

1 Reset: Rare Manipulation of the Three Trees

1.1 Lecture

1.1.1 Refresher: What are The Three Trees?

- the Working Tree/Directory (WT)
 - Contains files you normally edit via tools that aren't Git
 - Place to experiment, figure out what you might want to add to your Repository
- the Staging Area/Index (SA)
 - Place where you prepare the next commit to your Repository
- HEAD
 - Reference to the checked-out branch or commit
 - Relates the other two states to the Repository (graph of commits that comprise the WT's history)

1.1.2 Everyday Manipulation of the Three Trees

Most of the time, you update the trees with each other's contents in only a few ways:

- Working Tree -> Staging Area: `git add`
- Staging Area -> HEAD: `git commit`

Moving HEAD Additionally, moving HEAD with `git switch` or `git checkout` causes a couple of automatic tree updates:

- HEAD -> Staging Area: `git switch/git checkout` (moving HEAD)
- Staging Area -> Working Tree: `git switch/git checkout` (moving HEAD)

NB: You can't move HEAD in this way if there are changes in the WT that would be overwritten by the move.

1.1.3 Rare Manipulation of the Three Trees with Reset

Reset can do up to three tree manipulations not present above, in sequence.

1. -> HEAD
 1. Moves the currently checked-out branch, which moves HEAD *without triggering the automatic updates shown above*
2. HEAD -> Staging Area
 1. Updates the Staging Area from HEAD's contents, *without triggering the automatic SA->WT update shown above*
3. Staging Area -> Working Tree
 1. Updates the Working Tree from the Staging Area's contents, *without regard for unstaged changes*

2. **NB:** The lack of regard for unstaged changes means that this step can destroy changes in your Working Tree that can't be recovered because they weren't in your Repository or Staging Area.

Each step in the sequence corresponds to a particular “flavor” of **reset**:

1. `git reset --soft` stops after the first
2. `git reset [--mixed]` stops after the second
3. `git reset --hard` stops after the third

1.2 Lab/Homework

Use the visualization tools from other labs to see how **HEAD**, the checked-out branch, the Staging Area, and the Working Tree change when using the various “flavors” of `git reset`.

1.2.1 Recommended Procedure

1. Set up a new Git Repository
2. Make a couple of trivial-ish commits
3. Use each “flavor” of `git reset` with each commit with and without additional Working Tree and Staging Area changes
 1. If you want to better understand how `git switch` and `git reset --hard` differ, just add `git switch` to the mix

NB: To checkout a commit with `git switch`, you'll need to use the `--detach` option to tell it that you're okay detaching **HEAD** from any branch.

1.3 Sources

- <https://git-scm.com/book/en/v2/Git-Tools-Reset-Demystified>