# Summary

Applies commits to a given revision in a linear fashion, as if the commits were made on the target thread of development.

Is rebase only applicable for merging different branches, or is it also useful for pulling in remote changes to a single branch.

# Usage

### Interactive rebase

git rebase -i [base commit on which to replay subsequent commits]

Use git rebase --i [last commit you want to keep]

git rebase --i HEAD~4

git rebase master

automatically process fixup / squash commits

git rebase -i --autosquash

reorders interactive rebase todo list such that commits marked as fixup and squash are ordered consecutively.

Editing todo list

squash

combine indicated commit with prior commit (retaining commit message of indicated commit)

fixup

combine indicated commit with prior commit (drops commit message of indicated commit)

Simplistic merge of rebase conflicts

In the context of a rebase...

git checkout --theirs core-integration/src/main/java/com/phlx/backoffice/web/service/ObjectFactory.java

Accept commit being applied on top of base

git checkout --ours core-integration/src/main/java/com/phlx/backoffice/web/service/ObjectFactory.java

Accept base commit

rebase uses recursive merge strategy by default

git rebase -m

git rebase -X ignore-all-space

Rebase using a different merge strategy

git rebase -s {strategy-name}

Apply a merge strategy option

git rebase -s {strategy-name} -X {strategy option}

Apply merge strategy options

pass options to rebase strategy using -X<option>

git pull --rebase -Xignore-all-space

Ignore whitespace as part of a rebase

git rebase --continue --ignore-whitespace

git rebase strategies

resolve

recursive

Stage changes before running rebase, then apply immediately afterwards

git rebase -i --autostash

# Concepts

Merge strategy

TODO

### Interactive rebase

Allows a series of commits to be applied on top of a given commit with user input as to how the commits should be applied.

Cannot be run when there are unstaged changes

# Terms

Forward-port

base commit

commit from which a branch diverges from a main branch of development

establishes a new baseline on which to base your changes

merge

combine one separate branch of development into another

merge strategy

merge conflict

git is unable to merge multiple lines of development without manual intervention. It isn’t willing to guess what your intent was with respect to the conflicting content.

fast-forward merge

merge in which the work you are attempting to merge into the current branch is directly upstream of the current branch. In this case, git simply moves the branch pointer to refer to the pointer representing the upstream branch.