

Checking in Files

Different version control systems have different semantics for the action of uploading changed files to the repository. Two common terms are *check in* and *commit*.

In those version control systems, for example, [Git](#), that distinguish between local and remote repositories, the term *commit* denotes uploading changes to a *local* repository. Uploading changes to a remote repository is referred to as *push*.

Regardless of the terminology, you can perform this operation with the VCS configured for a directory from within IntelliJ IDEA.

To check in (commit) changed files, perform these general steps

1. In the [Changes](#) tool window, select one or more files you want to check in (commit) to version control.
2. Open the [Commit Changes](#) dialog box by doing one of the following:
 - On the **Changes** tool window toolbar, or on the main toolbar, click .
 - Press **Ctrl+K**.
 - On the main menu, choose **VCS | Commit Changes**.

3. Review the changes to be committed in the [Details](#) pane. To do that, unfold the **Details** pane if it is hidden, and select the file in question in the [Changed Files](#) area.

The **Details** pane shows the base version and the local copy of the selected file. Examine the details of each change:

- To move to the next updated piece of code, click the **Next Change** button .
 - To return to the previous updated code fragment, click the **Previous Change** button .
 - To expand or narrow the context of an updated code fragment, position the cursor at the change in question, click the **More/Less Lines** button , and then specify the number of lines to be shown above and below the current code fragment.
4. Add a commit comment. As you type, IntelliJ IDEA checks the spelling and highlights words in question.

This functionality is available if the [Spelling code inspection](#) is enabled.

5. Specify which actions should be performed on the files before and after submitting them to the repository.
6. Click the **Submit/Commit** button to launch the *Check-in Changes* operations.

For Git and Mercurial. To have the changes immediately pushed to your Git or Mercurial repository, do one of the following:

- Hover the mouse pointer over the **Submit/Commit** button and select **Commit and Push** on the context menu.
- From the **Submit/Commit** drop-down list, select **Commit and Push**.

7. To save the changes as a patch in a text file, hover the mouse pointer over the **Submit/Commit** button and select **Create Patch** on the context menu.

Alternatively, use the **Submit/Commit** drop-down list to select the **Create Patch** item.

In the **Create Patch** dialog box, that opens, [configure the patch creation](#).

8. If any error occurs when trying to commit, IntelliJ IDEA displays an error message. For example, you might have changed a file that has been already edited by another team member, or you might run into a branch conflict. In these cases, you need to [merge edits](#), or [update your local copy](#). The error messages are VCS-specific.

Users of [JetBrains TeamCity](#) can obtain the TeamCity plugin for IntelliJ IDEA. Among the features of this plugin is Remote Run, which enables you to create a special *personal* build that does not affect the *real* build. Your changes are built and run through your test suite. If all tests are passed, your changes are automatically committed to version control.

See Also

Procedures:

- [Resolving Commit Errors](#)

Reference:

- [Version Control Reference](#)
- [Commit Changes Dialog](#)
- [Changes Tool Window](#)

Web Resources:

- [Developer Community](#) 