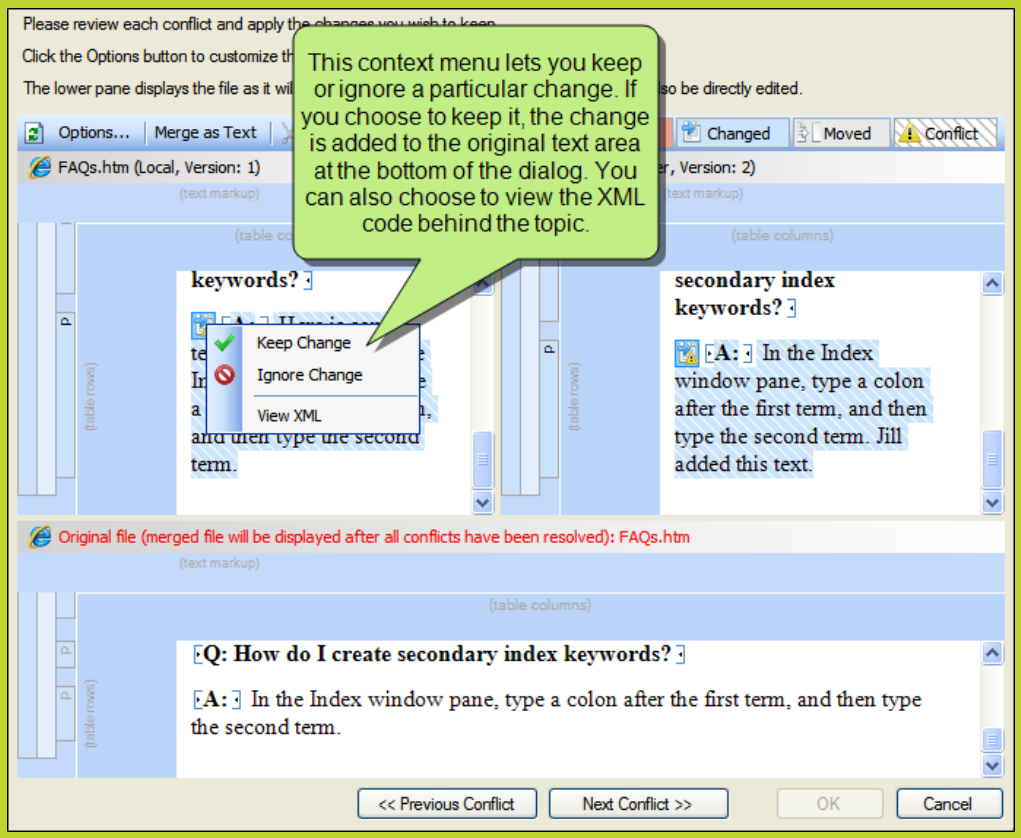


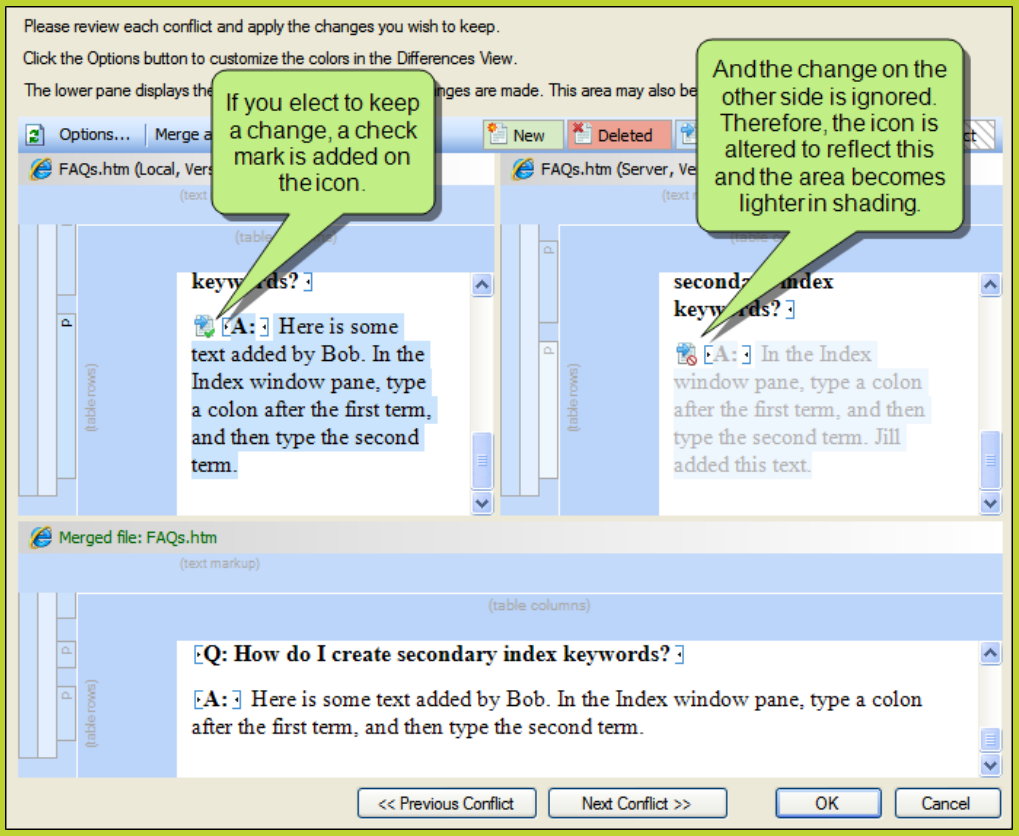
Bob selects **Merge changes in merge tool**. The Merge Changes dialog opens.



So what action can Bob take at this point for merging the file changes?

Bob can right-click on the icon next to the change on either the local or server side. This displays a context menu, which lets Bob either keep or ignore a particular change.

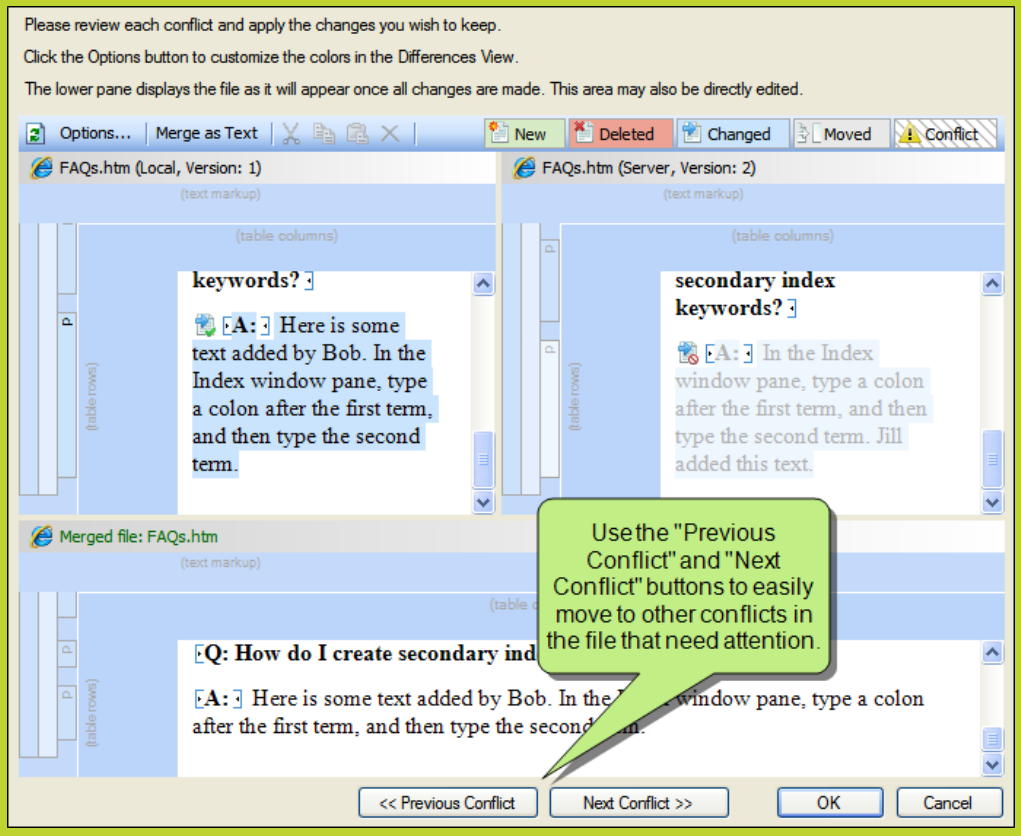




Another way of accomplishing the same thing is to left-click on an icon. When you do this, the change from one side is kept and the change from the other side is ignored. If you left-click the icon again, the results are toggled; the first change is ignored and the other is kept.

What if Bob wants to use his changes as well as Jill's, and possible tweak the paragraph a bit more? He can click in the area at the bottom of the dialog and simply type content.

When he is finished resolving the first conflict, Bob can use the "Previous Conflict" and "Next Conflict" buttons at the bottom of the dialog to work on other conflicts in the file.



After all conflicts have been resolved, Bob clicks **OK**. The merged topic is now checked in to source control.

Icons for File Differences

Following are descriptions of the various icons you might see in the Merge Changes dialog when merging files.

Icon	Description
	Changed Content—Conflict This indicates that the content has changed and a conflict exists. Therefore, user action is required. This icon appears in both the left and right panes of the Merge Changes dialog at the spot of the conflict. You must click on one of the icons to "Accept" that particular change.
	Conflict Change—Accepted This indicates that a change with a conflict has been accepted after clicking on . The bottom pane of the Merge Changes dialog updates to show the accepted content.
	Conflict Change—Not Accepted This indicates that a change with a conflict will not be accepted. When you accept a conflict change and that side of the Merge Changes dialog displays , the opposite pane where the conflict exists will display .
	Changed Tag—Conflict This indicates that the tag has changed and a conflict exists. Therefore, user action is required. This icon appears in both the left and right panes of the Merge Changes dialog at the spot of the conflict. You must click on one of the icons to "Accept" that particular change.
	Conflict Tag—Accepted This indicates that a change with a conflict in a tag has been accepted after clicking on . The bottom pane of the Merge Changes dialog updates to show the accepted content.
	Conflict Tag—Not Accepted This indicates that a change with a conflict in a tag will not be accepted. When you accept a conflict change and that side of the Merge Changes dialog displays , the opposite pane where the conflict exists will display .

Following are descriptions of conflict icons.

Icon	Description
	Changed Content—Conflict This indicates that the content has changed and a conflict exists. Therefore, user action is required. This icon appears in both the left and right panes of the Merge Changes dialog at the spot of the conflict. You must click on one of the icons to "Accept" that particular change.
	Conflict Change—Accepted This indicates that a change with a conflict has been accepted after clicking on . The bottom pane of the Merge Changes dialog updates to show the accepted content.

Icon	Description
	Conflict Change—Not Accepted This indicates that a change with a conflict will not be accepted. When you accept a conflict change and that side of the Merge Changes dialog displays , the opposite pane where the conflict exists will display .
	Changed Tag—Conflict This indicates that the tag has changed and a conflict exists. Therefore, user action is required. This icon appears in both the left and right panes of the Merge Changes dialog at the spot of the conflict. You must click on one of the icons to "Accept" that particular change.
	Conflict Tag—Accepted This indicates that a change with a conflict in a tag has been accepted after clicking on . The bottom pane of the Merge Changes dialog updates to show the accepted content.
	Conflict Tag—Not Accepted This indicates that a change with a conflict in a tag will not be accepted. When you accept a conflict change and that side of the Merge Changes dialog displays , the opposite pane where the conflict exists will display .

CHAPTER 8

Adding Files to Source Control

When working in a project that is connected to a source control application, there may be occasions when you have files in your local copy of the project that are not yet part of the source control copy. For example, when you add a new topic in your local copy of the project, that file will not be included in the source control copy of the project until you add it.

HOW TO ADD EXISTING FILES TO SOURCE CONTROL

1. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files in the window pane that you want to add, right-click, and select **Source Control>Add**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Add**.
   - **Menu** Select **File>Source Control>Add**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Add**.

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

   The Check In dialog opens. The selected files are listed with check boxes next to them.

2. (Optional) Enter an optional comment tied to the check-in. This enables you to keep an audit trail for a file. The comment can then be viewed from the History dialog, which can be accessed from the Source Control ribbon, the File menu, or the Source Control button .
3. (Optional) If you want to see all files with pending changes (rather than only those you selected), click .
4. Make sure to click the check box next to each file you want to check in so that it contains a check mark.
5. If you want to automatically check the files out of source control, select **Keep Checked Out**.
6. Click **Check In**.

CHAPTER 9

Requesting a User to Check In Files

If you discover that another user already has that file checked out, you can automatically send a request to that user to check in the file. This is a completely automated process involving instant messaging or email. Not only are the messages sent to the appropriate individuals automatically, but the mechanisms for checking in and checking out the necessary files are built in to those messages. This means that you and the other authors on your team do not need to hunt down the correct files manually.

How do you know if another user has a file checked out? Open the Pending Check-Ins window pane—select **Source Control>Pending Check-Ins** (if using the ribbon view) or **View>Pending Check-Ins** (if using the menu view). You can also opoen the File List window pane (**View>File List**). If you have files checked out, you will see the phrase "Checked Out" in the Status column, with your user name displayed in the User column. If another person has a particular file checked out, you will see that individual's user name in the User column, and the Status column will be blank. If this is the case, you can use the steps below to request that user to check in the file. See "Viewing Files That are Checked Out" on page 74.

Note: You cannot use this feature if you have disabled instant messaging.

HOW TO REQUEST A USER TO CHECK IN SOURCE CONTROL FILES

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Request Check In**.
   - **Ribbon** Select the **Source Control** ribbon. In the **Instant Messaging** section select **Request Check In**.
   - **Menu** Select **File>Source Control>Request Check In**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Request Check In**.

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

   The Send Request Check-In Message dialog opens. The individual who has the file(s) checked out is automatically selected in the dialog.
3. Select either **Send Instant Message** or **Send Email**.
4. (Optional) Enter a message to the other user.

   If you leave the **Message** field blank, Flare sends a default message.
5. Click **Send**. A popup indicates that the instant message or email has been sent to the other user.
6. Click **OK**.
7. If you are working from the Project Properties dialog, click **OK**.

   When the other user receives the instant message, that person can click a button in the message that automatically checks the file in.

   When the other user receives the email, it will contain an attachment. When that person opens the attachment, the file is automatically checked in.

   Afterwards, you will receive a message indicating that the file has been checked in, and you can click a button in that message to automatically check the file out.

CHAPTER

Requesting a User to View Files

If you want another author to review a file and provide comments, this can be done by performing a request to view files. Similar to the process for requesting check-ins, this feature uses instant messaging to streamline the workflow.

Note: You cannot use this feature if you have disabled instant messaging.

HOW TO REQUEST A USER TO VIEW SOURCE CONTROL FILES

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Request View File**.
   - **Ribbon** Select the **Source Control** ribbon. In the **Instant Messaging** section select **Request View File**.
   - **Menu** Select **File>Source Control>Request View File**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Request View File**.

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

   The View File Request dialog opens.
3. Select **Send Instant Message**.
4. (Optional) Enter a message to the other user.

   If you leave the **Message** field blank, Flare sends a default message.
5. In the **Recipients** section, select the individuals to whom you want to send the message. You can select multiple persons by holding the **SHIFT** or **CTRL** key and selecting a range of users or individuals users.
6. Click **Send**. A popup indicates that the instant message has been sent to the other user.
7. Click **OK**.

When other users receive your message, they can open the specified file by clicking a button in the message prompt.

CHAPTER

Additional Source Control Tasks

In addition to the many features already covered in this guide, there are many more tasks related to source control that you can perform in Flare.

This chapter discusses the following:

Disabling the Get Latest Prompt for Source Control 58
Enabling Source Control Status Checks 59
Rolling Back to an Earlier Version of a File 60
Setting Color Options for Project File Differences 62
Unbinding a Project from Source Control 63
Undeleting Source Control Files 64
Undoing a Check-Out of Source Control Files 66
Using an API to Integrate Source Control 67
Viewing Differences in Source Control Files 68
Viewing Files That are Checked Out 74
Viewing the History of Source Control Files 78

Disabling the Get Latest Prompt for Source Control

By default, when you open a project that is bound to source control, a message automatically asks if you wanted to get the latest version of files. However, you can disable this prompt in the Options dialog. Therefore, in the future when you open the project you will no longer see the message, and the project will open without replacing any local files with the latest ones from source control.

HOW TO DISABLE THE GET LATEST PROMPT FOR SOURCE CONTROL

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

   The Options dialog opens.
2. Select the **Source Control** tab.

   Note: This tab will not be visible if your project is not yet bound to source control. See "Binding a Project to Source Control" on page 21.
3. Click the check box **Do not prompt to get latest when opening source control bound projects**.
4. Click **OK**.

Enabling Source Control Status Checks

If you are using source control integration in Flare, you can check for frequent status changes automatically. You can specify the number of minutes and seconds when you want Flare to ping the source control repository and get status changes for files that have been checked out, checked in, moved, deleted, etc. The upside of this feature is that you can ensure that the source control status information is always up to date. The downside is that you may experience slower performance due to this constant communication over the network.

HOW TO ENABLE SOURCE CONTROL STATUS CHECKS

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

   The Options dialog opens.
2. Select the **Source Control** tab.

   Note: This tab will not be visible if your project is not yet bound to source control. See "Binding a Project to Source Control" on page 21.
3. Click the check box **Enable background status checks**. A check mark in the box indicates that the feature is enabled.
4. Enter the number of minutes and or seconds between each status update.
5. Click **OK**.

Note: If you elect to disable this feature (disabled is the default setting), you can manually check for status updates by refreshing the Pending Check-Ins window pane. See "Viewing Files That are Checked Out" on page 74.

Rolling Back to an Earlier Version of a File

One of the benefits of Flare's integrated source control is that you can view the history and differences for a particular file. You can view code and content differences between two source control versions of the same file. This is useful if you need to roll back to an earlier version of a file.

See "Introduction" on page 5, "Viewing the History of Source Control Files" on page 78, and "Viewing Differences in Source Control Files" on page 68.

E X A M P L E

Let's say that you have been working on a particular topic for a few days. Each day you check out the latest version of that topic file from source control, make your changes, and check the file back in to source control at the end of the day. At a certain point, you determine that you need to "roll back" to an earlier version of the file, using it to replace the latest version. Therefore, you use this feature to view the highlighted differences between the current version and an older version of the file. Once you have identified the older version that you want to use, you can perform a "get" of that version.

HOW TO ROLL BACK TO AN EARLIER VERSION OF A FILE

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>View History**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **View History**.
   - **Menu** Select **File>Source Control>View History**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **View History**.

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

   The History dialog opens.

3. (Optional) View the differences between two versions of the file. This may help you decide which version of the file you want to retrieve. (Another way is to look at the dates for each version in the History dialog.)

   To view the differences, follow these steps.

   a. Select the first file version from the list.
   b. Hold the **CTRL** key and select the second file version from the list.
   c. Click **Show Differences**.

      A window pane opens displaying the first version selected on the right side and the second version selected on the left side. The differences between the versions are highlighted.

      > **Note:** This split window pane opens only if you are using Microsoft Visual SourceSafe. If you are using Microsoft Team Foundation Server, the Show Differences Dialog, which is a simple text-based viewer, opens instead. However, if you install Team Explorer, a more robust window pane will open.

   d. When you are finished viewing the differences, close the window pane.
4. In the History dialog, select the version of the file to which you want to roll back.
5. Click **Get Selected Version**. That file is downloaded from source control and replaces the local copy of the file in your project.
6. In the History dialog, click **Close**.

Setting Color Options for Project File Differences

If you are using Flare's integrated source control features, you can view differences between files in various ways. One way is to view file differences between a local version of a Flare project and the source control version.

When viewing file differences between a local version of a Flare project and the source control version, you can select color options to display the files. Color coding makes it easier to discern where differences between files occur.

For more information see "Viewing Differences in Source Control Files" on page 68 and "Introduction" on page 5.

E X A M P L E

By default the files that are included only in your local copy are displayed as green in the Differences Editor, and the files that are included only in source control are displayed in red. You can use this dialog to change the local-only files to blue and the source control–only files to yellow.

HOW TO SET COLOR OPTIONS FOR PROJECT FILE DIFFERENCES

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Show Differences**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Show Differences**.
   - **Menu** Select **File>Source Control>Show Differences**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Show Differences**.

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

   The Differences Options dialog opens.
3. Click **OK**.The Differences Editor opens.
4. In the local toolbar of the Differences Editor, click **Options**. The Folder Differences dialog opens.
5. Change the text or background color for any of the difference types. To do this, simply click in the appropriate **Text** or **Background** cell and select **Pick Color**. In the Color Picker dialog, choose the new color.
6. Click **OK**.

Unbinding a Project from Source Control

Use the following steps if you have previously connected your Flare project to source control but now wish to unbind it (i.e., remove the connection).

HOW TO UNBIND A PROJECT FROM SOURCE CONTROL

1. Open the 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 **Properties** section select **Project Properties**.
   - **Menu** Select **Project>Project Properties**.
   - **Project Toolbar** Click Project Properties...

     To open the Project toolbar from the menu view, select **View>Toolbars>Project**.

   The Project Properties dialog opens.
3. Select the **Source Control** tab.
4. Click **Unbind Project**.
5. Click **OK**.

Undeleting Source Control Files

If you have a file marked as "Deleted" but then want to add it to source control, you can undelete the file, which then allows you to do this. You might encounter this situation if you rename a file and then create a new file with that name, or if you rename a file to the name of a previously deleted file.

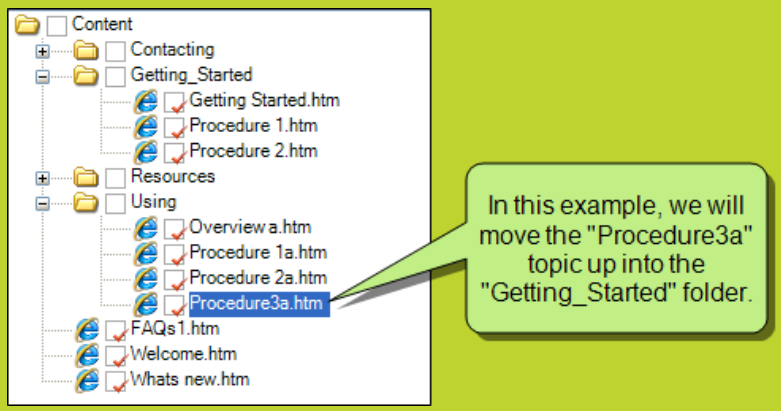
HOW TO UNDELETE A SOURCE CONTROL FILE

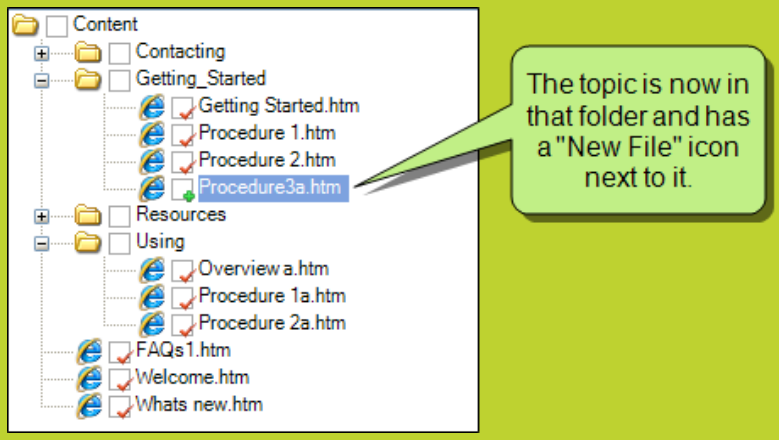
1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Undelete**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Undelete**.
   - **Menu** Select **File>Source Control>Undelete**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Undelete**.

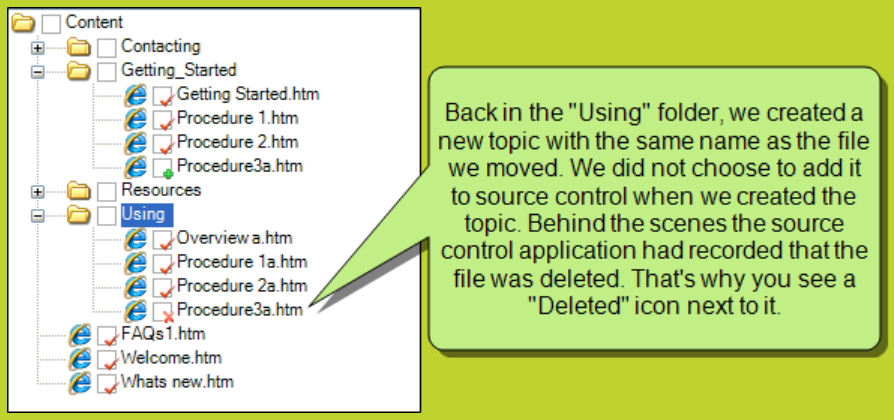
     To open the Standard toolbar from the menu view, you can select **View>Toolbars>Standard**.

E X A M P L E

Let's say you open the Content Explorer and drag a file from one folder to another. Then you create a topic with the same name in the original folder. That new topic has a "Deleted" status and must be undeleted before it is added to source control.







In order to add the new file to source control, you need to remove the "Deleted" status. Therefore, you right-click on the file, select **Source Control**, and choose **Undelete**.

Undoing a Check-Out of Source Control Files

If you have files checked out from source control but do not want them checked out anymore, you can use the "Undo Check Out" option instead of checking in the files.

HOW TO UNDO A CHECK-OUT OF SOURCE CONTROL FILES

1. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files in the window pane that you want to check out, right-click, and select **Source Control>Undo Check Out** (for selected files) or **Source Control>Project>Undo Check Out All** (for all files in the project).
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Undo Check Out** (for selected files). Alternatively, in the **Project** section you can click **Undo Check Out All** (for all files in the project).
   - **Menu** Select **File>Source Control>Undo Check Out** (for selected files) or **File>Source Control>Project>Undo Check Out** (for all files in the project).
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Undo Check Out** (for selected files) or **Project>Undo Check Out All** (for all files in the project).

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

   The Undo Check Out dialog opens. The selected files are listed with check boxes next to them.
2. (Optional) If you want to see all files with pending changes (rather than only those you selected), click .
3. Make sure to click the check box next to each appropriate file so that it contains a check mark.
4. Click **Undo Check Out**.

Using an API to Integrate Source Control

If you are working with a source control application other than those directly supported in Flare (i.e., Microsoft Visual SourceSafe, Microsoft Team Foundation Server, Perforce, Apache Subversion), you can use the Microsoft Source Code Control API (SCC API) to integrate that application with your Flare project.

The SCC API is an interface specification defined by Microsoft that defines the interface between an application (in this case, Flare) and a source control provider (e.g., CVS, IBM Rational ClearCase), providing hooks for a number of common source control operations (such as check out, check in, get latest, and so on).

Many source control providers adhere to this interface as a standard for source control integration, including (but not limited to) the following.

- CVS
- IBM Rational ClearCase
- CS-RCS
- Team Coherence
- QVCS-Pro

If you are using a source control package that supports the Microsoft SCC API, install the plug-in that is provided by that tool and follow any included instructions. This plug-in from your source control provider should be installed on the client machine (i.e., the computer that is running Flare). Once you have done this correctly, you should be able to select that plug-in from the Flare interface whenever you are asked to select a source control provider (e.g., when binding a project to source control).

Note: It is not mandatory that you use the SCC API to use a source control application for your Flare project files. It is only necessary if you want to take advantage of the source control integration features inside Flare. Otherwise, you can manually add your Flare project files to any source control application.

Viewing Differences in Source Control Files

One of the benefits of Flare's integrated source control is that you can view the history and differences for a particular file. You can view differences between files in the following ways.

- **Two Versions of Same Source Control File (History/Roll Back)** You can view code and content differences between two source control versions of the same file. This is useful if you need to roll back to an earlier version of a file. See "Rolling Back to an Earlier Version of a File" on page 60 and "Getting the Latest Version" on page 29.

  EXAMPLE

  Let's say that you have been working on a particular topic for a few days. Each day you check out the latest version of that topic file from source control, make your changes, and check the file back in to source control at the end of the day. At a certain point, you determine that you need to "roll back" to an earlier version of the file, using it to replace the latest version. Therefore, you use this feature to view the highlighted differences between the current version and an older version of the file. Once you have identified the older version that you want to use, you can perform a "get" of that version.

- **Two Different Source Control Files** You can view code and content differences between the source control version of two different files.

  EXAMPLE

  Let's say that you have two topics that are quite similar in some aspects, but different in others. Rather than trying to scan the two topics with your eyes to determine the differences, you can select the two files in the Flare interface and use this feature. Flare will retrieve the source control versions of the two topic files and display the content and code of each side-by-side, highlighting the areas where differences occur.

- **Local Versus Source Control Version of a File** You can view code and content differences between the local version of a file and the source control version of that file.

  EXAMPLE

  Let's say that you check out a procedure topic from source control and then add some lines of text to your local copy of that topic file. You save your changes. Later that day, you want to revisit the new content that you wrote, but you cannot remember exactly which lines of text you added. Therefore, you use this feature to highlight the text differences between your local checked-out version of the file and the version stored in

the source control application. The new lines of text are highlighted on the side displaying the local version of the file.

» **Local Versus Source Control Version of All Files in a Folder** You can view file differences between the local version of the files in a folder and the source control version. Most likely, you will use this to view all of the difference between the local files and source control files in either your Content Explorer or Project Organizer.

E X A M P L E

Let's say that you have checked out all of the files for a Flare project. During the course of the day, you add several new topics to the project, but you forget to add them to source control during the creation process. At the end of the day, you check in all of the files in the project to source control. Afterwards, you realize that you forgot to add the new files to source control. Therefore, you use this feature to see the file-level differences between your local copy of the Content Explorer root folder and the source control copy. The differences are color coded, so you can easily identify the files in question. (By default, the files that are included only in your local copy are green, and the files that are included only in source control are red.) You can then right-click on the files that were added only to the local copy, and you can select to add them to source control.

HOW TO VIEW DIFFERENCES BETWEEN TWO VERSIONS OF THE SAME SOURCE CONTROL FILE

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>View History**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **View History**.
   - **Menu** Select **File>Source Control>View History**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **View History**.

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

   The History dialog opens.
3. From the list, select the first file version that you want to compare.
4. Hold the **CTRL** key and select the second file version from the list.
5. Select **Show Differences**.

   A window pane opens displaying the first version selected on the right side and the second version selected on the left side. The differences between the versions are highlighted.

   > **Note:** This split window pane opens only if you are using Microsoft Visual SourceSafe. If you are using Microsoft Team Foundation Server, the Show Differences Dialog, which is a simple text-based viewer, opens instead. However, if you install Team Explorer, a more robust window pane will open.

6. When you are finished viewing the differences, close the window.
7. In the History dialog, click **Close**.

HOW TO VIEW DIFFERENCES BETWEEN TWO DIFFERENT SOURCE CONTROL FILES

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **View** ribbon. In the **Explorer** section select **File List**.
   - **Menu** Select **View>File List**.
   - **Keyboard Shortcut** Press **CTRL+SHIFT+J**.

   The File List window pane opens.
2. (Optional) In the local toolbar, click the down arrow in the **Filter** field and select the types of files that you want to see in the window pane.
3. From the list, select the first file version that you want to compare.
4. Hold the **CTRL** key and select the second file version from the list.
5. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Show Differences**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Show Differences**.
   - **Menu** Select **File>Source Control>Show Differences**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Show Differences**.

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

   A window pane opens displaying the first version selected on the right side and the second version selected on the left side. The differences between the versions are highlighted.

   > **Note:** This split window pane opens only if you are using Microsoft Visual SourceSafe. If you are using Microsoft Team Foundation Server, the Show Differences Dialog, which is a simple text-based viewer, opens instead. However, if you install Team Explorer, a more robust window pane will open.

6. When you are finished viewing the differences, close the window.

HOW TO VIEW DIFFERENCES BETWEEN THE LOCAL VERSION OF A FILE AND THE SOURCE CONTROL VERSION

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>Show Differences**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Show Differences**.
   - **Menu** Select **File>Source Control>Show Differences**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Show Differences**.

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

   If the files are identical, a message indicates it.

   If the files are not identical, a window pane opens displaying the local version on the left side and the source control version selected on the right side. The differences between the versions are highlighted.

   > **Note:** This split window pane opens only if you are using Microsoft Visual SourceSafe. If you are using Microsoft Team Foundation Server, the Show Differences Dialog, which is a simple text-based viewer, opens instead. However, if you install Team Explorer, a more robust window pane will open.

3. When you are finished viewing the differences, close the window.

HOW TO VIEW DIFFERENCES BETWEEN THE LOCAL VERSION OF ALL FILES IN A FOLDER AND THE SOURCE CONTROL VERSION

1. In the Content Explorer or Project Organizer, select the relevant folder.
2. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **Show Differences**.
   - **Menu** Select **File>Source Control>Show Differences**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **Show Differences**.

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

   The Differences Options dialog opens.
3. (Optional) Use this dialog to specify the type of information that you want to see in the Differences Editor. (You can also choose these options from the local toolbar of the Differences Editor.)
   - **Show files that are only in the Left folder** Displays the files on the left side of the Differences Editor. The left side is used to show the local copies of your project files.
   - **Show files that are only in the Right folder** Displays the files on the right side of the Differences Editor. The right side is used to show the source control copies of your project files.
   - **Show files that are different in both folders** Displays the files where differences occur between the local copy and source control copy of the project. For example, the left side might display files that you have created in your local copy but have not yet been added to source control.
   - **Show files that are the same in both folders** Displays the files that are the same in the local copy as they are in the source control copy.
   - **Recursive** Displays files recursively. In other words, if you have files contained within folders, selecting this button will ensure that you see all of the files, not just the folders.
4. Click **OK**.

   The Differences Editor opens.
5. (Optional) In the Differences Editor, use the buttons in the local toolbar to customize the information shown in the editor.

Viewing Files That are Checked Out

You can use the Pending Check-Ins window pane to view all of the files that you have checked out and need to be checked in, as well as files that others have checked out. You can use the File List window pane to view files that are checked out by other users.

HOW TO VIEW FILES THAT YOU AND OTHERS HAVE CHECKED OUT—PENDING CHECK-INS WINDOW PANE

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **Source Control** ribbon. In the **Changes** section select **Pending Check-Ins**.
   - **Menu** Select **View>Pending Check-Ins**.

   The Pending Check-Ins window pane opens.
2. (Optional) You can use the **Filter** field to limit the files that are displayed.
   - **All Files** Displays all files.
   - **Topic Files** Displays only the topic (HTM and HTML) files.
   - **MasterPage Files** Displays only the master page (FLMSP) files.
   - **Page Layout Files** Displays only the page layout (FLPGL) files.
   - **Snippet Files** Displays only the snippet (FLSNP) files.
   - **Stylesheet Files** Displays only the stylesheet (CSS) files.
   - **Image Files** Displays only the image files (BMP, EMF, EXPS, GIF, HDP, JPG, JPEG, PNG, SWF, TIF, TIFF, WDP, WMF, XAML, XPS) and media files (SWF).
   - **Multimedia Files** Displays only multimedia (audio and video) files.
   - **GIF Files** Displays only GIF files.
   - **JPEG Files** Displays only JPG and JPEG files.
   - **PNG Files** Displays only PNG files.
   - **Flash Movie Files** Displays only SWF files.
3. (Optional) You can use the following toggle buttons in the local toolbar to limit the files that are displayed.

	This filters the Pending Check-Ins window pane to show or not show files that other users have checked out.
	This filters the Pending Check-Ins window pane to show or not show files that are out of date.

4. Take note of the **Status** and **User** columns. (You may need to use the scroll bar to view these columns.)

   - **Status** Displays the status of the file, such as whether you have it checked out. This status is blank if another user has the file checked out through source control.

     - **Checked Out** This indicates that the file has been checked out from source control. You can check in the file when you are ready (if you are the user who has checked it out).
     - **Pending Add** This indicates that you have a file in your project but have not yet added it to the integrated source control application. This might occur, for example, if you create a new topic and do not add the file to source control during the topic creation process. To resolve this, simply right-click on the file and select **Source Control>Add**.
     - **Out of Date** This indicates that the file is not current (i.e., the local copy of the file is older than the source control copy). This might happen, for example, if another user checks out the file, makes changes to it, and checks it back in to source control. If this occurs, you can check the file out or get the latest version of the file from source control.

     Note: You can click the refresh button in the local toolbar to make sure you have the most recent status for each file. Another option is that you can use a feature to automatically ping the source control repository periodically, thus refreshing this information frequently. However, you may experience slower performance with this automatic status update option set. See "Enabling Source Control Status Checks" on page 59.

   - **User** Displays the user name. If you see the name of another user in this column, it means that the file is checked out by that user through source control. If necessary, you can request that the other user check in the file so that you can work on it. See "Requesting a User to Check In Files" on page 53.

HOW TO VIEW FILES THAT ARE CHECKED OUT BY OTHERS—FILE LIST WINDOW PANE

1. Do one of the following, depending on the part of the user interface you are using:
   - **Ribbon** Select the **View** ribbon. In the **Explorer** section select **File List**.
   - **Menu** Select **View>File List**.
   - **Keyboard Shortcut** Press **CTRL+SHIFT+J**.

   The File List window pane opens.
2. (Optional) You can use the **Filter** field to limit the files that are displayed.
   - **All Files** Displays all files.
   - **Topic Files** Displays only the topic (HTM and HTML) files.
   - **MasterPage Files** Displays only the master page (FLMSP) files.
   - **Page Layout Files** Displays only the page layout (FLPGL) files.
   - **Snippet Files** Displays only the snippet (FLSNP) files.
   - **Stylesheet Files** Displays only the stylesheet (CSS) files.
   - **Image Files** Displays only the image files (BMP, EMF, EXPS, GIF, HDP, JPG, JPEG, PNG, SWF, TIF, TIFF, WDP, WMF, XAML, XPS) and media files (SWF).
   - **Multimedia Files** Displays only multimedia (audio and video) files.
   - **GIF Files** Displays only GIF files.
   - **JPEG Files** Displays only JPG and JPEG files.
   - **PNG Files** Displays only PNG files.
   - **Flash Movie Files** Displays only SWF files.
3. Take note of the **Status** and **User** columns. (You may need to use the scroll bar to view these columns.)
   - **Status** Displays the status of the file, such as whether you have it checked out. This status is blank if another user has the file checked out through source control.

> **Note:** You can click the refresh button in the local toolbar to make sure you have the most recent status for each file. Another option is that you can use a feature to automatically ping the source control repository periodically, thus refreshing this information frequently. However, you may experience slower performance with this automatic status update option set. See "Enabling Source Control Status Checks" on page 59.

» **User** Displays the user name. If you see the name of another user in this column, it means that the file is checked out by that user through source control. If necessary, you can request that the other user check in the file so that you can work on it. See "Requesting a User to Check In Files" on page 53.

Viewing the History of Source Control Files

One of the benefits of Flare's integrated source control is that you can view the history for a particular file, including who checked in the file and when it was checked in. You can also view differences between different versions of the file and roll back to an older version if necessary. For more information see "Introduction" on page 5, "Viewing Differences in Source Control Files" on page 68, and "Rolling Back to an Earlier Version of a File" on page 60.

HOW TO VIEW THE HISTORY OF A SOURCE CONTROL FILE

1. In one of the window panes (e.g., Content Explorer, Project Organizer, File List window pane, Pending Check-Ins window pane), select the relevant files(s).
2. Do one of the following, depending on the part of the user interface you are using:
   - **Pending Check-Ins Window Pane** From the **Source Control** ribbon or **View** menu, open the Pending Check-Ins window pane. Select the files, right-click, and select **Source Control>View History**.
   - **Ribbon** Select the **Source Control** ribbon. In the **File** section select **View History**.
   - **Menu** Select **File>Source Control>View History**.
   - **Standard Toolbar** Click the down arrow next to the **Source Control** button and select **View History**.

     To open the Standard toolbar from the menu view, you can select **View>Toolbars>Standard**.
3. The History dialog opens. The following are explanations of the different parts of this dialog.
   - **Version** Displays a number for each version of the file. The lower the number, the older the version. The higher the number, the more recent the version.
   - **Users** Displays the name of the user who has been working on the file.
   - **Date** Displays the date and time when the action has occurred.
   - **Action** Displays the action that has taken place for the file (e.g., check in).
   - **Comment** Displays the comment (if any) associated with the file. A comment can be added to a file when you check that file in to source control. This enables you to maintain an audit trail for the file's history.
   - **Get Selected Version** Performs a "get" of a particular version of a file, thus rolling back to that version of the file. The local version of the file is replaced with the source control version that you selected. See "Rolling Back to an Earlier Version of a File" on page 60 and "Getting the Latest Version" on page 29.
   - **Show Differences** Opens a dialog that lets you view the differences between two versions of a file. If you select one row in the History dialog and view the differences, you will see the content differences between the version that you selected and the version of the file in your local copy of the Flare project. If you select two files in the History dialog (by holding down the CTRL key) and view the differences, you will see the content differences between those two versions of the file.
4. In the History dialog, click **Close**.

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