Stashing and Unstashing Changes

Sometimes it may be necessary to revert your working copy to match the HEAD commit but you do not want to lose the work you have already done. This may happen if you learn that there are upstream changes that are possibly relevant to what you are doing or if you need to make some urgent fixes.

Stashing involves recording the difference between the HEAD commit and the current state of the working directory (stash).

> Changes to the index can be stashed as well.

Unstashing involves applying a stored stash to a branch.

You can apply a stash to an existing branch or create a new branch on its basis.

> A stash can be applied as many times as you need to any branch you need, just switch to the required branch. Keep in mind that:
>
> - Applying a stash after a series of commits results in conflicts that need resolving.
> - A stash cannot be applied to a "dirty" working copy, that is a working copy to which changes have been made since the latest commit.

You can store stashes in a list as long as you need and remove them from the list when necessary.

> As you might have noticed, *Stashing* and *Unstashing* have much in common with Shelving and Unshelving respectively.
>
> The only difference is in the way patches are generated and applied:
>
> - Patches with **stashed** changes are generated by Git itself. To apply them later, you do not need IntelliJ IDEA.
> - Patches with **shelved** changes are generated by IntelliJ IDEA. Normally, they are also applied through the IDE. Applying shelved changes outside IntelliJ IDEA is also possible but requires additional steps.

To save modifications to a new stash

1. On the main menu, choose **VCS | Git | Stash Changes**. The Stash dialog box opens.
2. Select the relevant Git root and make sure that the correct branch is checked out.
3. In the **Message** text box, describe the changes to be stashed.

   > To stash the local changes and bring the changes staged in the index to your working tree for examination and testing, select the **Keep Index** check box.

To apply a stash to the same branch

1. On the main menu, choose **VCS | Git | Unstash Changes**. The Unstash Changes dialog box opens.
2. Select the Git root where you want to apply a stash and make sure that the correct branch is checked out.
3. In the **Stashes** list, select the relevant stash.

   > Examine the stash descriptions and the names of the branches where specific stashes were created to find the stash you need.

4. Click the **View** button to open the **Paths affected in commit** dialog box and find out which files are affected in the selected stash.
5. Specify additional unstash options:
   - To have the selected stash removed from the list after it is applied, select the **Pop stash** check box.
   - To have the stashed index modifications applied, select the **Reinstate Index** check box.

     > This operation may fail if you have conflicts. This happens because conflicts are stored in the index, where you can no longer apply the changes as they were originally.

To create a new branch on the basis of a stash

1. On the main menu, choose **VCS | Git | Unstash Changes**. The Unstash Changes dialog box opens.
2. Select the Git root where you want to apply a stash.
3. In the **Stashes** list, select the relevant stash.
4. In the **As new branch** text box, type the name of the new branch to be created.

To remove one or several available stashes

1. On the main menu, choose **VCS | Git | Unstash Changes**. The Unstash Changes dialog box opens.
2. In the **Stashes** list, select the stashes to be removed and click the **Drop** button.

   > To remove all stashes from the list, click the **Clear** button.

See Also

Concepts:

- Shelved Changes
- Version Control with IntelliJ IDEA

Procedures:

- Shelving and Unshelving Changes
- Version Control with IntelliJ IDEA

Reference:

- Version Control Reference

- Stash Dialog
- Unstash Changes Dialog

External Links:

- http://schacon.github.com/git/git-stash.html

Web Resources:

- Developer Community