



MADCAP LINGO 11 r3

What's New

Copyright © 2023 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
9171 Towne Center Drive, Suite 335
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 1	
Introduction	5
CHAPTER 2	
Command Line Activation	7
How to Use Command Line Activation	8
CHAPTER 3	
Source Control Enhancements	11
New Shortcut Options in Status Bar	12
Branch Management Dialog Enhancements	16
Bind Detection, Disabling Providers, and Unbinding Providers	20
Ignore File for Git	28
CHAPTER 4	
Start Page Redesign	32
Page Options on Left	33
License Expiration Banner	34
Responsive to Window Size	35
APPENDIX	
PDFs	36

Cheat Sheets	36
User Guides	36

Introduction

Following are the new features available in Lingo 11 r3.

For more information about each feature discussed in this manual, open the online Help and refer to the "What's New" topic. Links are provided in some feature descriptions, taking you to topics that contain additional information and steps.



NOTE These are all features that were added to MadCap Flare in the past and have been inherited by this new release of Lingo.



"Command Line Activation" on page 7

- Activate (or deactivate) Lingo from the command line
- Helpful for companies using imaged machines and not able to activate Lingo from the user interface
- Available only for customers with enterprise floating licenses



"Source Control Enhancements" on page 11

- New shortcut options in status bar (lower-right)
- Branch Management dialog enhancements
- Bind detection, disabling providers, and unbinding providers
- Add or edit ignore file for Git




"Start Page Redesign" on page 32

- Options on left allow for more space to work
- New banners display expiration dates for different license types
- Links to take action on expiring license
- Responsive to window size

Command Line Activation

Lingo allows you to activate (or deactivate) the application from the command line. This can be helpful for companies using imaged machines and are not able to activate Lingo from the user interface.

 **IMPORTANT** This feature is available only for customers with enterprise floating licenses.

I How to Use Command Line Activation

1. Open a text editor, such as Notepad++, create a new file named "license.json." Save the file at the root level of the Lingo.app folder in Program Files where you installed Lingo.

☆ **EXAMPLE** If you installed Lingo at C:\Program Files (**this is just an example; your files might be installed in a different folder, such as Program Files x86**), you would save the JSON file like this:

```
C:\Program Files\MadCap Software\MadCap Lingo  
11\Lingo.app\license.json
```

2. Within the JSON file, enter the following. Have your IT department help you complete the file's parameters if necessary.

Parameters

- **LicenseKey (Required)** The purchased license key, which must be an enterprise floating key. You should have this key in the email that you received from MadCap Software when you purchased Lingo.
- **EmailAddress (Required)** The email address to use with activation.
- **FirstName (Optional)** The first name of the person activating.
- **LastName (Optional)** The last name of the person activating
- **UseProxy (Optional)** A true or false flag to indicate if you're using proxy settings.
- **Proxy (Optional)** Proxy settings to use, if needed, to access the Internet. All settings are required if the proxy is set.
 - **Address** The proxy address.
 - **Port** The proxy port.
 - **UseAuthentication** A true or false flag to indicate if you are using authentication with proxy.
 - **Username** The user account for proxy.
 - **Password** The user password for proxy.

Format

```
{
  "LicenseKey": "xxxx-xxxx-xxxx-ec712-448b8-9e11b-a18b3-4cedb", //required
  "EmailAddress": "email@me.com", //required
  "FirstName": "MadCap", //optional
  "LastName": "Simon", //optional
  "UseProxy": false, //optional
  "Proxy": {
    "Address": "",
    "Port": 0,
    "UseAuthentication": false,
    "Username": "",
    "Password": ""
  } //optional
}
```


3. After the completed JSON file is saved to the Lingo.app folder, open the command prompt on your computer.
4. Navigate to the location where the JSON file is saved.

☆ **EXAMPLE** Type the following in the command prompt and press **ENTER**:

```
cd\Program Files\MadCap Software\MadCap Lingo 11\Lingo.app
```

5. Type any of the following commands and press **ENTER**:
 - lingo -activate
 - lingo -deactivate

☆ **EXAMPLE** If you want to activate Lingo to work in it:



```
C:\Program Files\MadCap Software\MadCap Lingo 11\Lingo.app>lingo -activate
```

Source Control Enhancements

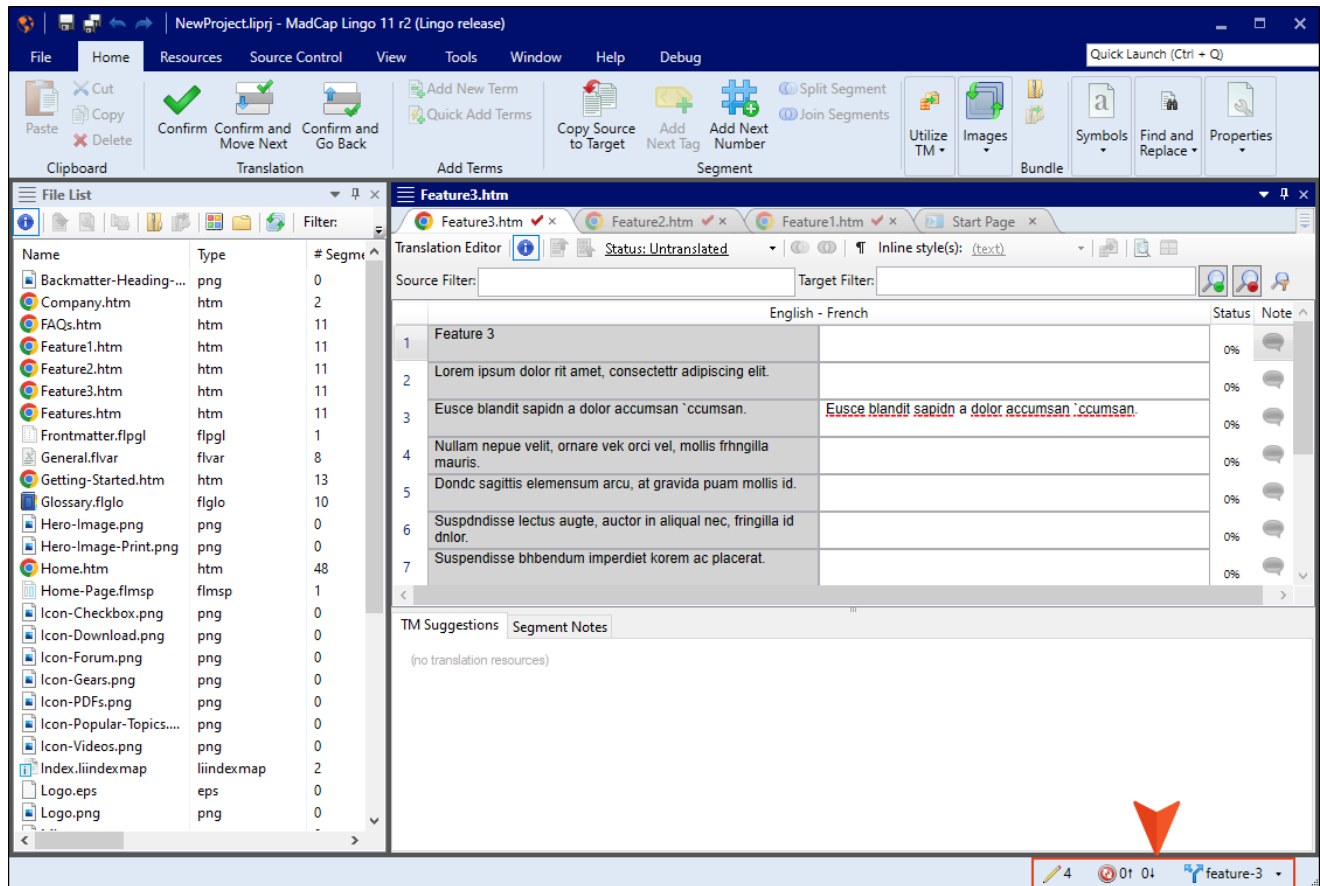
Various enhancements have been made for projects that are bound to source control, especially Git.

This chapter discusses the following:

New Shortcut Options in Status Bar	12
Branch Management Dialog Enhancements	16
Bind Detection, Disabling Providers, and Unbinding Providers	20
Ignore File for Git	28

New Shortcut Options in Status Bar

The following options are now visible in the status bar in the lower-right corner of Lingo when the project is bound to Git. These options make it easier to know which branch is active, as well as whether changes need to be committed or synchronized. You can also more quickly take actions on branches when necessary.



NOTE If you do not see this option, make sure **View > Status Bar** is enabled.

Changes to be Committed

If you see a number next to this button, it indicates how many files with changes need to be committed.



NOTE A number might not always immediately be shown next to button. Clicking somewhere in the interface helps it to populate.

When you click the button, the Source Control Explorer opens. Select the **Pending Changes** view, which you can use to add a comment and commit the changes.

The screenshot shows the Source Control Explorer interface with the 'Pending Changes' view selected. The interface includes a text box for a comment, a 'Commit Included' button, and a tree view of files. A red box highlights four files: Feature1.htm.xlf, Feature2.htm.xlf, Feature3.htm.xlf, and FAQs.htm.xlf. Callouts provide instructions: 'Click here to commit the changes below.' points to the 'Commit Included' button; 'You are required to add a comment when you commit changes to files.' points to the comment text box; 'In this example, there are four files that have been changed and will be committed.' points to the 'Included Changes (4)' section; and 'Here are the four files.' points to the highlighted files.

Commits to be Synchronized

If you see a number next to the up arrow on this button, it indicates that you have commits locally that need to be pushed to the remote repository.

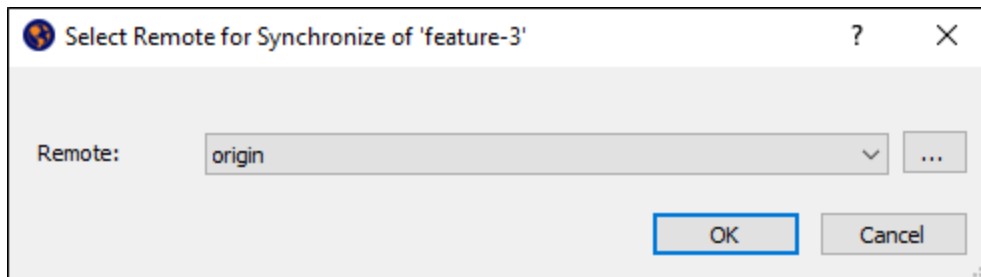


If you see a number next to the down arrow on this button, it indicates that there are pending remote commits that you still need to pull to your local repository. A number displays here only after you do a fetch (i.e., "git fetch"). This is a command that you cannot perform from the Lingo interface, but you can do it using another tool such as Git Bash. Another way for a number to display next to the down arrow is if you first do a pull on one branch, then switch to another branch that has pending remote commits.



NOTE After committing changes, you might need to click somewhere in the interface before numbers populate next to this button.

When you click the button, Lingo opens a dialog that lets you choose the remote repository. Lingo then synchronizes the changes, first pulling any commits from the remote repository (whether they include conflicting changes or not), and then pushing local commits up to the server.

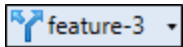


If there are conflicts, the Resolve Conflicts dialog opens first, which you can use to select the changes from either repository or to merge the changes.

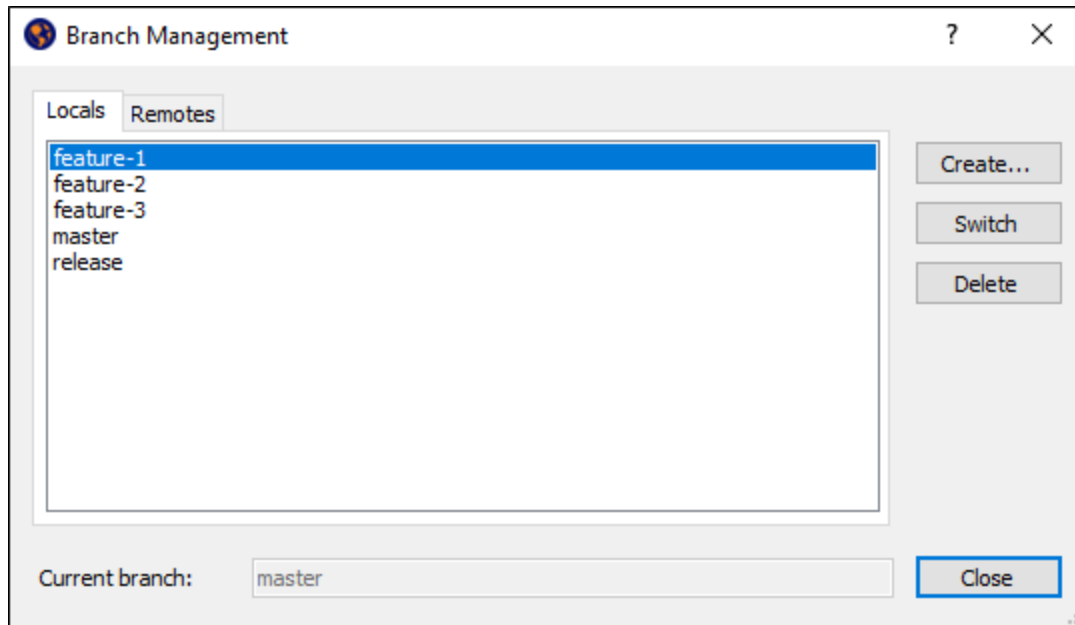
Once you are done resolving the conflicts, you need to try to synchronize the changes again.

Current Branch

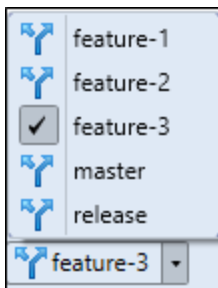
The option on the right displays the branch that is active.



If you click the button displaying the name, the Branch Management dialog opens, letting you work with branches (e.g., create, switch, delete).



If you click the drop-down next to the button, you can see all of your other branches and select one to quickly switch to it.




I Branch Management Dialog Enhancements

The Branch Management dialog now lets you see both local and remote Git branches. This means you can easily get remote branches that you don't yet have locally. Also, you can delete a branch either locally or remotely, or both.

How to Get a Remote Branch

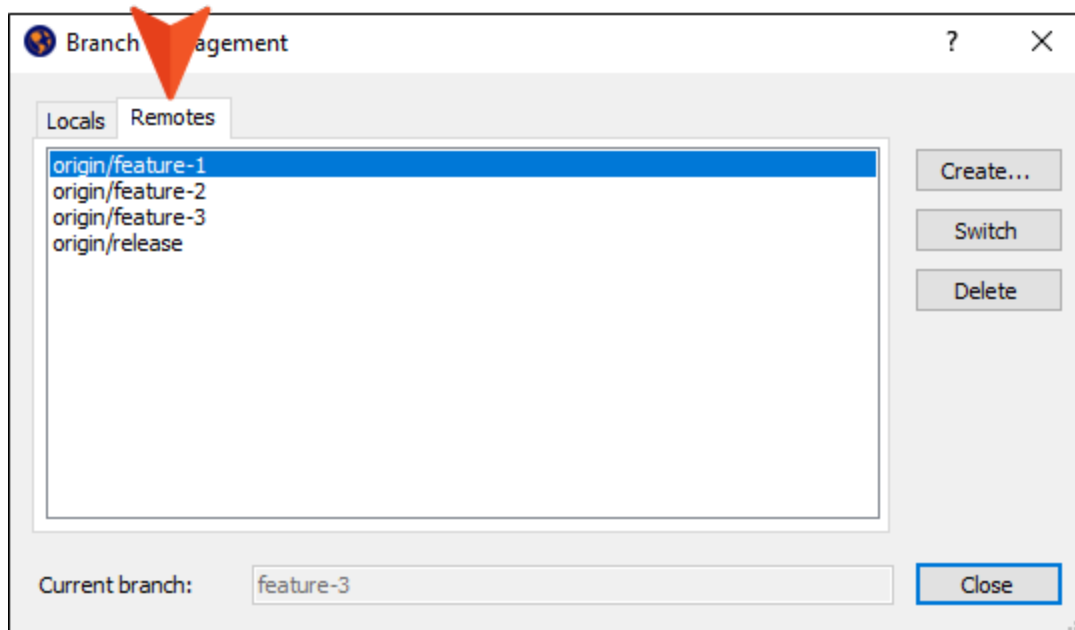
1. With any branch selected as the active one, do a pull. This makes all remote branches available for selection in the Lingo interface, even if you haven't yet added a remote branch locally.
2. Open the Branch Management dialog in one of the following ways:

- **Status Bar** In the lower-right of Lingo, click the name of the active branch.

 **NOTE** If you do not see this option, make sure **View > Status Bar** is enabled.


- **Ribbon** Select **Source Control > Branch** (the face of the button, not the drop-down).
- **Right-Click** If you have the Pending Changes or File List window pane open, right-click any file and select **Source Control > Project > Branch**.

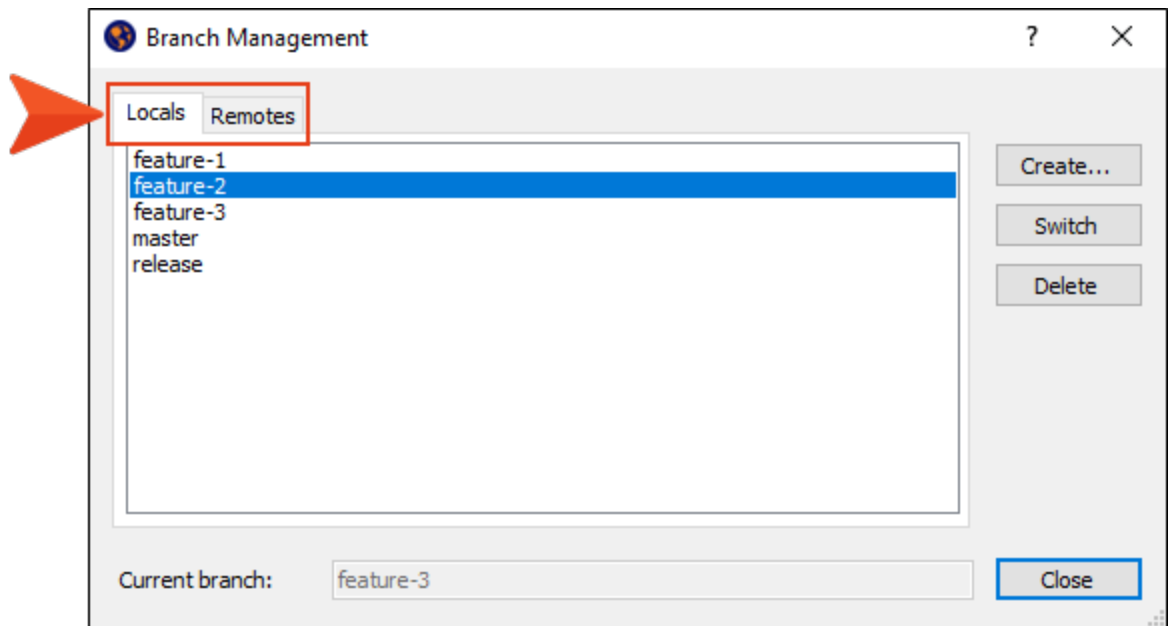
3. Select the **Remotes** tab.



4. Select the branch you want to get (i.e., check out) from the remote repository, and click **Switch**. This adds the branch to the Locals tab, and it also makes that new local branch the active one in the Lingo interface.
5. Close the Branch Management dialog.

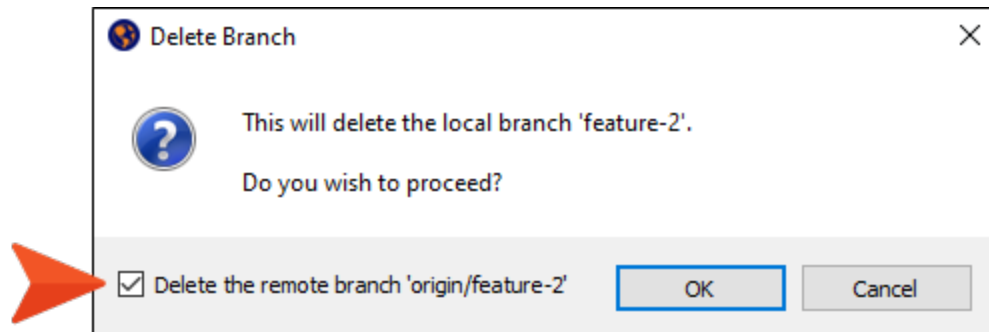
How to Delete Branches

1. Make sure the branch you want to delete is not the active branch.
 2. Open the Branch Management dialog in one of the following ways:
 - **Status Bar** In the lower-right of Lingo, click the name of the active branch.
-  **NOTE** If you do not see this option, make sure **View > Status Bar** is enabled.
- **Ribbon** Select **Source Control > Branch** (the face of the button, not the drop-down).
 - **Right-Click** If you have the Pending Changes or File List window pane open, right-click any file and select **Source Control > Project > Branch**.
 3. Select the **Locals** or the **Remotes** tab, depending on which branch you want to delete. If you want to delete a branch both locally and remotely, select either tab.

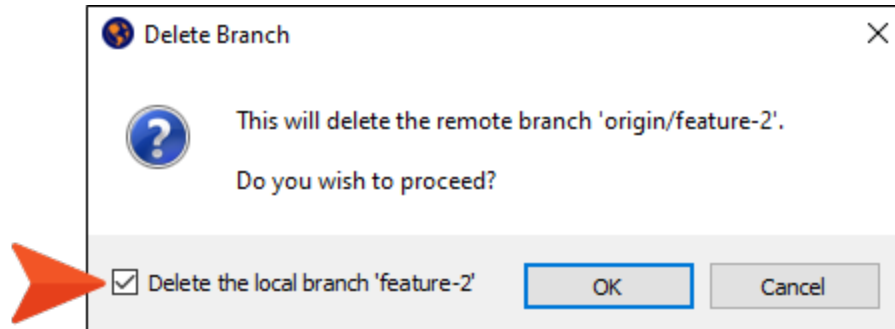


4. Select the branch you want to remove and click **Delete**.

5. If you selected a local branch, you are asked if you also want to delete the remote branch (if one exists). Click the check box if you do; otherwise, leave it disabled.



- If you selected a remote branch, you are asked if you also want to delete the local branch (if one exists). Click the check box if you do; otherwise, leave it disabled.



6. Click **OK**.
7. Close the Branch Management dialog.

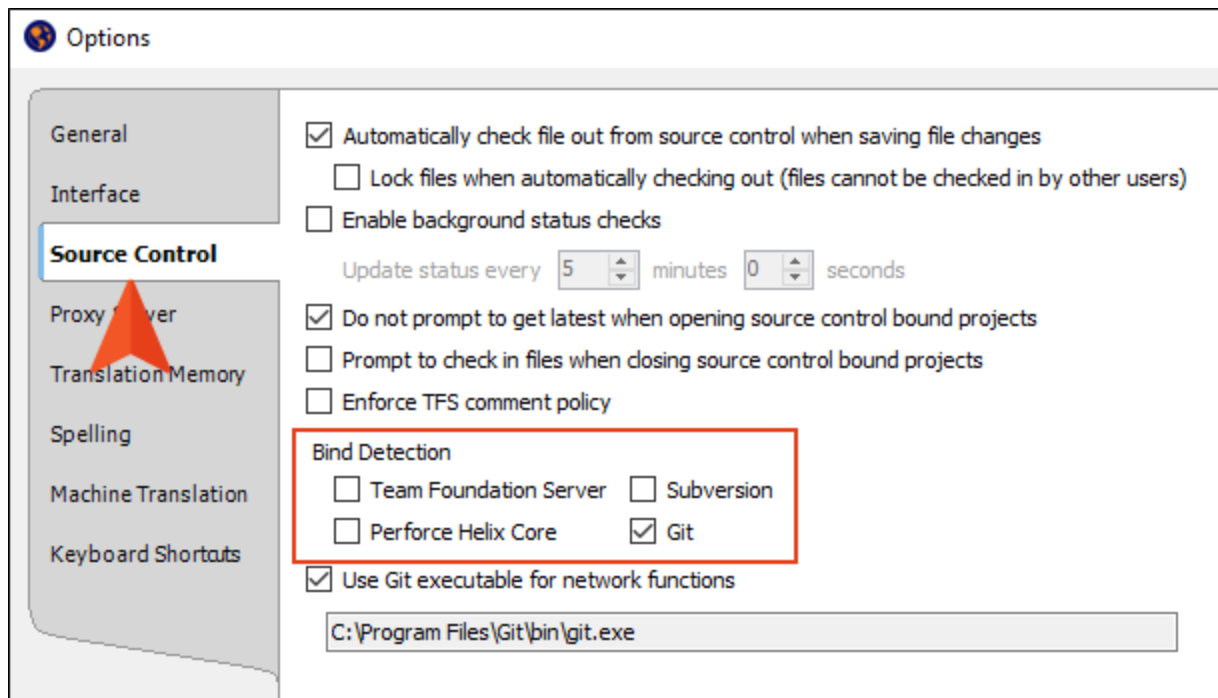
Bind Detection, Disabling Providers, and Unbinding Providers

Lingo has options for bind detection, disabling providers, and unbinding providers. Although these are separate features, they are all somewhat related. In fact, the bind detection option for Git can be used to automatically disable the provider for all your projects.

The new functionality in this version of Lingo revolves mostly around the ability to disable providers. However, information about bind detection and unbinding providers is also added below. This might help you better understand the relationships between these features.


Bind Detection


Lingo's bind detection settings are found on the Source Control tab of the Options dialog.



Bind detection scans your project when you load it to see if the project has been previously bound to source control. If a binding is detected, you then have the option of applying the binding and committing the project to source control. Depending on the provider you are using, Lingo may search the file system and its artifacts, as well as contact and query servers, to find potential source control bindings.

When you open a Lingo project that hasn't been bound to source control before, the bind detection option is disabled for Perforce Helix Core, Subversion, and Team Foundation Server. It is enabled by default for Git. If you bind a project to source control using the Lingo interface, the option is then automatically enabled.

 **NOTE** If you bind a project to Git using the Lingo interface, the detection will automatically happen behind the scenes. If you use a tool outside of Lingo for the Git binding, you may or may not want to enable bind detection. For example, you might create a folder where you store all of your Lingo projects, and you use an external tool to create the bindings, with a .Git folder stored at the root level of that main folder. In that case, you would want the bind detection option on this tab disabled for Git.

 **NOTE** You can use bind detection as an alternative to importing a Lingo project. If you have received a Lingo project file (e.g., by copying it from a server, by opening it from a network location), you can simply open the file and Lingo will search for and apply existing source control bindings.

Disabling Providers

By default, when a project is bound to source control, the provider (Git, Perforce Helix Core, Subversion, or Team Foundation Server) is enabled. This means that the source control interface elements in Lingo are visible, and you can use them to perform various tasks (e.g., commits, synchronize changes).

Disabling a provider means that the source control interface elements are no longer shown. This does not mean you cannot use source control. As long as the provider is still bound to the project, you can perform source control tasks in a third-party tool outside of Lingo.

How to Disable a Provider for All Projects Globally (Git Only)

This is the easiest method if you have multiple projects and you want to disable the Git provider in Lingo for all of them.

1. Select **File > Options**.
2. Select the **Source Control** tab.
3. In the **Bind Detection** section, remove the check mark next to **Git**.
4. Click **OK**.
5. Close the project and then reopen it so that the option can take effect.

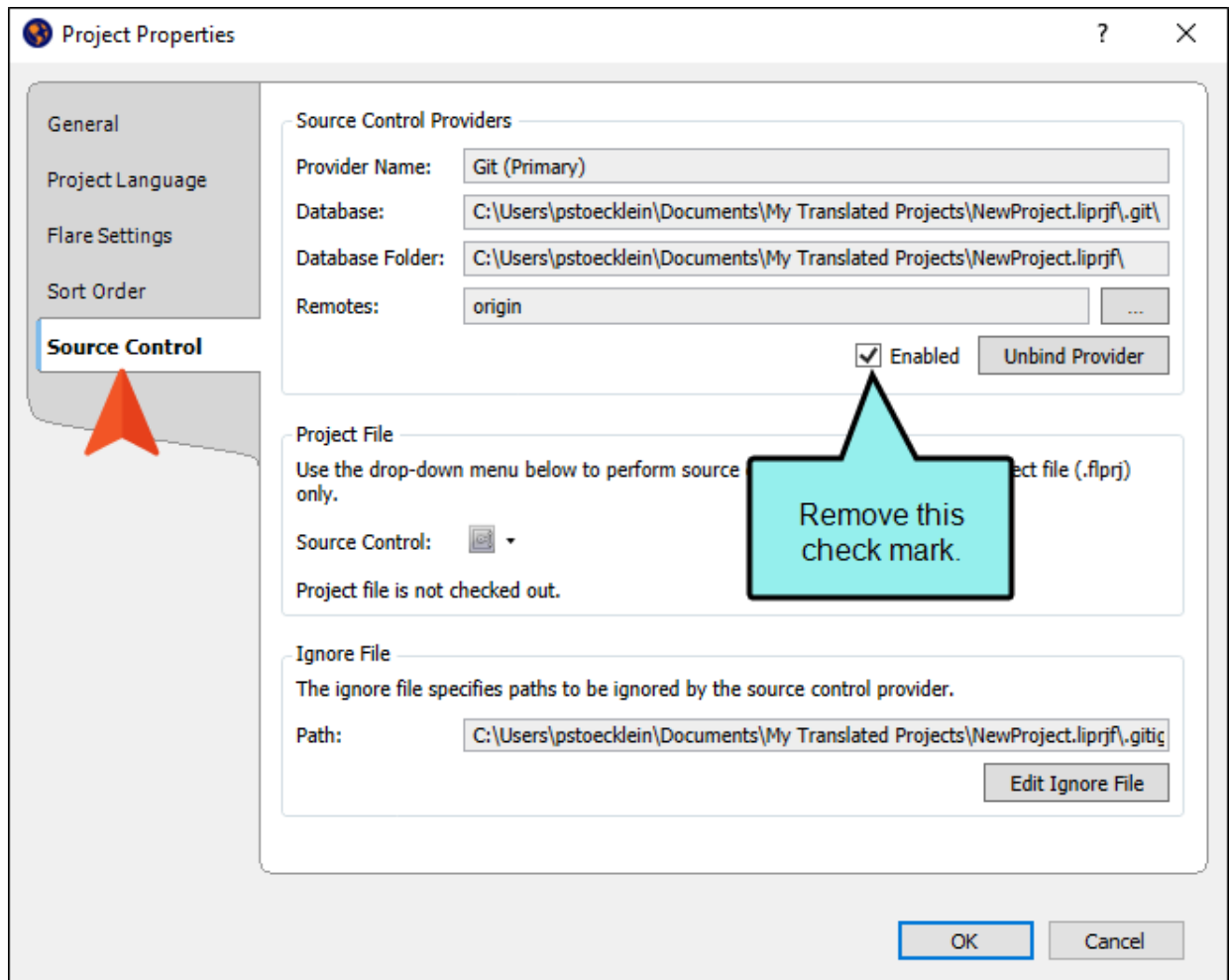
The provider option that is found in the Project Properties dialog or the Settings view in the Source Control Explorer (i.e., the "Enabled" check box) will then automatically be disabled.

How to Disable a Provider in an Individual Project

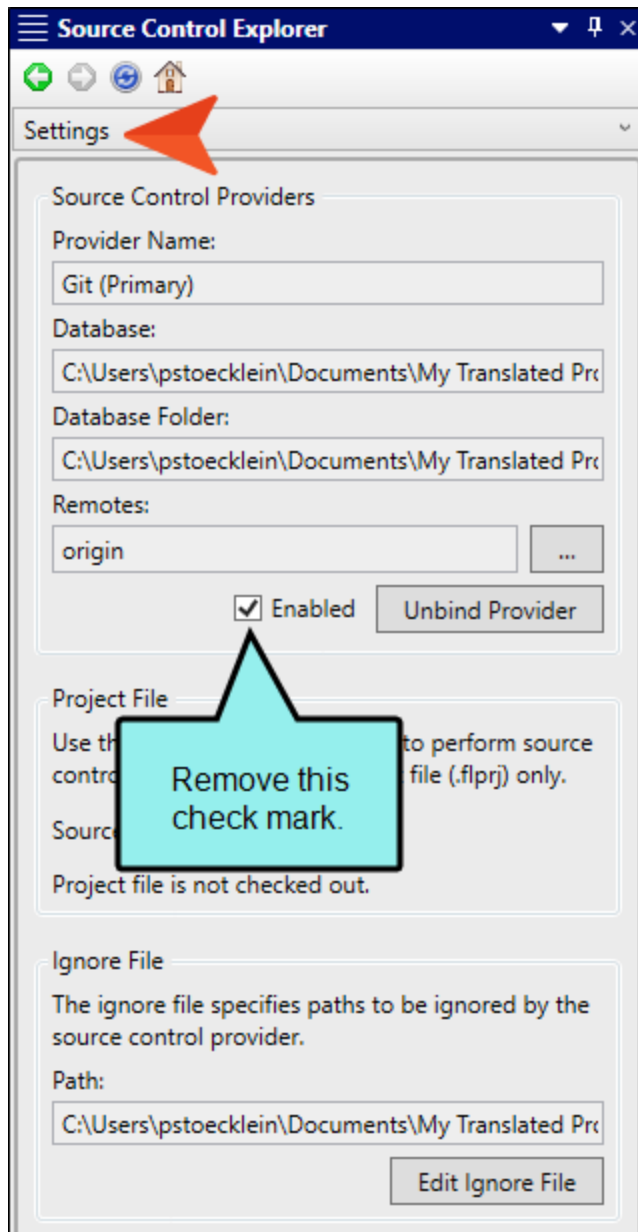
Use this method if you want to disable a provider in just one project, rather than many projects.

1. Do one of the following, depending on the part of the user interface you are using:
 - **Project Properties** Select **File > Project Properties**.
 - **Source Control Explorer** Select **View > Source Control Explorer**. Then, in the window pane, click Settings.
2. Click **Enabled** to remove the check mark.


PROJECT PROPERTIES DIALOG




SOURCE CONTROL EXPLORER




3. If you used the Project Properties dialog, click **OK**.


 **NOTE** If you disable a Git provider, the local repository will continue to track your changes in case you later decide to enable the provider once again.

If you disable one of the other providers (Perforce Helix Core, Subversion, Team Foundation Server), your changes after that point will not be tracked. Therefore, if you later enable the provider again, it will not have recorded any changes made since the time that you disabled it.

 **NOTE** When you disable a provider, that information is written to the registry on your computer.

 **NOTE** If you disable a provider, but then perform one of the following actions in the Lingo interface, the provider will automatically become enabled once again.

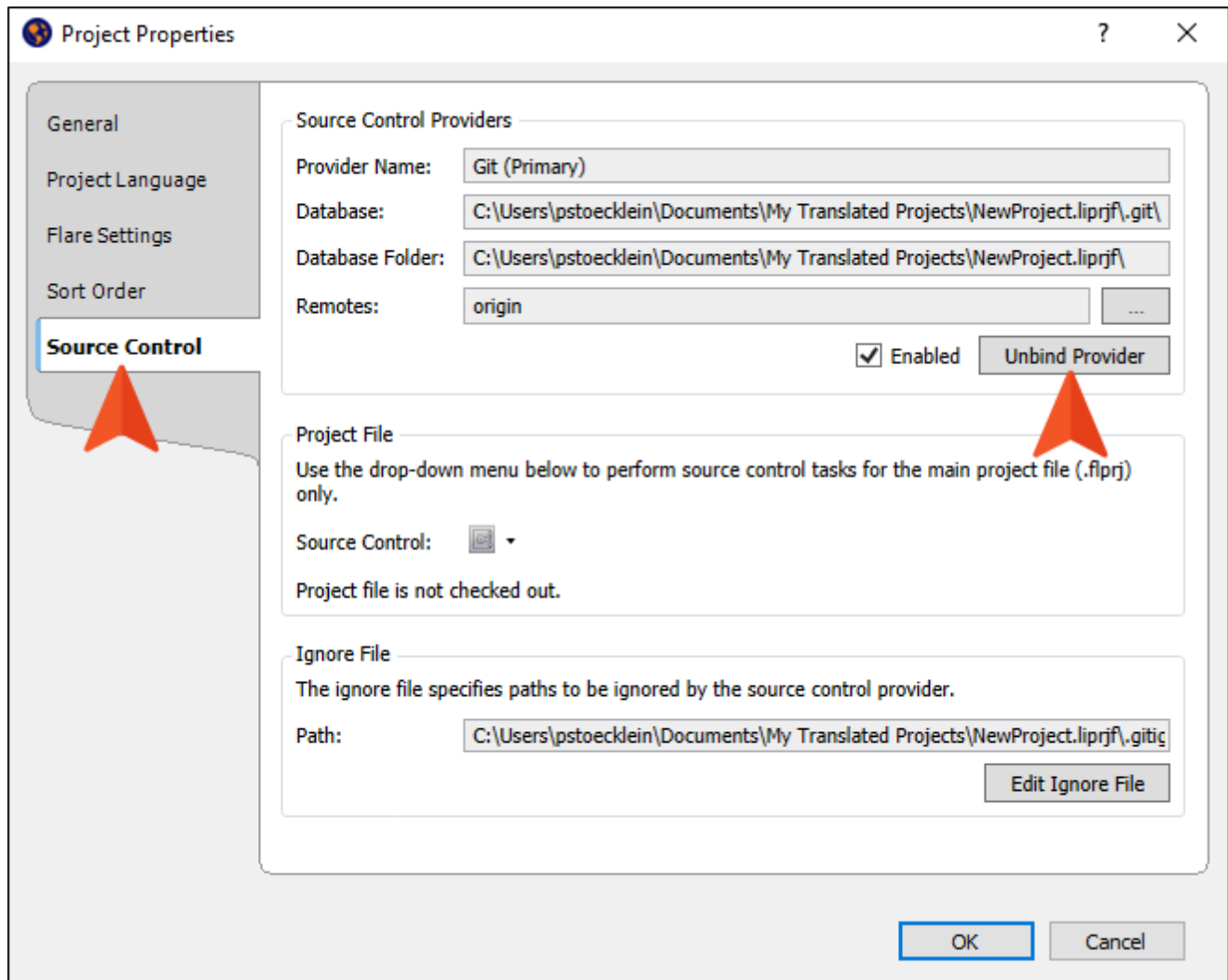
- Bind an existing project
- Bind a new project
- Import a project from source control

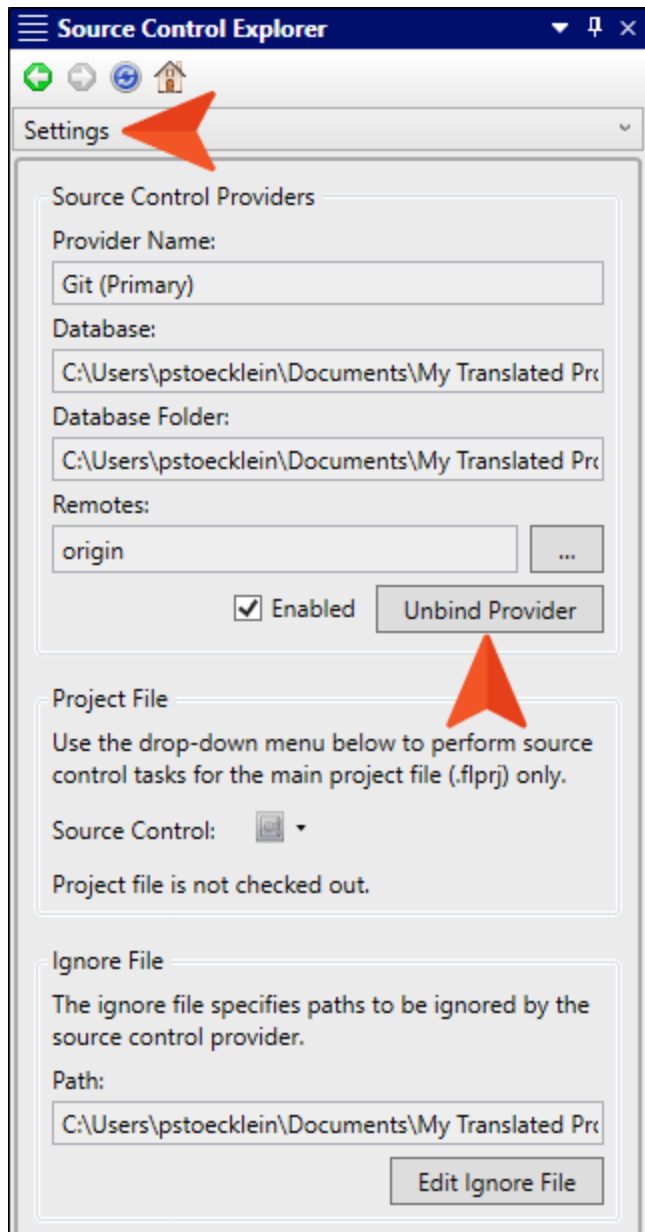
 **NOTE** Having a provider enabled in Lingo does not interfere with your workflow if you are performing source control actions exclusively outside of Lingo. Even if a provider is enabled in the project and the source control user interface elements are visible, this does not mean Lingo is automatically performing any source control actions with your files. It simply means Lingo is recognizing the binding, so it reflects your activities (e.g., the Pending Changes window is populated when you make edits in topics). However, if you prefer not to see any of this in Lingo, you can disable the provider.

Unbinding Providers

When you unbind a provider, it means you are removing the connection altogether between the Lingo project and the local repository.

You can unbind a provider via the Project Properties dialog or the Settings view in the Source Control Explorer. Click the **Unbind Provider** button.





I Ignore File for Git

The `.git` folder is the local repository for a Lingo project and is included when your project is bound to Git. The contents of this folder are updated automatically when you perform source control tasks. The additional `.gitignore` file is necessary to make sure that certain folders are ignored when you push and pull files. In most cases, you do not need to do anything with the `.git` folder or `.gitignore` file.

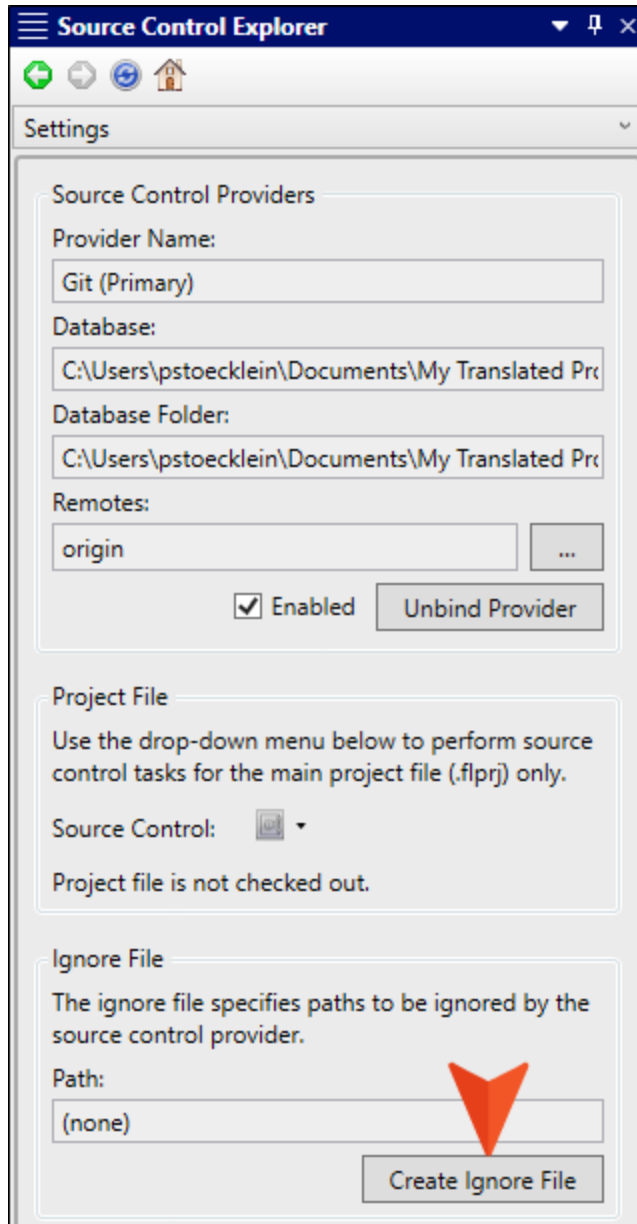
If you bind a project to Git outside of Lingo (i.e., you do not use Lingo's interface to do the binding), you should make sure that you have a `.gitignore` file. You can add the `.gitignore` file by selecting an option in the Source Control Explorer or Project Properties dialog. Once you have a project containing a `.gitignore` file, the button in the interface changes to "Edit Ignore File," so that you can open the file to make edits to it.

How to Add an Ignore File

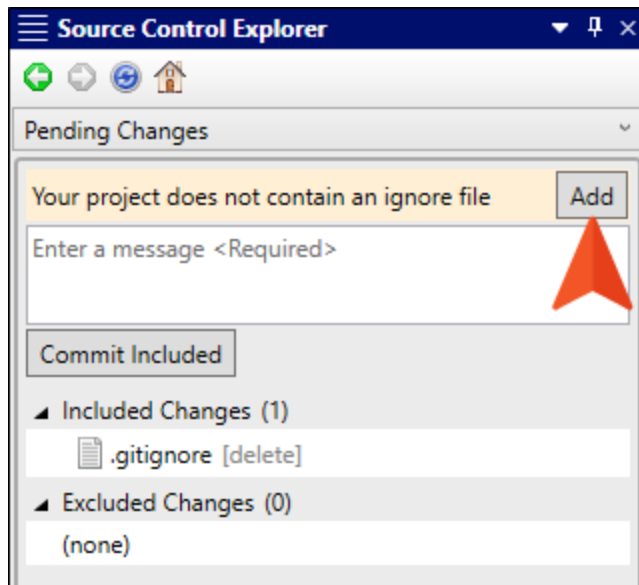
The following steps show how to do this in the Source Control Explorer. You can also accomplish this in the Project Properties dialog (**Project > Project Properties**) on the **Source Control** tab.

1. Select **View > Source Control Explorer**.
2. Do one of the following:

- Click **Settings**. Then in the **Ignore File** section, click **Create Ignore File**.



- Click **Pending Changes**. Then next to the message at the top, click **Add**.

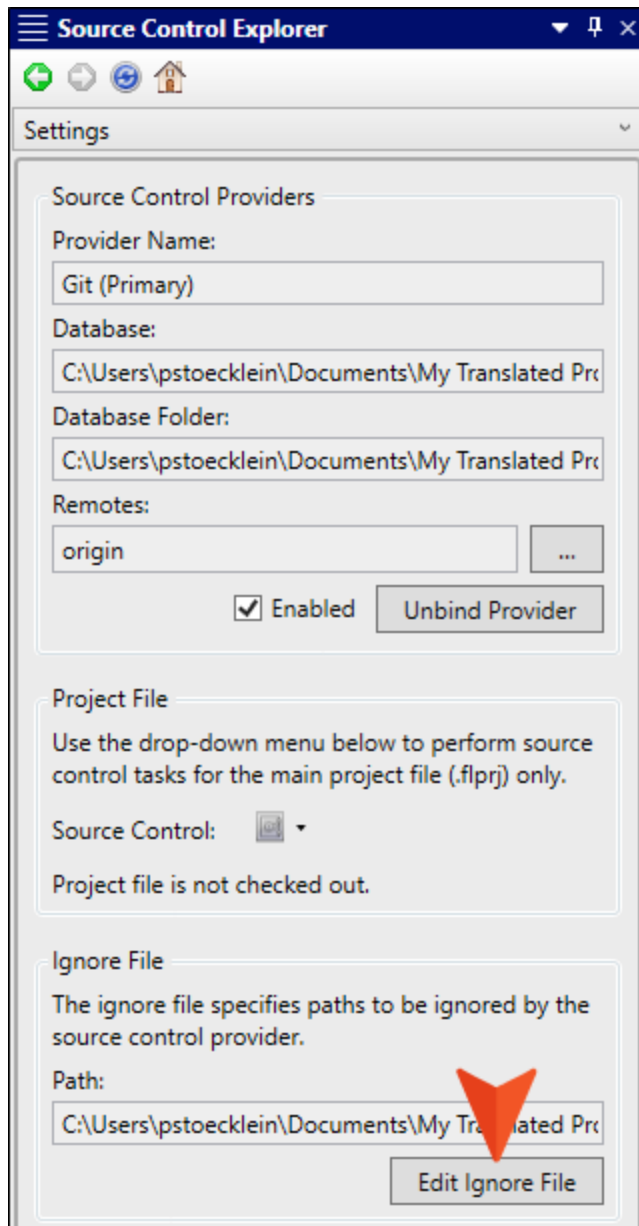


3. Make edits in the Text Editor and save your changes.

How to Edit an Ignore File

The following steps show how to do this in the Source Control Explorer. You can also accomplish this in the Project Properties dialog (**Project > Project Properties**) on the **Source Control** tab.

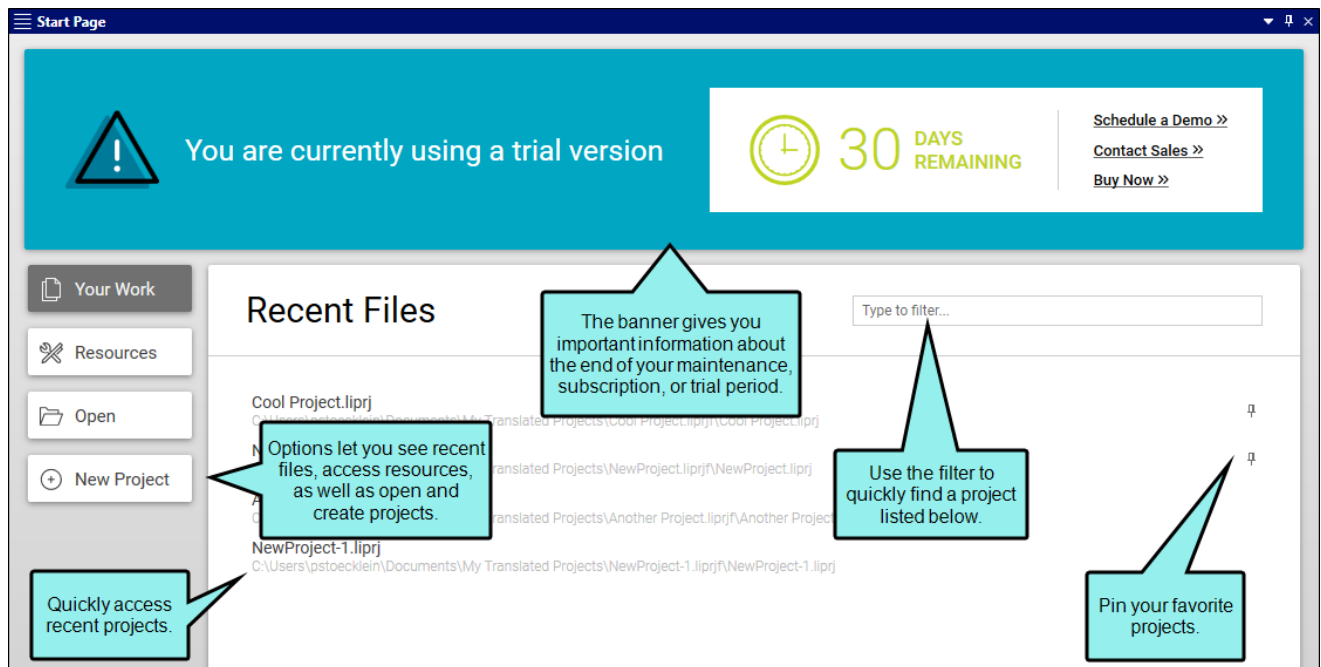
1. Select **View > Source Control Explorer**.
2. Click **Settings**.
3. In the **Ignore File** section, click **Edit Ignore File**.



4. Make edits in the Text Editor and save your changes.

Start Page Redesign

The Start Page in Lingo has been redesigned to make it more efficient, display a banner with the expiration date for the license type, provide links to take action, and allow for more space to display links and information.



This chapter discusses the following:

Page Options on Left	33
License Expiration Banner	34
Responsive to Window Size	35

I Page Options on Left

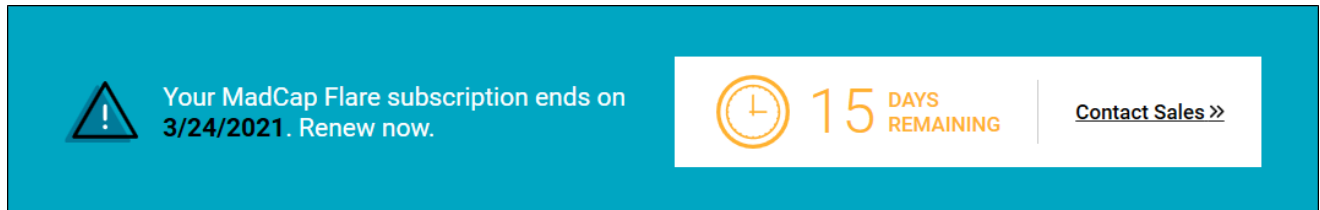
A series of options on the left side of the Start Page can be used to access different information or perform specific actions.

- **Your Work** Displays recent projects, with a filter field at the top. This is the same information that was previously shown on the left side of the Start Page in previous versions. But by clicking this option, you can now see more of it.
- **Resources** This is the same dynamic information that was previously displayed at the bottom of the Start Page. Again, by moving it to a separate page view, more space becomes available to list resources.
- **Open** Lets you open an existing Lingo project.
- **New Project** Opens the wizard so that you can create a new Lingo project.

License Expiration Banner

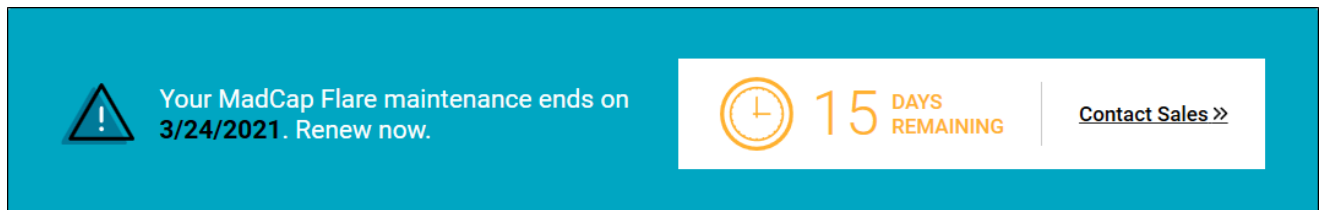
Depending on your type of license (e.g., subscription, maintenance, trial), a banner displays when a deadline for renewing or purchasing is approaching. You can click a link to take action (e.g., Contact Sales). If you have more than 90 days left, you will not see a banner.

Subscription



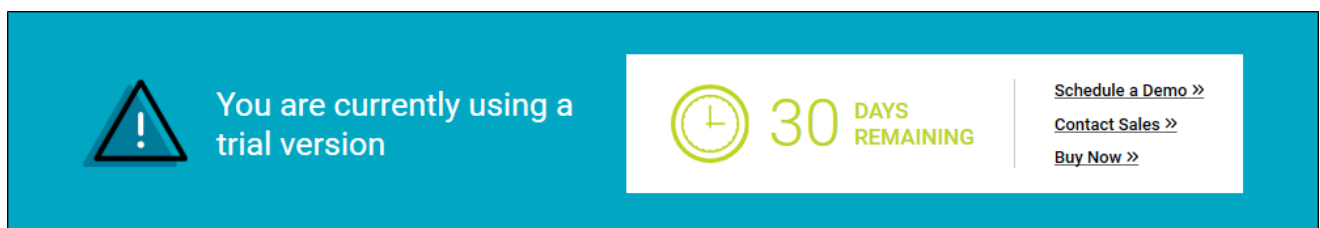
A blue banner with a white warning triangle icon on the left. The text reads: "Your MadCap Flare subscription ends on 3/24/2021. Renew now." To the right, there is a white box containing a clock icon, the number "15" in large orange font, and "DAYS REMAINING" in smaller orange font. Further right, there is a link "Contact Sales >>" in blue text.

Maintenance



A blue banner with a white warning triangle icon on the left. The text reads: "Your MadCap Flare maintenance ends on 3/24/2021. Renew now." To the right, there is a white box containing a clock icon, the number "15" in large orange font, and "DAYS REMAINING" in smaller orange font. Further right, there is a link "Contact Sales >>" in blue text.

Trial

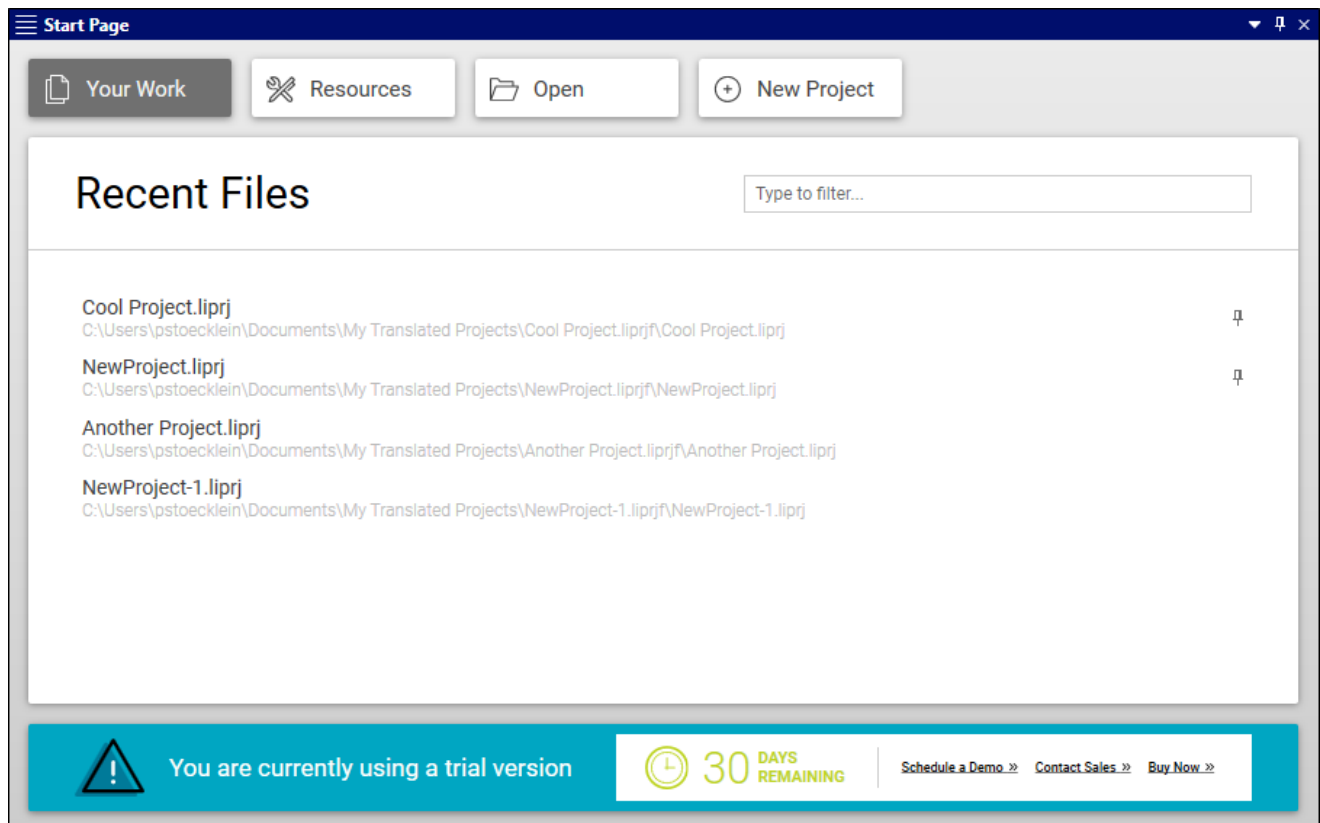


A blue banner with a white warning triangle icon on the left. The text reads: "You are currently using a trial version". To the right, there is a white box containing a clock icon, the number "30" in large green font, and "DAYS REMAINING" in smaller green font. Further right, there are three links: "Schedule a Demo >>", "Contact Sales >>", and "Buy Now >>" in blue text.

I Responsive to Window Size

The layout of the Start Page changes depending on the size of the window.

For example, if the window is large, the page options are shown on the left, and the information banner is at the top. But if the window is reduced in size, the page options are shown at the top, and the information banner is at the bottom. A horizontal bar might also appear if the window becomes narrow enough.



PDFs

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

I Cheat Sheets

Shortcuts Cheat Sheet

I User Guides

Alignment Guide

Getting Started Guide

Key Features Guide

Source Control Guide: Git

Source Control Guide: Perforce Helix Core

Source Control Guide: Subversion

Source Control Guide: Team Foundation Server

Termbases Guide

Touring the Workspace Guide

Translation Guide

What's New Guide