

Switching Between Working Directories

The Mercurial integration with IntelliJ IDEA provides the possibility to switch update the repository's [working directory](#) to the specified [changeset](#) or a specific [line of development](#). Changesets can be identified by their hashes or by previously assigned [tag identifiers](#).

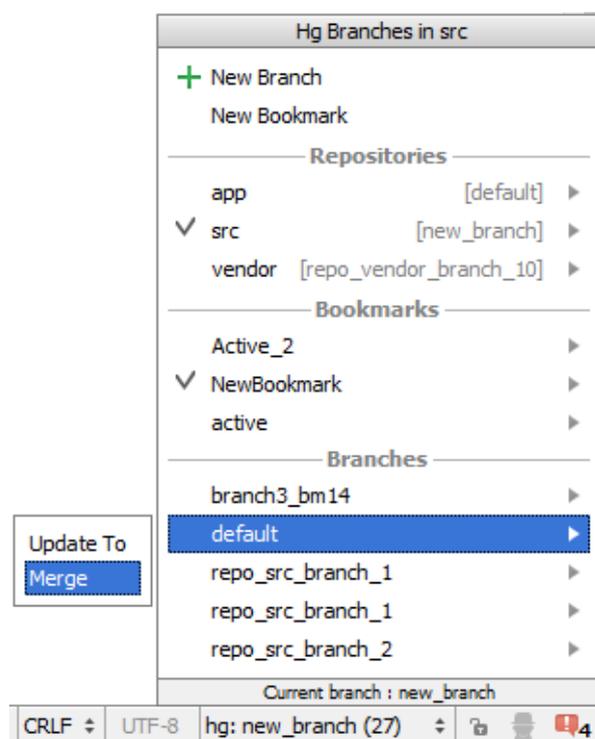
On this page:

- [Opening the Branches pop-up list](#)
- [Switching to another named branch or bookmark](#)
- [Switching to another changeset](#)

You can update a named branch or a bookmark to another named branch, another bookmark, or a specific changeset identified by a tag or a revision number.

- Updating a named branch or a bookmark to another named branch or bookmark means updating to its *head*.

Updating to named branches and bookmarks can be invoked through the menu item **VCS | Mercurial | Update to**, which opens the Mercurial-specific **Switch Working Directory** dialog box of from the **Branches** pop-up list.



- Updating a named branch or a bookmark to a changeset means updating the branch *head* to the specified changeset. A changeset can be identified either by a revision number or a tag, see <http://mercurial.selenic.com/wiki/Tag?highlight=28\bCategoryGlossary\b29>.

Updating a named branch or a bookmark to a specific changeset can be invoked *only* through **VCS | Mercurial | Update to**.

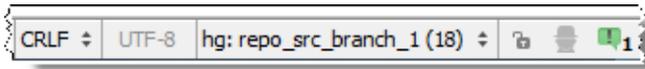
By default, *Mercurial* requires that before update the current working directory should be *clean*, that is, it should not contain any uncommitted changes. Otherwise the update operation fails and IntelliJ IDEA shows the corresponding error message. The message also recommends that you clean the current working directory by running the `hg update <target branch, bookmark, or changeset> -C` to discard the uncommitted changes.

If your current working copy is not clean, you can either commit the changes or shelve them as described in [Shelving and Unshelving Changes](#). IntelliJ IDEA provides the possibility to discard any uncommitted changes when the update operation is already invoked. This option is available only in the Mercurial-specific **Switch Working Directory** dialog box.

Opening the Branches pop-up list

To open the **Branches** pop-up list, do one of the following:

- On the **Status** bar, click the name of the current named branch in the dedicated **hg** area.



- On the main menu, choose **VCS | Mercurial | Branches**.
- On the context menu of the **Editor** or **Changes** tool window, choose **Mercurial | Branches**.

Switching to another named branch or bookmark

Updating a named branch or a bookmark to another named branch or bookmark means updating to its *head*.

1. Make sure, your current working directory is *clean*, that is, it does not contain any uncommitted changes. Commit or shelve the changes, if any.

If you invoke update through the **Switch Working Directory** dialog box, you can also prevent conflicts by having any uncommitted changes discarded.

2. Invoke update by doing one of the following:

- In the **Branches** pop-up list, click the name of the branch or bookmark to update to, then choose **Update to** on the pop-up menu:
- Choose **VCS | Mercurial | Update to** on the main menu or **Mercurial | Update to** on the context menu of the **Editor**.

In the **Switch Working Directory** dialog box that opens:

1. Choose the target repository from the **Repository** drop-down list which shows all the Mercurial repositories available under the current project root.
2. Choose the **Branch** or **Bookmark** option and choose the named branch or bookmark to update the current working directory to.
3. To prevent failures during update if the current working directory is not clean, select the **Overwrite locally modified files (no backup)** check box. The uncommitted changes will be discarded.

Switching to another changeset

1. Make sure, your current working directory is *clean*, that is, it does not contain any uncommitted changes. Commit or shelve the changes, if any.

If you invoke update through the **Switch Working Directory** dialog box, you can also prevent conflicts by having any uncommitted changes discarded.

2. Invoke update by doing one of the following:
 - In the **Branches** pop-up list, click the name of the branch or bookmark to update to, then choose **Update to** on the pop-up menu:
 - Choose **VCS | Mercurial | Update to** on the main menu or **Mercurial | Update to** on the context menu of the **Editor**.

In the **Switch Working Directory** dialog box that opens:

1. Choose the target repository from the **Repository** drop-down list which shows all the Mercurial repositories available under the current project root.
 2. Choose the **Branch** or **Bookmark** option and choose the named branch or bookmark to update the current working directory to.
 3. To prevent failures during update if the current working directory is not clean, select the **Overwrite locally modified files (no backup)** check box. The uncommitted changes will be discarded.
3. Resolve conflicts. As soon as a conflict takes place, the **Files Merged with Conflicts** dialog box opens with a list of conflicting files. Use the controls of the dialog box to resolve the problems:
 - To have the version of the current working directory preserved, click **Accept Yours**.
 - To have the version of the branch you are merging with preserved, click **Accept Theirs**.
 - To resolve the conflicts manually, click **Merge** and use the **Conflict Resolution Tool**, as described in [Resolving Conflicts](#).

If no conflicts arise during update, the operation passes silently and the update log is shown in the **Version Control** tool window.

See Also

Concepts:

- [Version Control with IntelliJ IDEA](#)

Procedures:

- [Using Mercurial Integration](#)
- [Version Control with IntelliJ IDEA](#)

Reference:

- [Switch Working Directory Dialog](#)
- [Mercurial Reference](#)
- [Version Control Reference](#)

External Links:

- <http://www.selenic.com/mercurial/hg.1.html#update> 
- <http://mercurial.selenic.com/wiki/UnderstandingMercurial> 

Web Resources:

- [Developer Community](#) 

