Git Integration

Table of Contents

- Overview
- Git Setup and Configuration
- Creating a Git Repository
- Cloning a Git Repository
- Opening an Existing Git Repository
- Working with Changes
- Committing Changes
- Pull, Push, and View Repository Status
- Git Command Output
- Switching the Active Repository
- Deleting a Repository
- Authentication for Private Online Repositories

Git integration for Profound.js was added in version 4.10.0.

Overview

Git integration allows you to manage Git repositories within the Visual Designer. Repositories can be created using the Designer, or cloned from GitHub or other online sources. The Designer can track changes to the repository, show diffs, make commits, and more.

Git Setup and Configuration

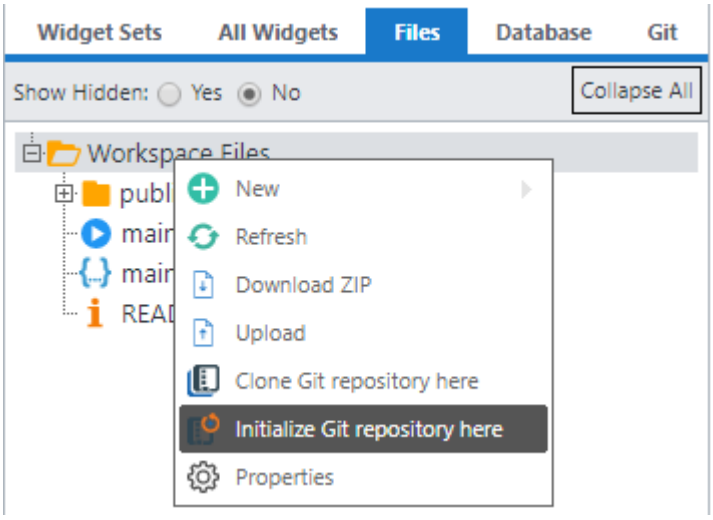
This section only applies to the Profound.js version of the Visual Designer. On NodeRun.com, Git setup/configuration is taken care of for you.

The Designer uses your system's Git command. If Git is not installed on the system, or not found on your system's PATH environment variable, Git integration features will not be available. If Git is installed in a location not included in the PATH environment variable, the gitPath configuration option can be used to tell the Designer where to locate it.

Git requires that you configure your user display name and email address before you can commit changes. See here for first time Git setup instructions. Git configuration commands can be run in your favorite command shell, or using the Designer's built-in terminal emulator.

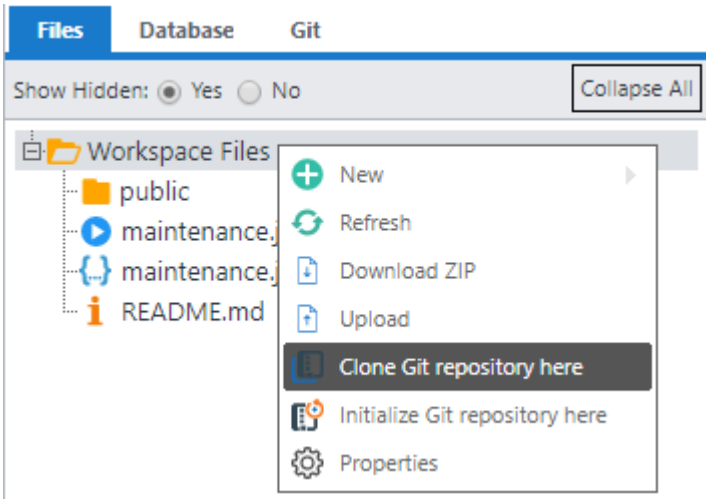
Creating a Git Repository

To create a Git repository, right-click a directory in the Files tree and select **Initialize Git repository here.**

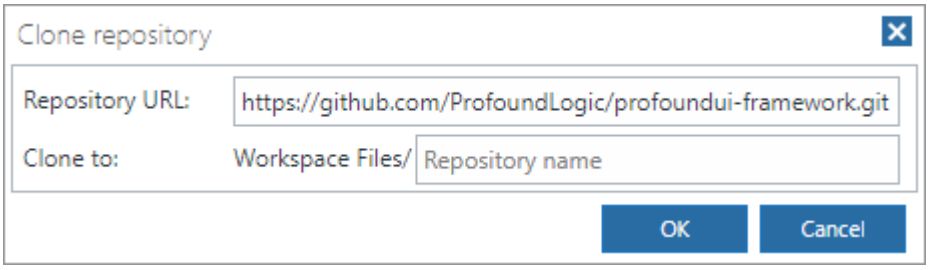


Cloning a Git Repository

To clone a Git repository, right-click the directory where you would like the repository to go and select **Clone Git repository here**.



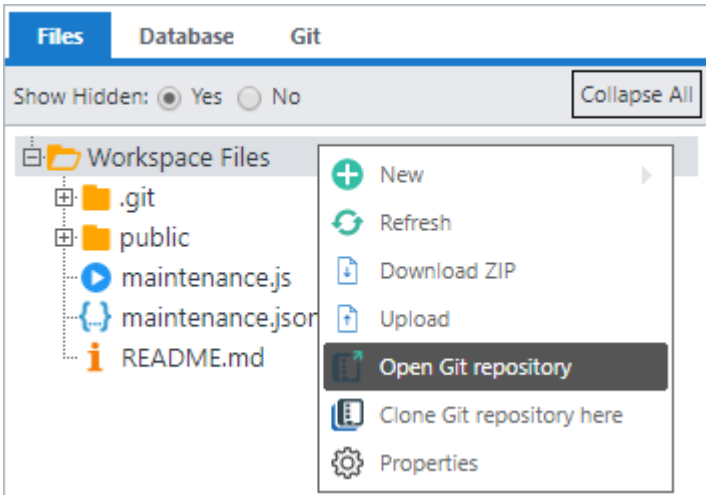
On the Clone Repository dialog, enter a repository URL and optional directory name:



The URL can be given in any format accepted by the Git command – i.e. HTTP/HTTPS, SSH, or a directory path on the local file system. If the 'Clone To' directory name is omitted, the name of the repository will be used as the directory name.

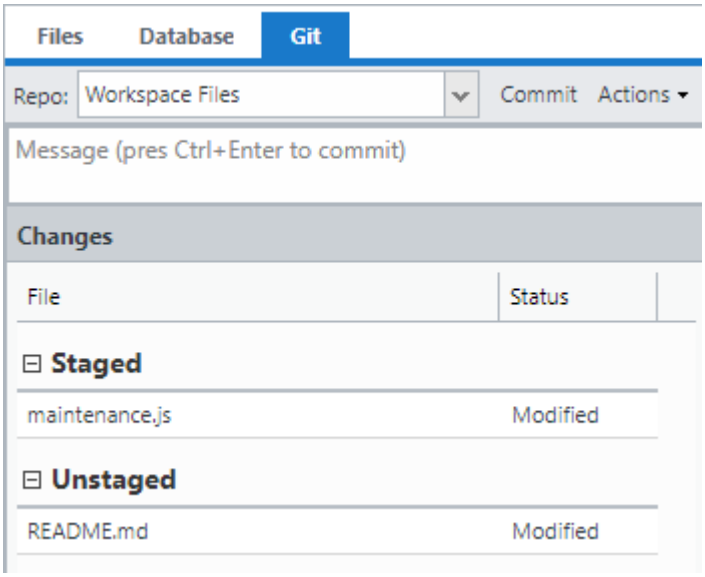
Opening an Existing Git Repository

If you already have a Git repository on the file system, you can open it by right-clicking the directory and selecting **Open Git repository**.



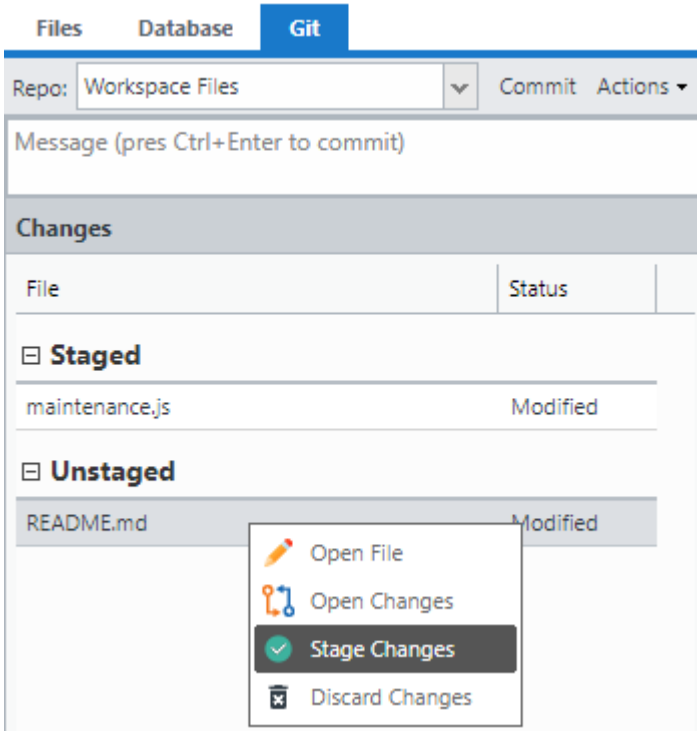
Working with Changes

The Designer will automatically monitor the repository's working directory and index for changes and display them on the Git Tab:

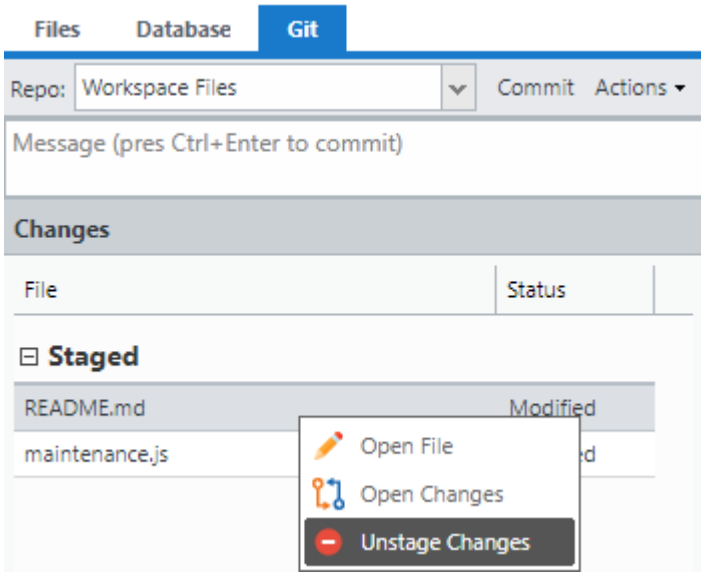


Entries are grouped into Staged and Unstaged changes. Multiple items can be selected on the Git tab by shift+click or ctrl+click. Multi-select works only within the same (Staged or Unstaged) group of items.

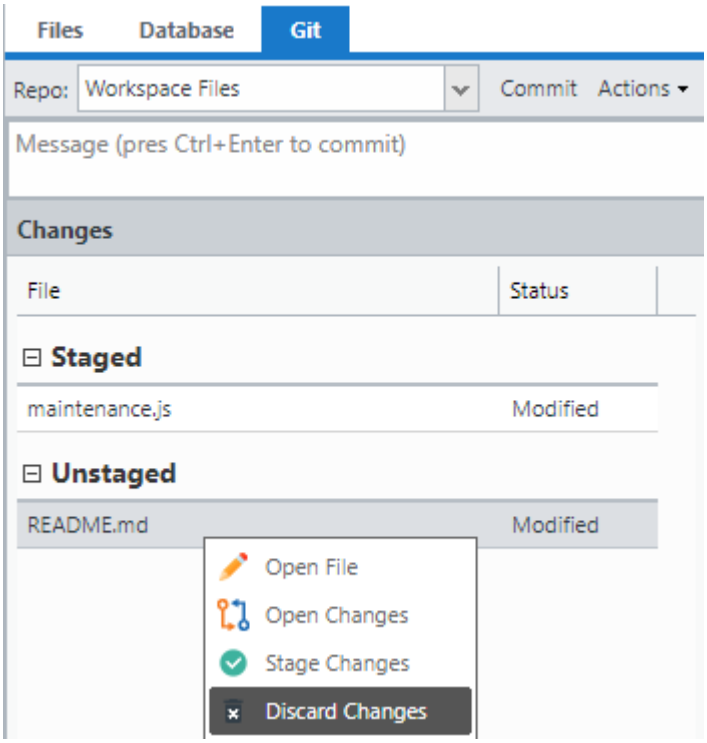
To stage changes, right-click an item under Unstaged and select **Stage Changes**.



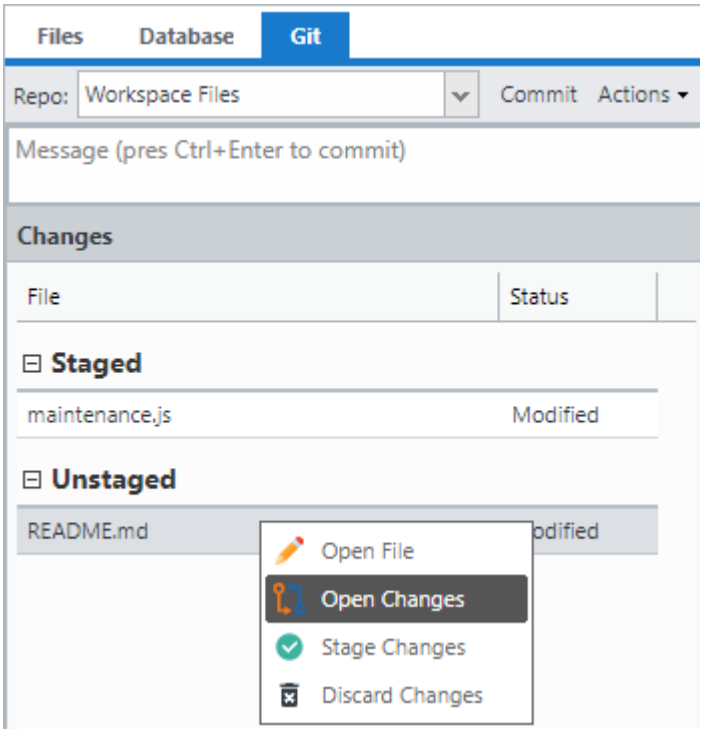
To unstage changes, right-click an item under Staged and select **Unstage Changes**.



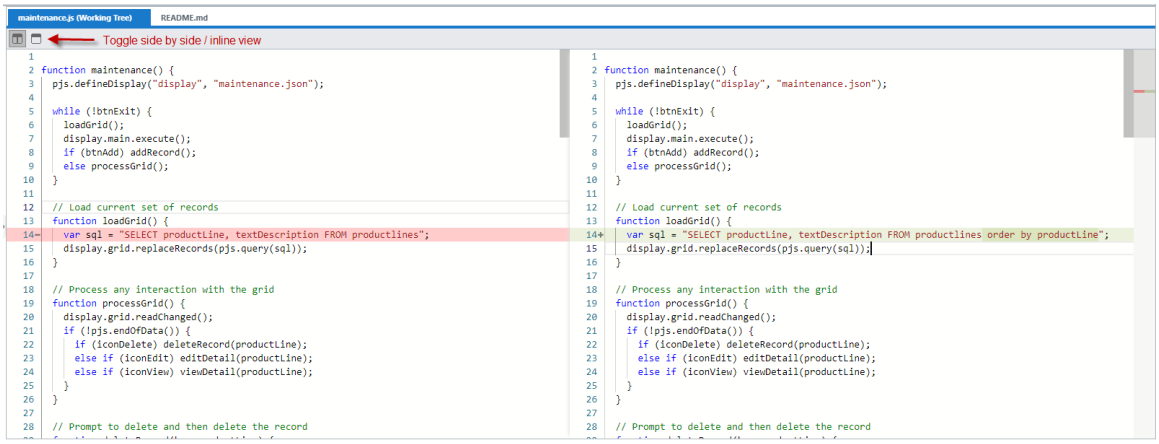
To discard changes, right-click an item under Unstaged and select **Discard Changes**.



To view changes, right-click an item under either group and select **Open Changes.**



If the item is for a new file, the file will open in a normal editor. This is the same as selecting **Open File.** If the item is for a modified file, then a split view will open that displays the differences.



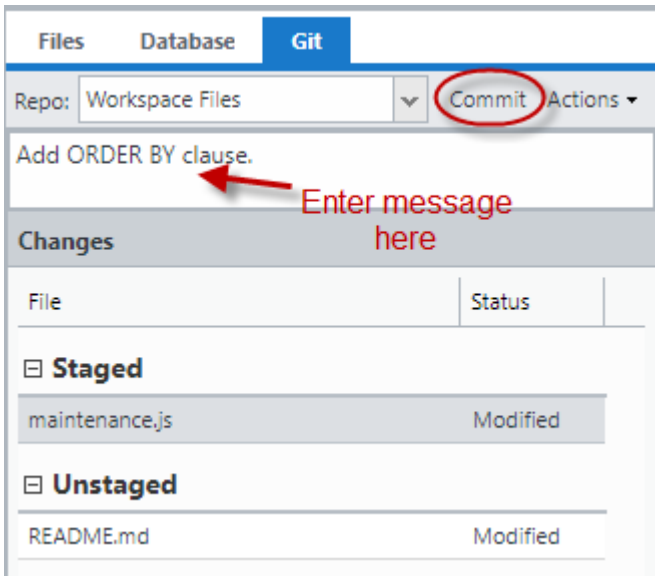
If viewing a diff in the working tree (Unstaged), then edits can be made on the right-side. If viewing a diff in the index (Staged), then the diff view is read-only. The buttons at the top of the editor panel can be used to toggle side-by-side vs line by line view.

Committing Changes

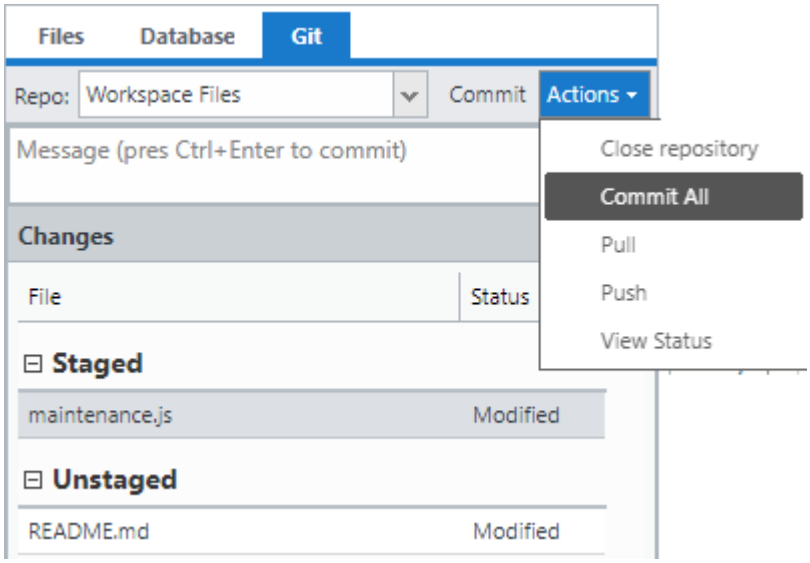
There are two ways to commit changes:

- Commit staged changes only
- Commit all changes

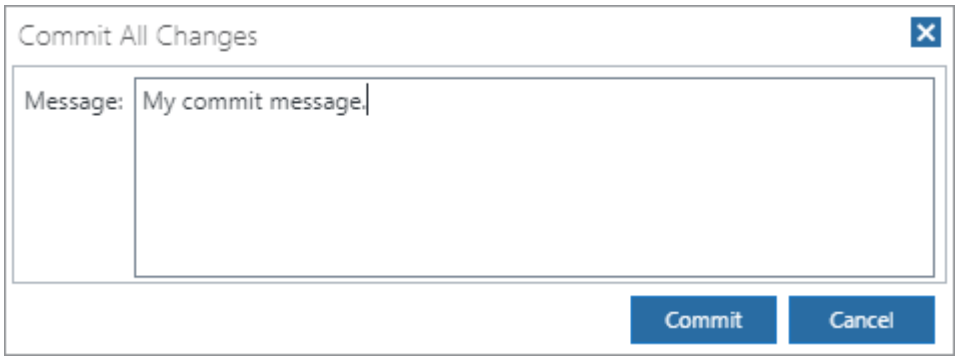
When there are staged changes, a message box and Commit button will appear at the top of the Git panel. To commit staged changes, enter a commit message and press ctrl+Enter or click **Commit**.



To all changes (both Staged and Unstaged), use the Commit All option on the **Actions** drop down:

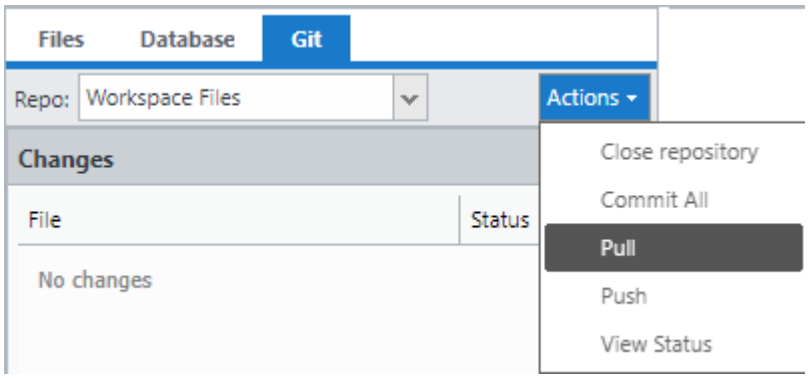


Enter your message in the Commit All Changes dialog and click **Commit**.



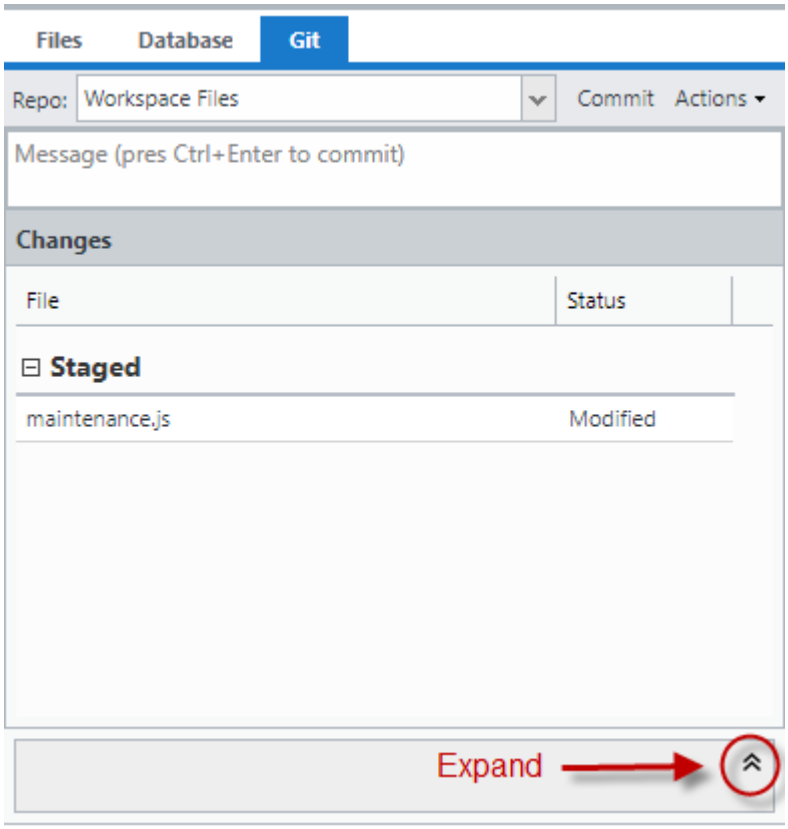
Pull, Push, and View Repository Status

You can pull/push to/from a remote and view repository status using the options on the **Actions** menu.

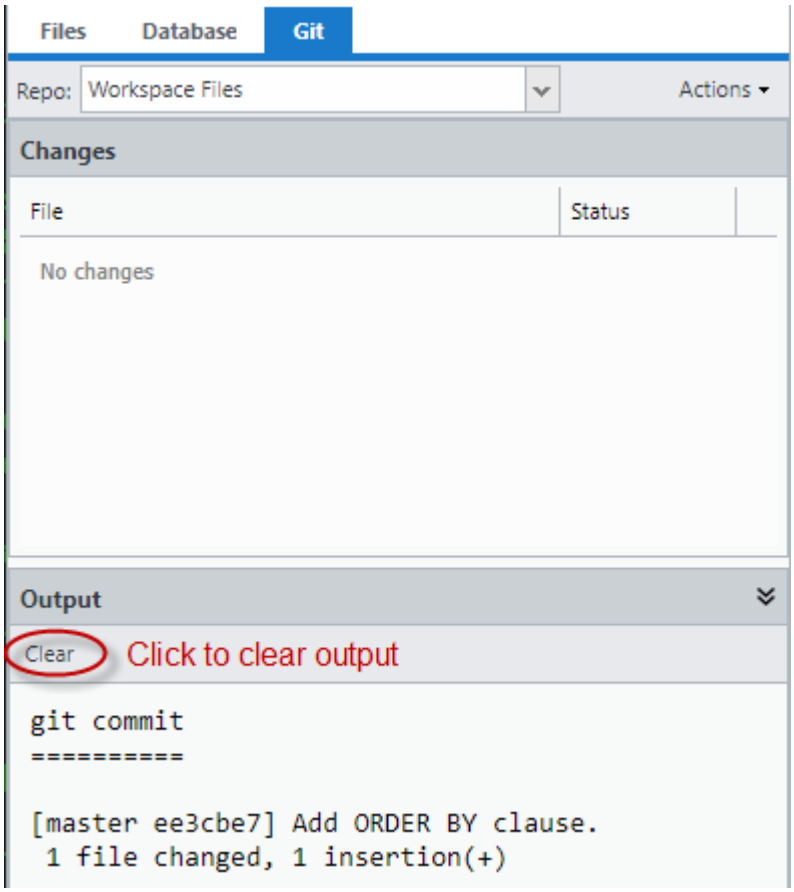


Git Command Output

Output from Git for all actions is displayed in the output panel at the bottom of the Git tab. If the panel is collapsed, click on the arrow icon to expand it:



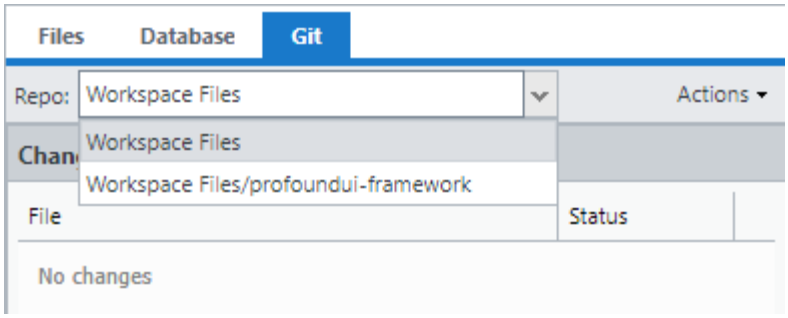
Output from all commands run during your Designer session is retained. Click the **Clear** button at the top of the output panel to clear previous messages.



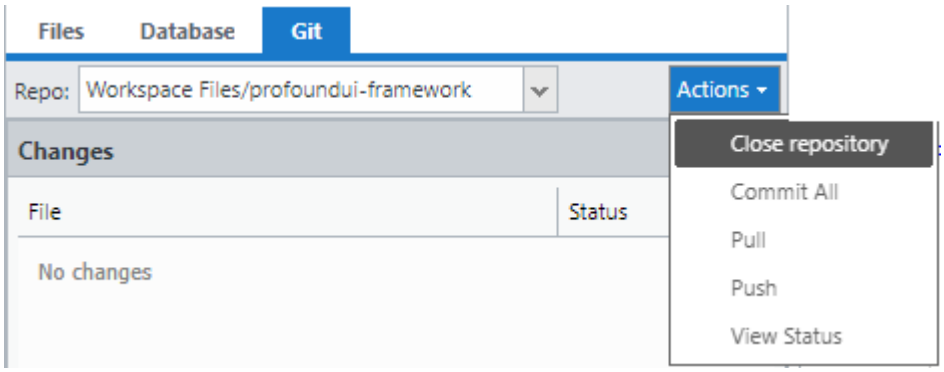
The output panel will automatically expand if there is an error when running a command.

Switching the Active Repository

The Designer stores a list of repositories that you have worked with. To change the active repository, use the drop down box at the top of the Git tab.



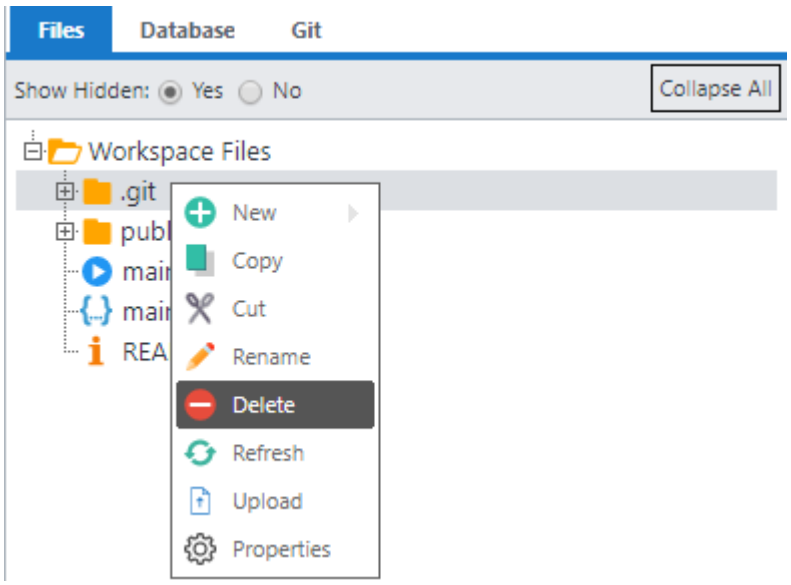
If you no longer need to work with a repository and want to remove it from the list, use the **Close** option on the **Actions** menu.



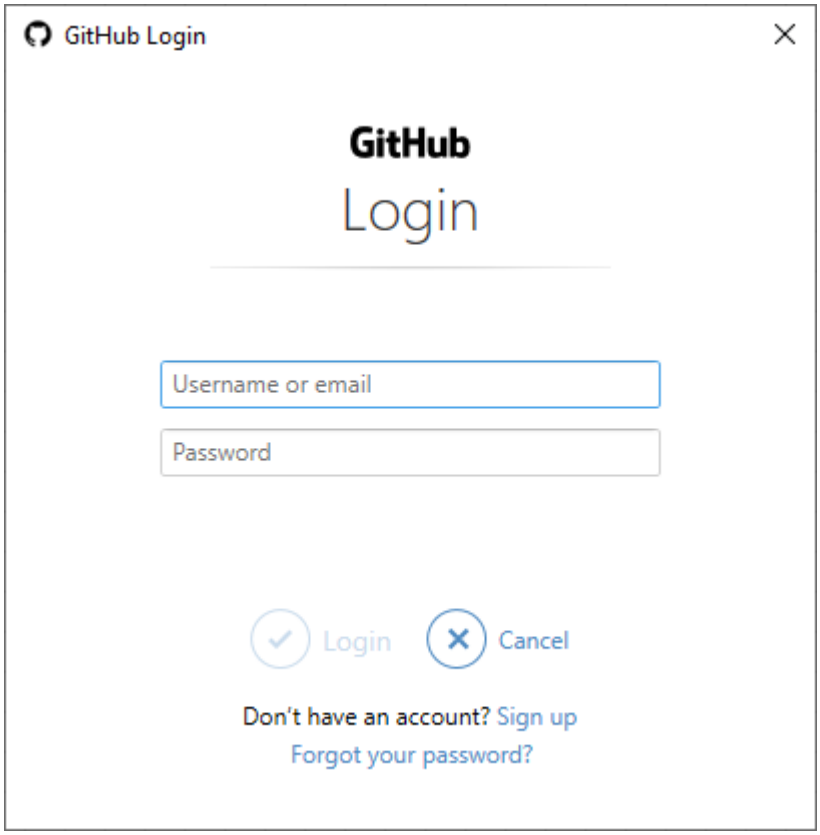
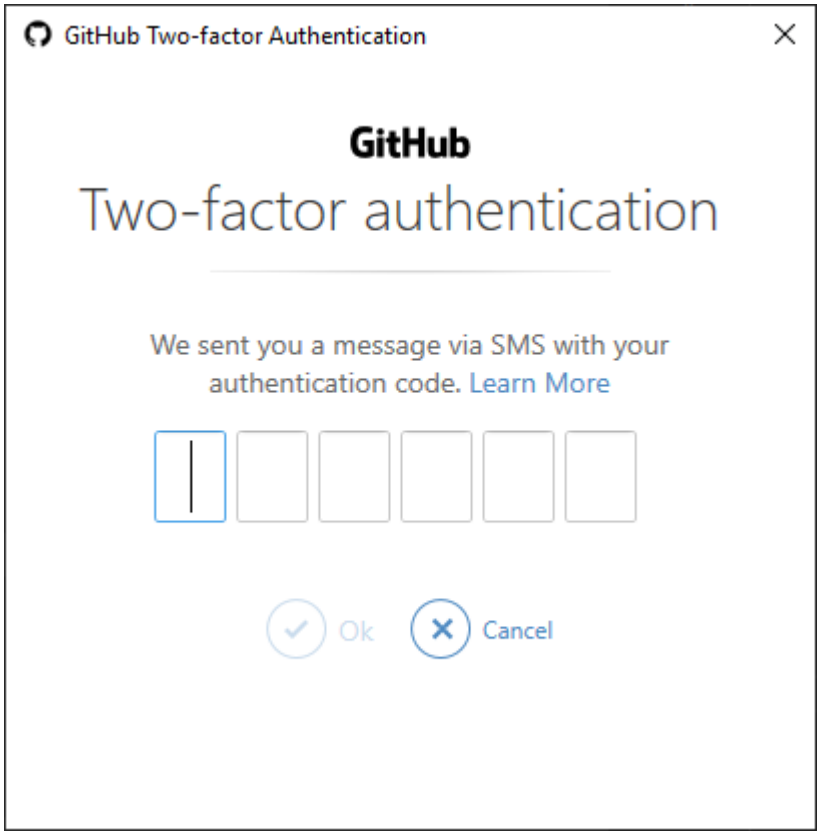
This option does not delete the repository, it simply removes it from your list of repositories to work with.

Deleting a Repository

Git repositories are stored in a hidden directory named '.git'. To delete a repository using the Files tab, select **Show Hidden = Yes**, right-click the .git directory, and select **Delete**.



Authentication for Private Online Repositories

To clone and to pull/push to/from online (i.e. GitHub) repositories that require authentication, it's recommended to use SSH keys. See here for instructions on setting up SSH key authentication with GitHub. You can use your favorite command shell or the Designer's built-in terminal emulator to run the commands for creating/inspecting keys.

Profound.js on Windows also supports authentication over HTTP/HTTPS for GitHub, including 2-factor authentication. This is not supported on other platforms. Git for Windows will automatically prompt for authentication as needed.