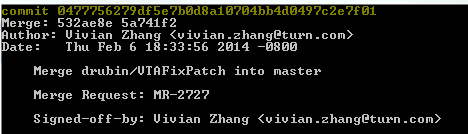
1. **How to edit the comments message of old merge commit?**

**Ans:**

1. Get the parent for the merge commit with “**git show <commit\_id>**”. It will return the parent id for that commit.

For example,



Here **532ae8e** is the parent of that commit.

1. Next run the rebase with --i (interactive) and –p (preserver-merge) options and parent id

For example, in our case it should be,

        git rebase --i –p **532ae8e**

1. This will open up the vim editor of git with the commits included for this parent. Next you need to mark a commit for editing. For that press I (insert) **before you can edit.**
2. Change the work **pick** into **reword** in front of the commit you would like to change. Once done press ESC and type **:wq.**
3. A new file will open up where you can edit and save you change.
4. In case of an issue with rebase you can just abort it with **git rebase –abort .**
5. **How to change commit message of recent commit?**

**Ans:**

git commit –amend –m “YOUR COMMIT MESSAGE”

**Example**:

git commit –amend -m "Code changed for // 72 characters only on this line

// New line Character

For Merging Issue on staging2 branch // description related to commited files

// new line character

Jira: DCR-181 // JIRA Ticket ID

Crucible: CRW-402 // CRUCIBLE Review ticket Id

Reviewed-by: Abc Xyz <[def@cybage.com](mailto:def@cybage.com)> //reviewer’s name and email

Signed-off-by: Abc Xyz <def@cybage.com>" // committer’s name and email

**Note:**

* Description for committed files is necessary; Commit message should not be like “Code Updated for” or “Code changed for” only. Please provide some description like why you changed files. What is the issue? etc.
* Check committers and reviewer’s name and email otherwise git won’t allow you to push the changes
* Crucible review ticket should be resolved or completed before pushing code to git repository

1. **How to search commit with specific message?**

**Ans:**

Execute below command

**git log --all --grep=**’YOUR SEARCH WORD’ // JIRA Ticket or description or name

1. **How to cherry-pick specific commit?**

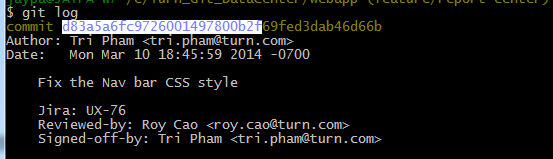
**Ans:**

Suppose I have two branch **Master** and **feature** branch .I want specific commit from master to feature branch then follow the below steps:

1. check out master branch and pull the latest changes

git pull --rebase origin master

1. copy the commit hash, which you want to apply on feature.



Commit Hash : d83a5a6fc

1. Go to feature branch and execute below command

git cherry-pick “COPIED COMMIT HASH”

Example:

Git cherry-pick d83a6a6fc

1. Execute git log and check whether your commit is there or not

(it will be there)

1. Push your changes

git push origin feature

1. **How to get the list of files committed in specific commit?**

**Ans:**

Execute below command

**git show –name-only** <commit#>

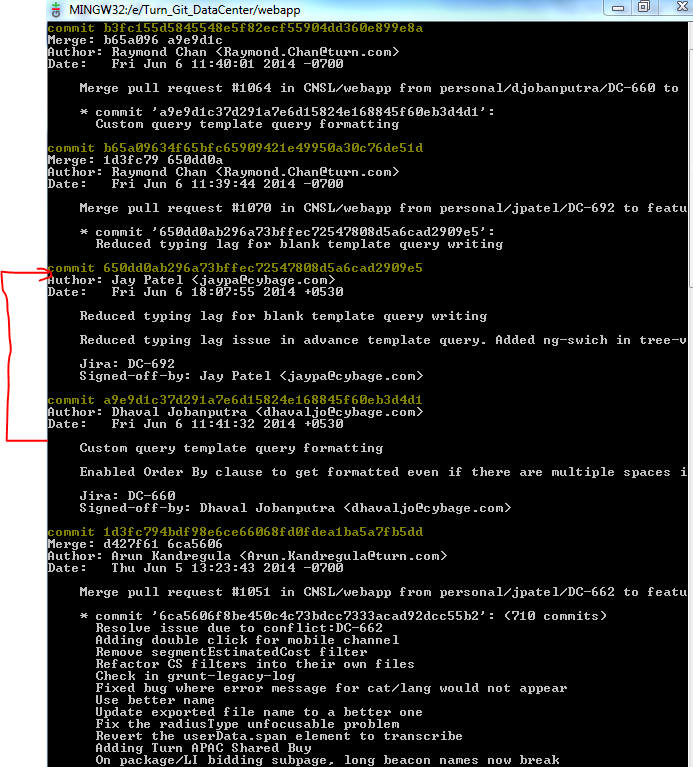
1. **How to delete multiple branch except feature or master branch?**

**Ans:**

**Execute :** git branch | grep -v "master" | xargs git branch –D

1. **How to squash/merge multiple commits in one commit?**

**Ans:**

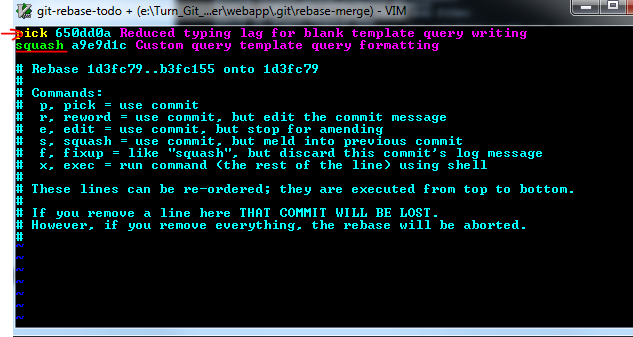


From above screen shot, I want to squash commit **a9e9d1c37** into **650dd0ab2.**

Find parent id of commits which we want to squash into one commit. Parent ID for commit **a9e9d1c37** is **1d3fc794.**

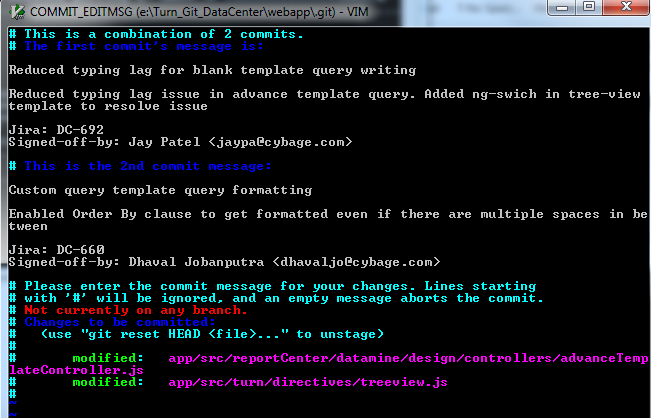
* **Execute : git rebase –I 1d3fc794**

It will open one editor in which you will get commits you want to squash. Mark squash against commits you want to squash. Find below screen shot

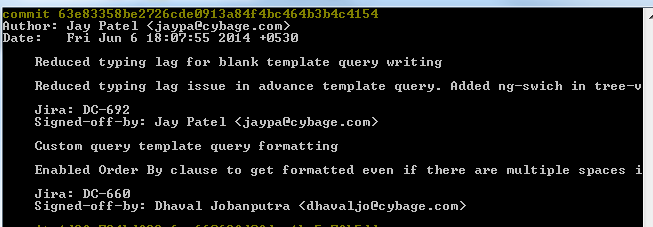


We want to squash a9e9d1c into 650dd0a so write squash before a9e9d1c instead of Pick and save and exit from editor with **“:wq”**

* + After saving git will open editor which shows commit messages of all the commits which we want to squash. You can modify those message and then save and exit from editor with “**:wq”**



* Your commits now squashed into one commit
* You can check with git log command.



This is how you can squash your commits. If you don’t want to preserve commit messages then you can mark commit as “**fixup**” instead of “**squash**”.

**MERGE COMMIT CAN’T BE SQUASHED**

1. **How to rename your local as well as remote branch?**

**Ans:**

If by mistake you wrongly named your branch and you pushed the code. Now if you want to rename your branch then follow bellow steps.

* git branch -m old-branch-name new-branch-name

**example:**

git branch –m personal/jatel/demo-branch personal/jpatel/demo-branch

**#delete remote branch with old name**

* git push origin :old-branch-name

**example:**

git push origin :personal/jatel/demo-branch

**#create remote renamed branch and setup remote branch tracking (-u flag)**

* git push -u origin new-branch-name

**example :**

git push –u origin personal/jpatel/demo-branch