

Git: To the Next Level (With a GUI)



Xavier Morera

PASSIONATE ABOUT TEACHING

@xmorera www.xaviermorera.com



“Small commits
allow for big wins”



Cherry Pick

Really useful command

Move single(or specific) commit(s)

From one branch to another

Why not a merge?

- You may not want everything
- Incompatibilities
- Specific feature in out of date branch



> git cherry-pick



Demo



Cherry Pick



Blame

Helps determine who made changes

- And why

History tells you files and changes

Last modification for each line

Revision and author

Known as annotate in TFS

Blame: Documentation/Facets.md			
Line	Author	Commit	Content
138	xaviermorera	f6b5aee8	- hard drive (2)
139	xaviermorera	f6b5aee8	- monitor (2)
140	xaviermorera	f6b5aee8	- search (2)
141	xaviermorera	f6b5aee8	- software (2)
142	xaviermorera	f6b5aee8	- camera (1)
143	xaviermorera	f6b5aee8	- copier (1)
144	xaviermorera	f6b5aee8	- multifunction printer (1)
145	xaviermorera	f6b5aee8	- music (1)
146	xaviermorera	f6b5aee8	- printer (1)
147	xaviermorera	f6b5aee8	- scanner (1)
148	xaviermorera	f6b5aee8	Pivot: instock with value false. Child Pivots:
149	xaviermorera	f6b5aee8	- electronics (4)
150	xaviermorera	f6b5aee8	- connector (2)
151	xaviermorera	f6b5aee8	- graphics card (2)
152	xaviermorera	f6b5aee8	'''
153	xmorera	5ee57262	
154	xmorera	5ee57262	TODO: Need to explain scenarios
155	xaviermorera	f6b5aee8	
156	xaviermorera	f6b5aee8	Additional information to be found in:
157	mausch	a519cb53	http://wiki.apache.org/solr/HierarchicalFaceting#Pivot_Facets
158	mausch	a519cb53	
159	mausch	a519cb53	http://wiki.apache.org/solr/SimpleFacetParameters#Pivot_.28ie_De
160	mausch	a519cb53	



Search

File Status

☒ Working Copy

Branches

master

NewDocumentation

pivot-fields

TestDocumentation

Tags

Remotes

origin

0.3.x

0.4.x

cursormark

DataImportHandler

debug-explain

document-model

ExtractingRequestHand

HEAD

javabin

linq

master

mlthandler

pivot-fields

solrcoreadmin-ioc

Stashes

On TestDocumentation: CI

Pivot

Search: Commit Messages

Commit Messages

File Changes

Authors

Description	Date	Author
Merge pull request #196 from xaviermorera/patch-4	31 May 2015 8:16	Mauricio
Including pivot facets information	7 May 2015 10:44	xaviermorera
Breaking change: enforce at least two fields in facet pivot. Avoid manual concatenation of fields.	5 Feb 2013 0:04	mausch <mauricio
removed duplicate pivot query serialization	1 Jun 2011 17:21	mausch <mauricio
Added Pivot and Grouping support for SOLR 4.0	1 Jun 2011 8:33	Klaas Vlietstra <kla

Sorted by path

Search

Commit: b60267ef2ee7e8c09ea97eec487ed8f0a5485774 [b60267e]

Parents: 7e145bad3c

Author: mausch <mauricioscheffer@gmail.com>

Date: Tuesday, February 5, 2013 12:04:08 AM

Labels: origin/pivot-fields, pivot-fields

Breaking change: enforce at least two fields in facet pivot.
Avoid manual concatenation of fields.

SolrNet.Tests/SolrFacetPivotQueryTest.cs

SolrNet/Impl/FacetQuerySerializers/SolrFacetPivotQuerySerializer.cs

SolrNet/PivotFields.cs

SolrNet/SolrFacetPivotQuery.cs

SolrNet/SolrNet.csproj

SolrNet.Tests/SolrFacetPivotQueryTest.cs

Reverse hunks

10 10

11 11

12 12

13 13

14 14

15 15

16 16

[Test]

public void SinglePivotTest() {

var q = new SolrFacetPivotQuery {

- Fields = new[] { "manu_exact", "inStock"},

+ Fields = new[] { new PivotFields("manu_exact", "inStock")},

MinCount = 1

};

22 22

23 23

24 24

25 25

26 26

27 27

28 28

[Test]

public void SinglePivotTestWithoutMinCount() {

var q = new SolrFacetPivotQuery {

- Fields = new[] { "manu_exact", "inStock"}

+ Fields = new[] { new PivotFields("manu_exact", "inStock")}

};

var r = serializer.Serialize(q);

35 35

36 36

37 37

[Test]

public void MultiplePivotTest() {

var q = new SolrFacetPivotQuery {

1 | TestDocumentation

Atlassian

Demo



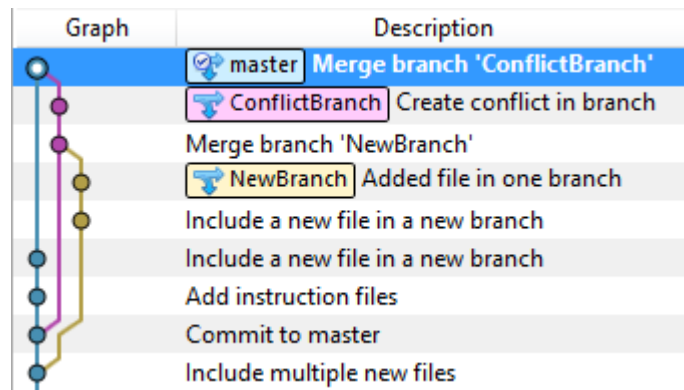
Blame and Search



Merging vs. Rebasing

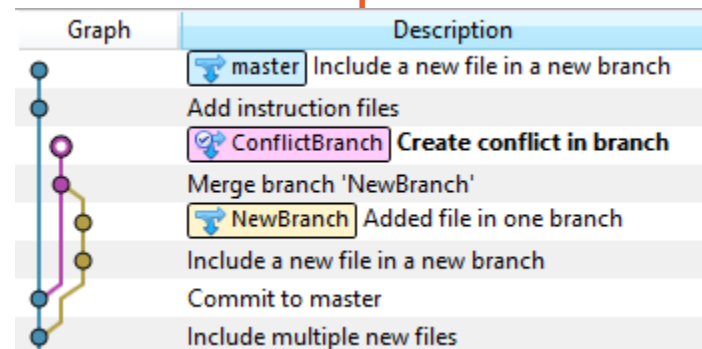
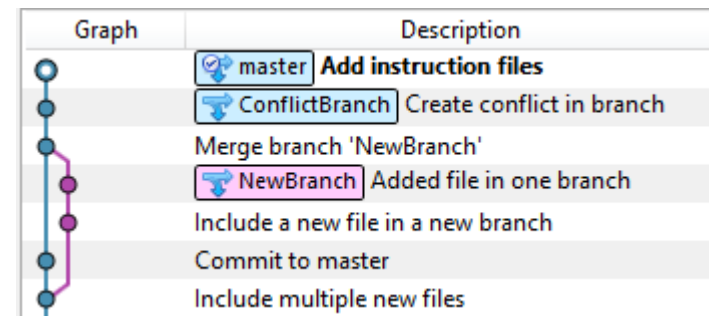
Merging

Brings two branches together



Rebasing

Applies commits on top linearly



Golden Rule of Rebasing

When NOT to do it: public branches

Rebase master into your feature branch

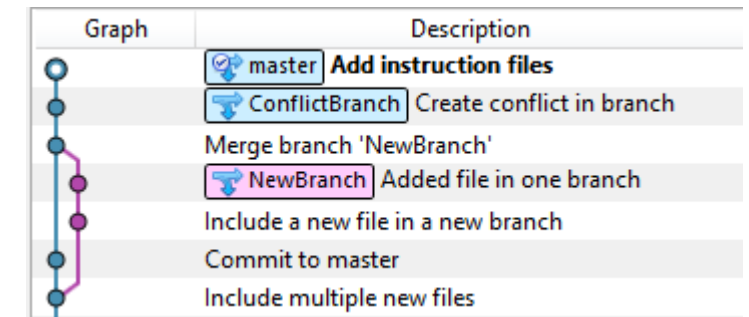
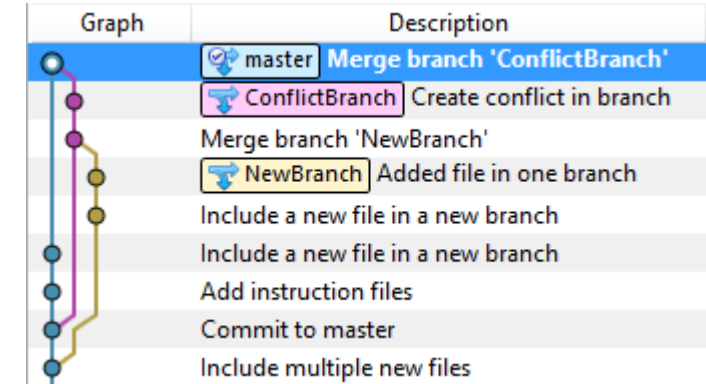
Happens in your repository

Confusion for everyone

If you insist: most likely force push required

Always think

- “Is anyone else looking at this branch?”



Merging vs. Rebasing

Pros

Simple to understand
Separate Commits (--no-ff)
Original Context

Merging

Need: parallel work in same branch
No useful historic purpose
Creates clutter

Cons

Pros

Simplifies history (linear)
Intuitive and clutter free way
Combine commits from multiple developers

Rebasing

Slightly more complex (conflicts)
Confusing if you push your commits

Cons



Demo



Merging vs. Rebasing



Fast Forward Merge

Special case

Branch... Work...

Merge

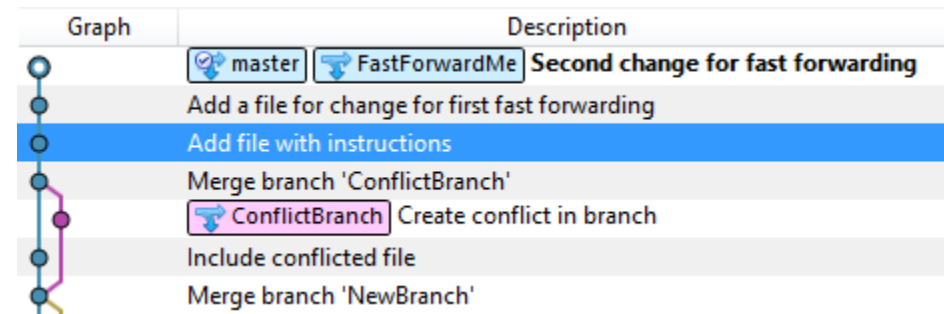
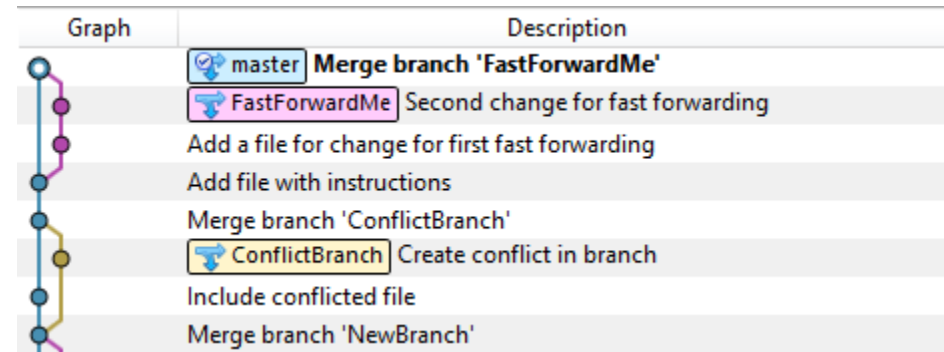
No commits in master?

Fast forward merge!

No merge commit

Just like rebase

[Optional]



Demo



Fast Forward Merge



Understanding Tags

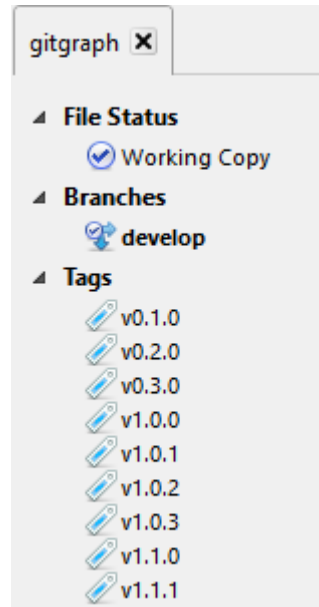
Mark important points in history

Typically releases (not limited to)

Lightweight

Like an immutable branch

Pointer to specific commit



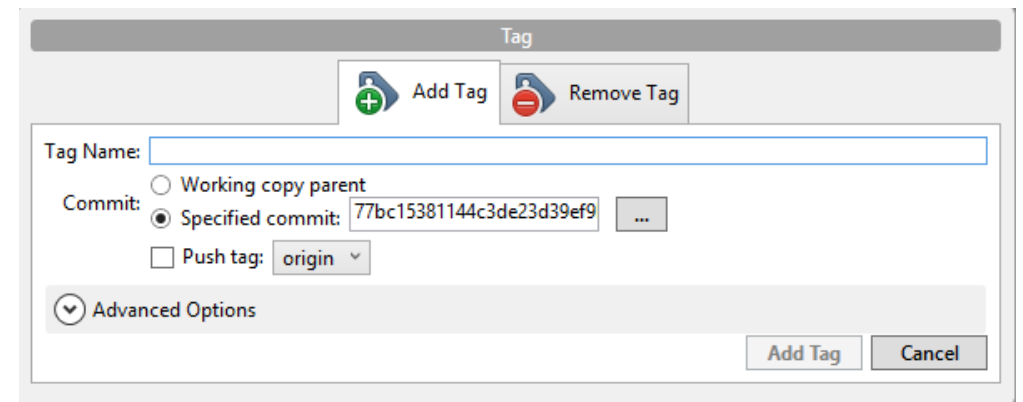
Annotated

Full objects

SHA1, tagger name, email, date, message

Can be signed and verified (GPG)

Recommended method

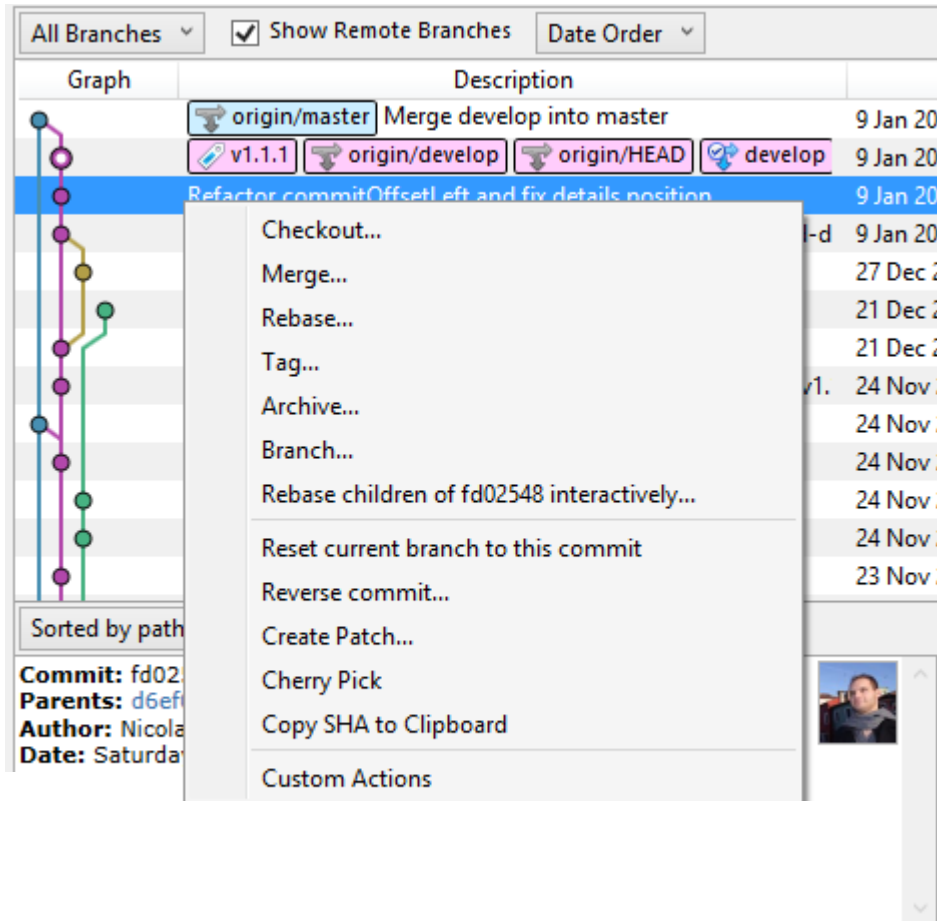


> git tag

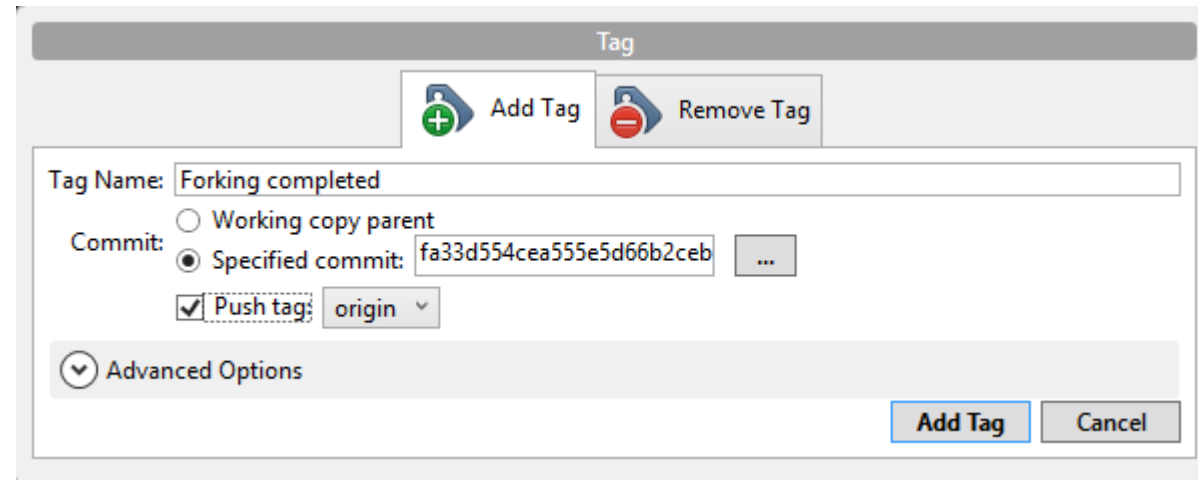


Understanding Tags

Tag Anywhere



Push Tags



> git tag



Demo



Tagging



Detached Head

Git → Direct acyclic graph of commits

- Fancy name for a tree of commits

Work in Branches

- Checkout

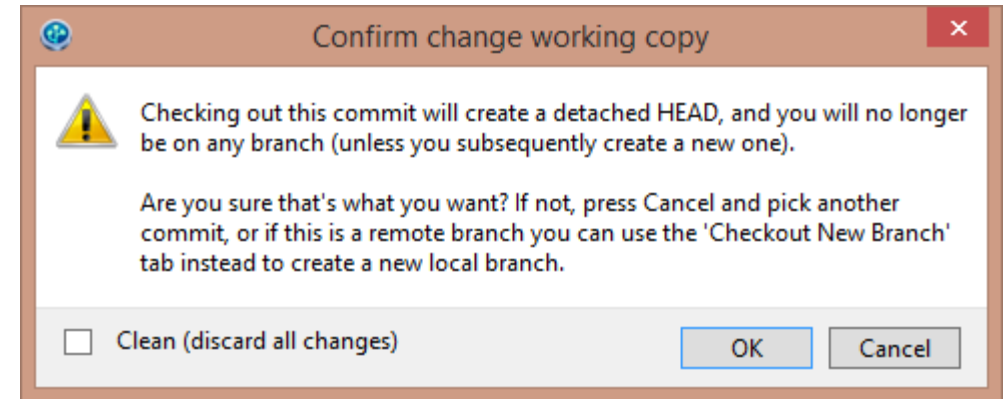
Checkout a commit?

No Head

This is called a Detached Head

And it will give you a headache

There is a way



Demo



Detached Head



Rewriting History

Git is designed to never lose a change

But...

It is also really powerful

Rewrite history

Usually only on your own to prepare a push

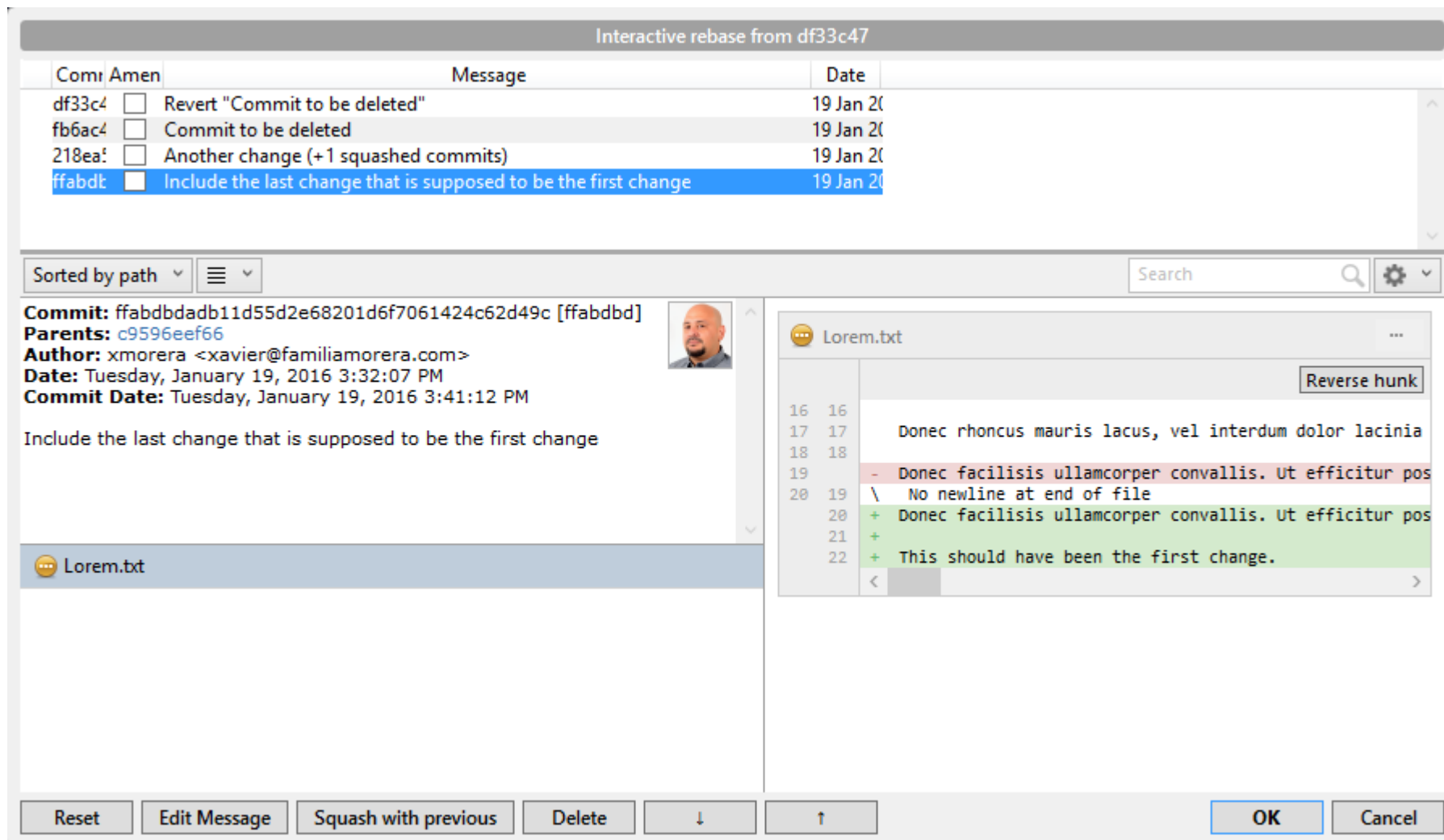
Rebase interactively

Disclaimers:

- Not recommended as normal action
- Don't amend public commits



Rebase Interactively



> git rebase -i



Reordering Commits

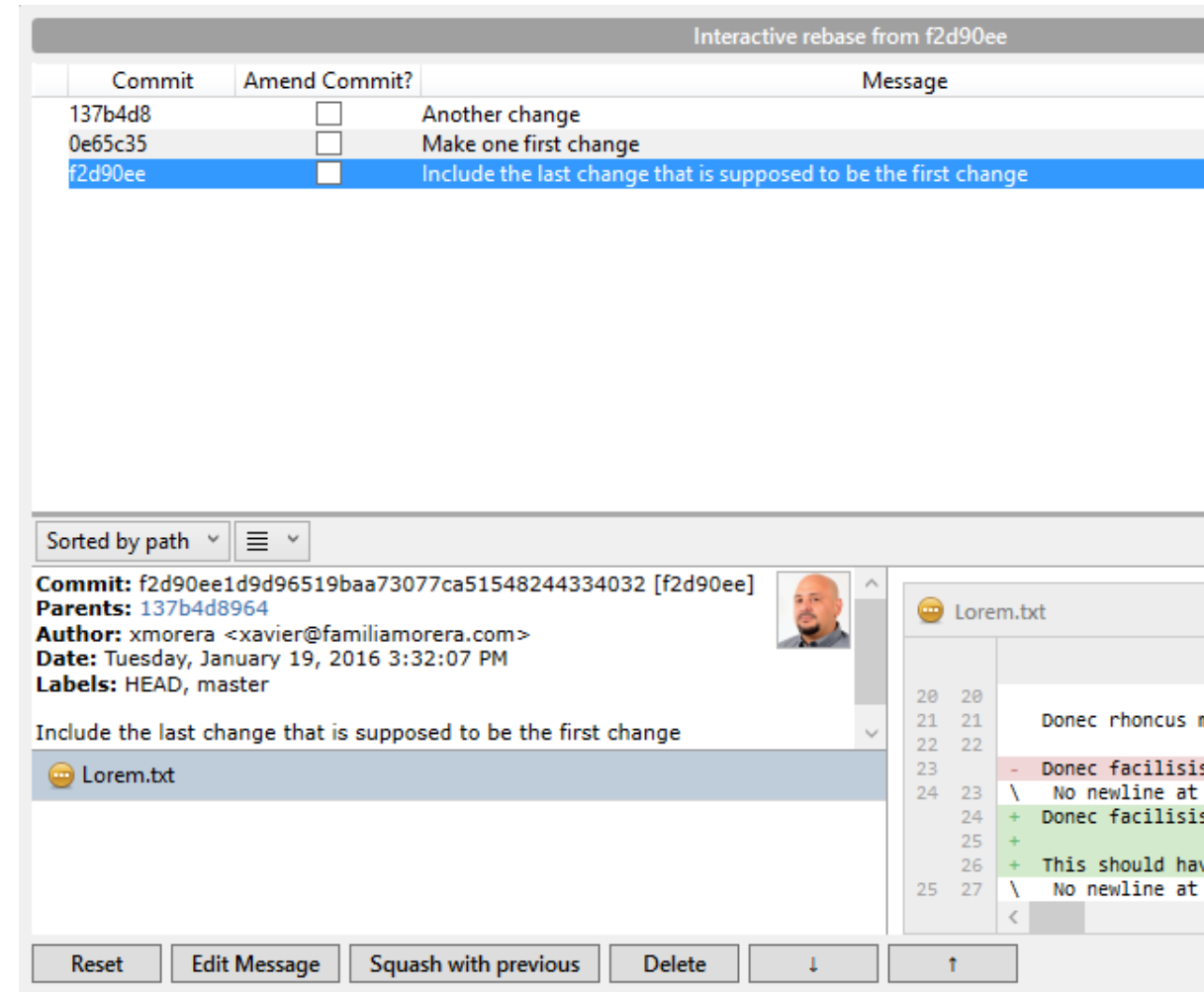
May need to reorder your commits

SourceTree → drag and drop

Not done

Until you click OK

And get confirmation



> git rebase -i

Demo



Reordering Commits



Editing Commit Comment

May need to change commit comment

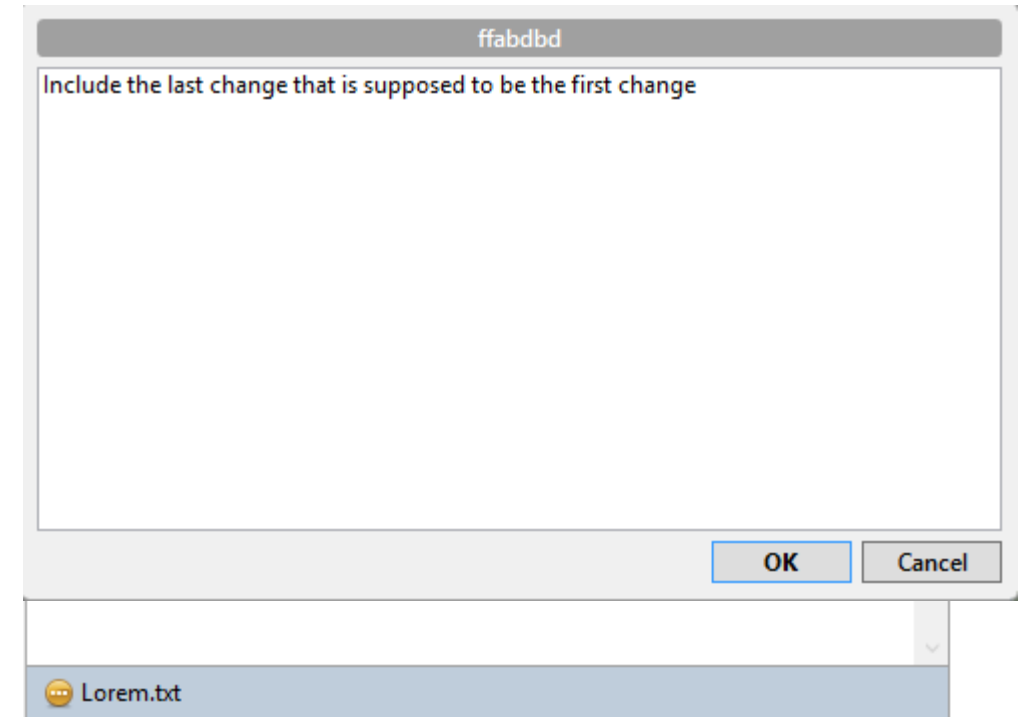
Rewriting history

Dead easy with SourceTree

Careful: on changes not pushed yet

Tip:

- Change author is possible too!
- git filter-branch



Amend Commit

Besides change comment and order

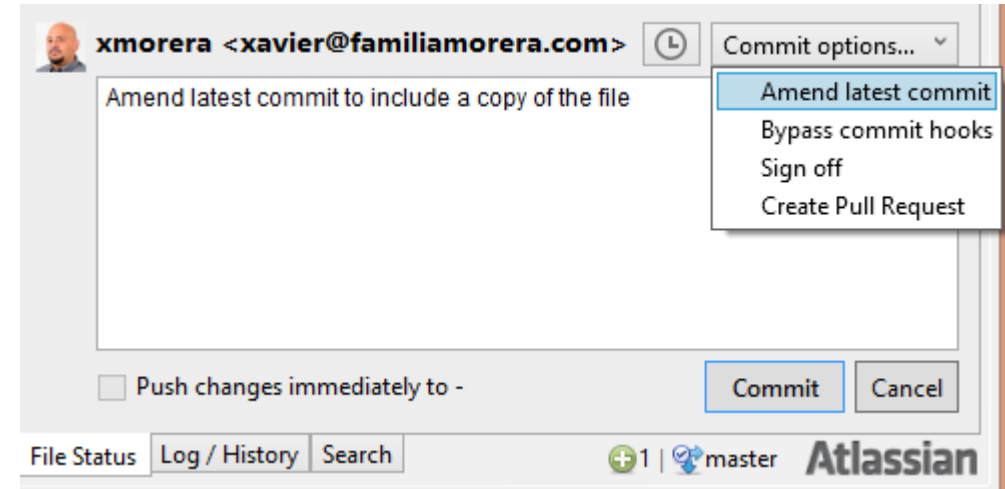
Change content (snapshot)

- Adding or removing

Disclaimer: Changes SHA1

Select Amend latest commit

From commit options



> git commit --amend



Demo



Editing and Amending Commits



Squash Commits

“Commit early, commit often”

End up with lots of commits

Lot of them being WIP (work in progress)

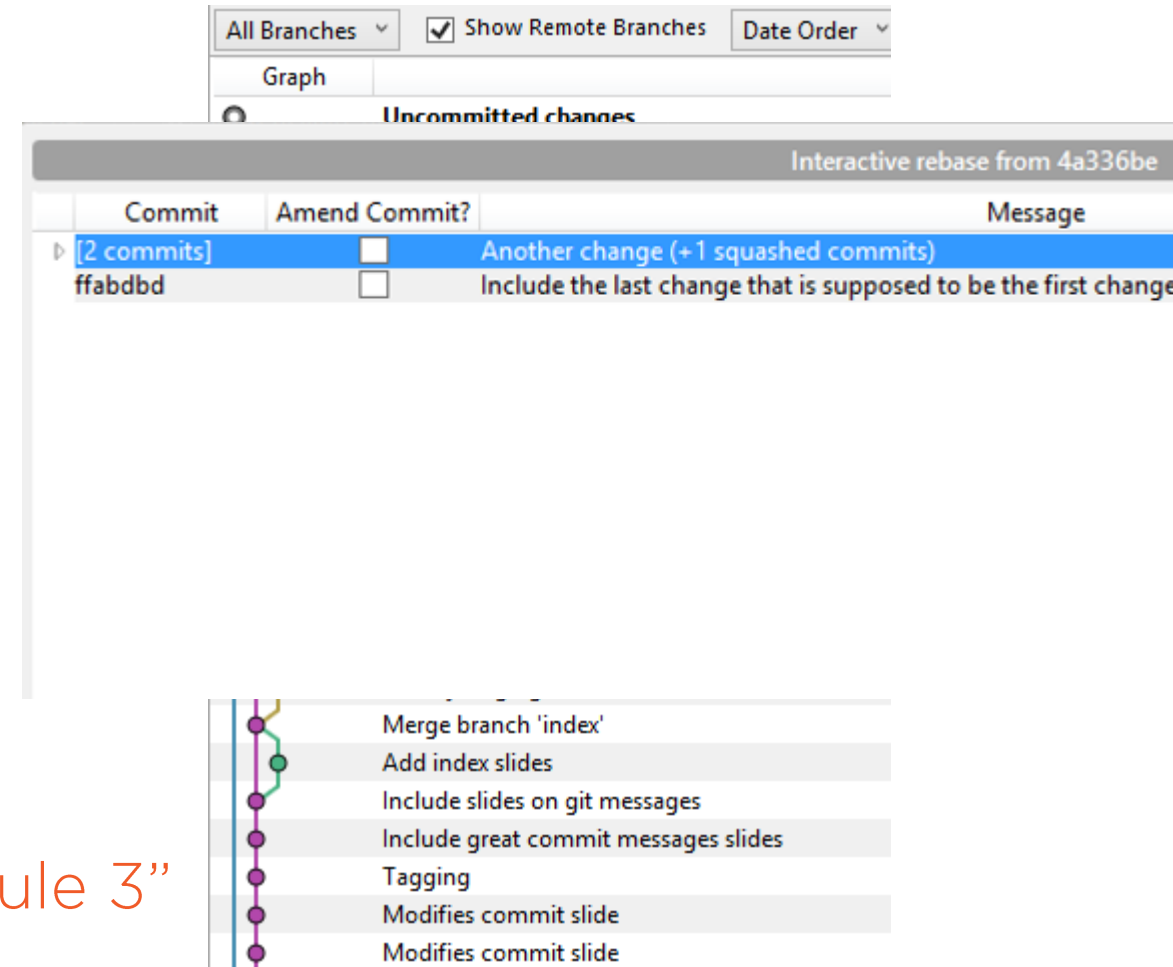
Squash multiple commits

Rebase interactively

Clean history

Overview as commit message

- “Includes completed slides for module 3”



Debate: To squash or not to squash...



Squash Commit Messages

Not about file revisions

It is about “Overview”

What bug is fixed?

What feature done?



Demo



Squash Commits



Deleting a Commit

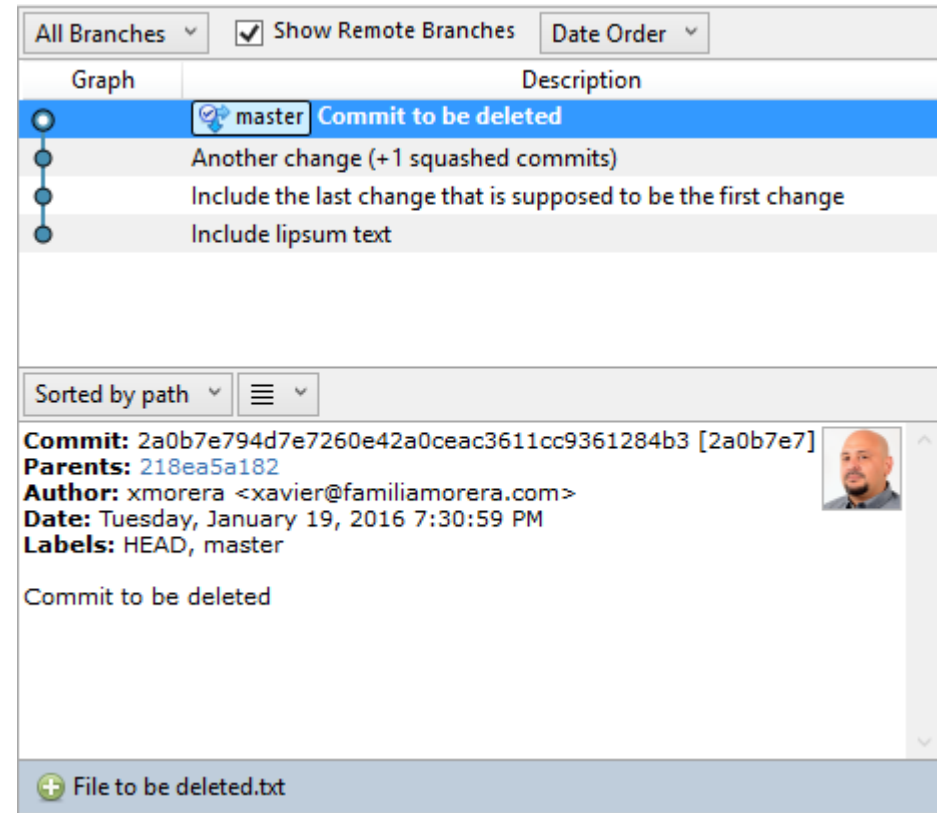
Preferred to only add in Git

But delete is possible

Careful

Try to avoid deleting

Rebase interactively



Deleting a Commit - Reset

Reset to Commit...

Are you sure you want to move the branch pointer?

Reset branch: master

To commit: 218ea5a: Another change (+1 squashed commits)

Using mode: **Mixed - keep working copy but reset index**

- Mixed - keep working copy but reset index
- Soft - keep all local changes
- Mixed - keep working copy but reset index
- Hard - discard all working copy changes

File Status

All Branches ☒ Show Remote Branches Date Order

Graph

Working Copy

Branches

master

Tags

Remotes

Commit: 218ea5a182c51f552c673eddf29b058df68cc269 [218ea5a]

Parents: ffabdbdadb

Author: xmorera <xavier@familiariorera.com>

Date: Tuesday, January 19, 2016 3:31:06 PM

Commit Date: Tuesday, January 19, 2016 6:12:39 PM

Another change (+1 squashed commits)

Squashed commits:

[6b098ba] Make one first change

Sorted by path

Search

Reverse hunk

1 + Lorem ipsum dolor sit amet, consectetur adipiscing elit. In suscipit orci et massa iaculis, e

2 +

3 + Make one first change

4 +

5 + Donec vitae laoreet quam. Pellentesque elit sem, placerat ac erat id, scelerisque rutrum nunc

6 +

7 + Duis gravida, lectus sed malesuada consequat, turpis quam tristique purus, at bibendum dolor

8 +

9 + Make another change

10 +

11 + Donec ornare massa at sapien malesuada dignissim. Nam at diam finibus, blandit orci eu, luctu

12 +

13 + Donec tincidunt, sem non iaculis tempor, lectus ex faucibus tellus, non sodales sem sem vitae

14 +

15 + Sed metus tellus, sagittis at tincidunt sed, mollis quis leo. Phasellus elementum nulla fring

16 +

17 + Curabitur et nunc vel nibh convallis feugiat. Aenean vulputate ligula ac molestie dictum. Do

18 +

19 + Pellentesque pellentesque pretium mollis. Aliquam faucibus diam ac aliquet pretium. Quisque

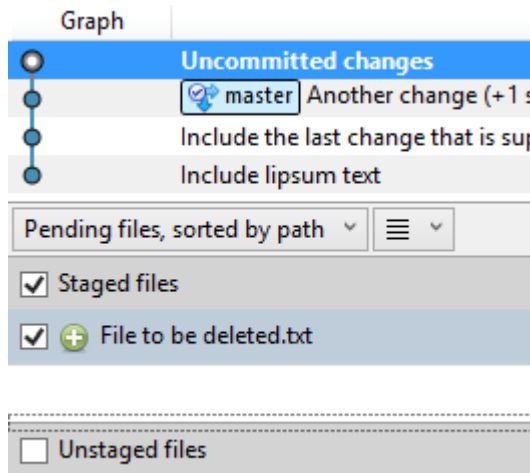
20 +

21 + Donec rhoncus mauris lacus, vel interdum dolor lacinia sit amet. Aenean ullamcorper vel dolor

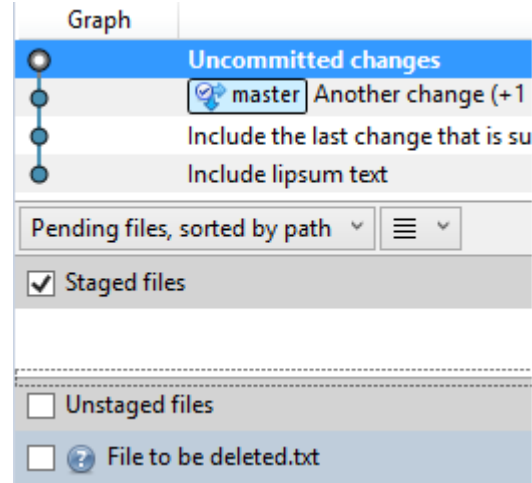
File Status Log / History Search

Clean master Atlassian

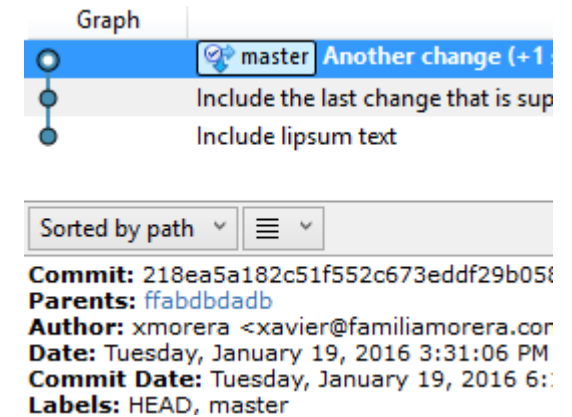
Deleting a Commit - Reset



Soft



Mixed



Hard



Demo



Deleting Commits



Reverse Commit

Mistakes can be made

- Let's avoid deleting commits

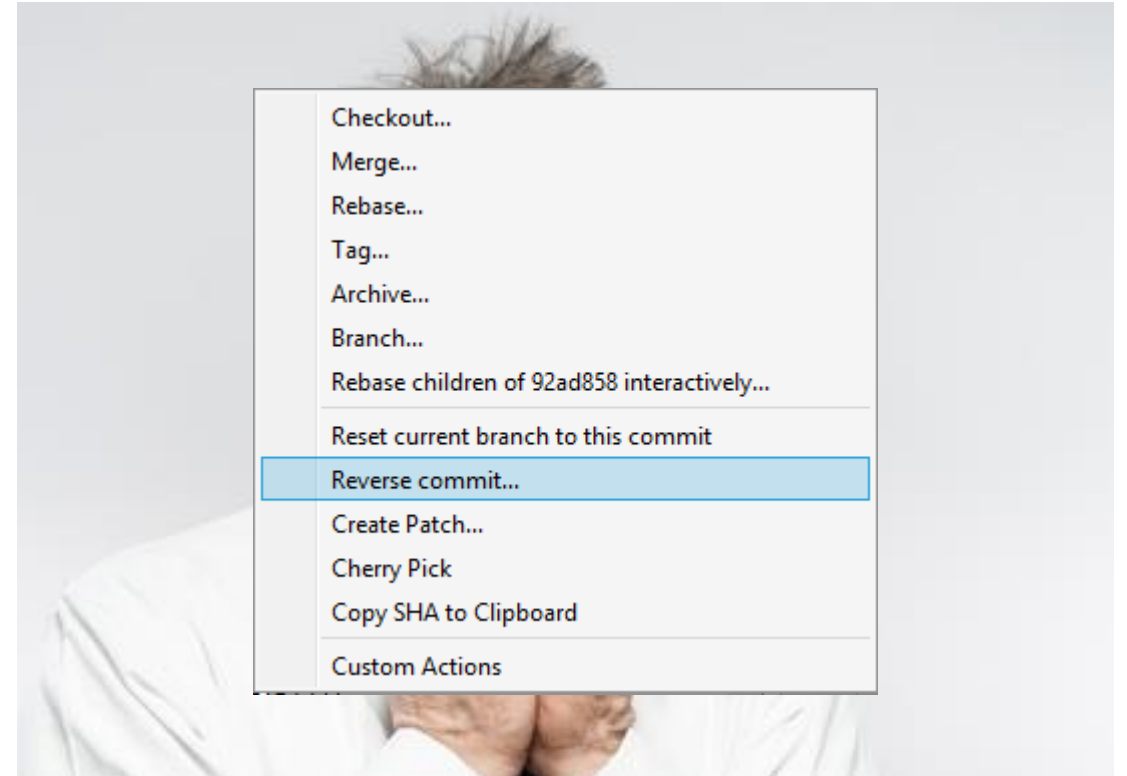
Reverse commit instead!

Revert

- Safe method

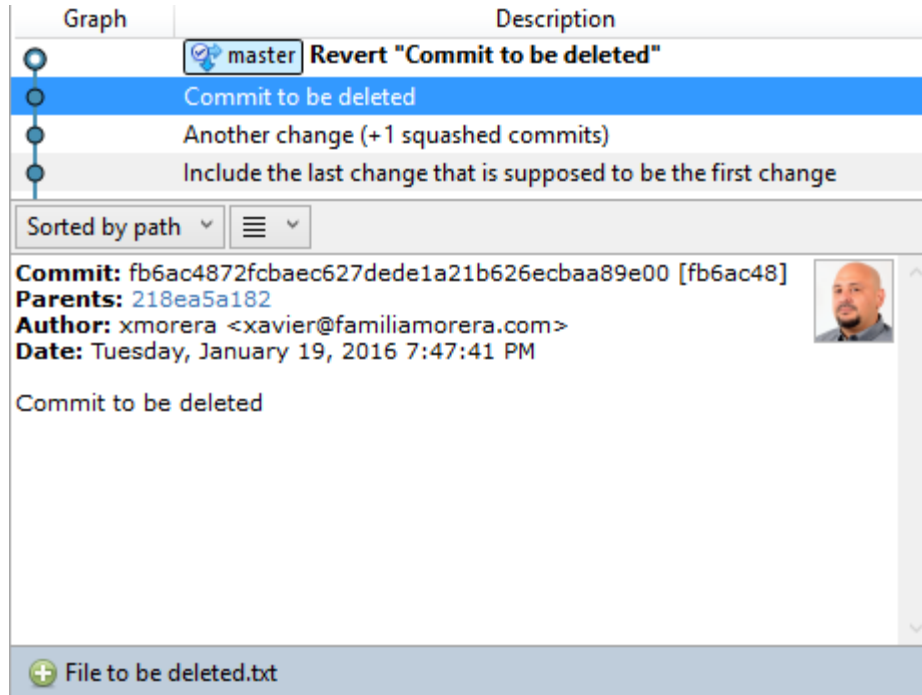
Undo committed snapshot

By creating a new commit



Reverse Commit

Before



The screenshot shows a Git commit history interface. The 'Graph' tab is selected, showing a vertical sequence of four commits. The top commit, labeled 'master', is 'Revert "Commit to be deleted"'. Below it are 'Commit to be deleted', 'Another change (+1 squashed commits)', and 'Include the last change that is supposed to be the first change'. The commit details for the top commit are: Commit: fb6ac4872fcbaec627dede1a21b626ecbaa89e00 [fb6ac48], Parents: 218ea5a182, Author: xmorera <xavier@familiariorera.com>, Date: Tuesday, January 19, 2016 7:47:41 PM. The commit message is 'Commit to be deleted'. At the bottom, a file 'File to be deleted.txt' is listed with a green plus icon.

Graph	Description
	master Revert "Commit to be deleted"
	Commit to be deleted
	Another change (+1 squashed commits)
	Include the last change that is supposed to be the first change

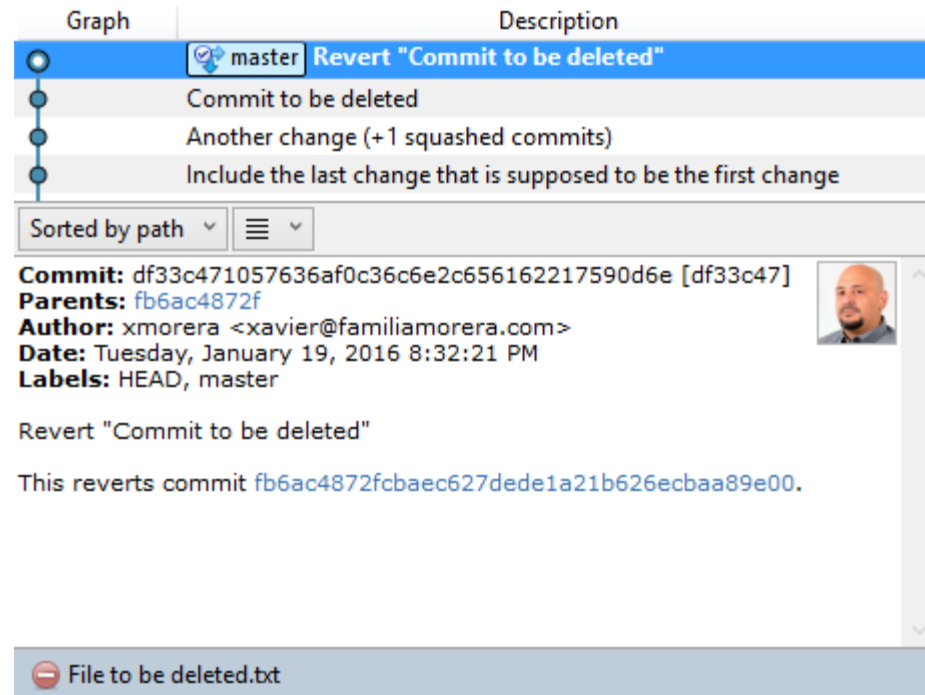
Sorted by path

Commit: fb6ac4872fcbaec627dede1a21b626ecbaa89e00 [fb6ac48]
Parents: 218ea5a182
Author: xmorera <xavier@familiariorera.com>
Date: Tuesday, January 19, 2016 7:47:41 PM

Commit to be deleted

File to be deleted.txt

After



The screenshot shows the same Git commit history interface after a reverse commit. The 'Graph' tab is selected, showing a vertical sequence of four commits. The top commit, labeled 'master', is 'Revert "Commit to be deleted"'. Below it are 'Commit to be deleted', 'Another change (+1 squashed commits)', and 'Include the last change that is supposed to be the first change'. The commit details for the top commit are: Commit: df33c471057636af0c36c6e2c656162217590d6e [df33c47], Parents: fb6ac4872f, Author: xmorera <xavier@familiariorera.com>, Date: Tuesday, January 19, 2016 8:32:21 PM, Labels: HEAD, master. The commit message is 'Revert "Commit to be deleted"'. Below the message, it says 'This reverts commit fb6ac4872fcbaec627dede1a21b626ecbaa89e00.'. At the bottom, the file 'File to be deleted.txt' is listed with a red minus icon.

Graph	Description
	master Revert "Commit to be deleted"
	Commit to be deleted
	Another change (+1 squashed commits)
	Include the last change that is supposed to be the first change

Sorted by path

Commit: df33c471057636af0c36c6e2c656162217590d6e [df33c47]
Parents: fb6ac4872f
Author: xmorera <xavier@familiariorera.com>
Date: Tuesday, January 19, 2016 8:32:21 PM
Labels: HEAD, master

Revert "Commit to be deleted"

This reverts commit fb6ac4872fcbaec627dede1a21b626ecbaa89e00.

File to be deleted.txt



Demo



Reverse Commit



Recover Deleted Commit

You HARD deleted a commit

Panic!

Not all is lost

No need to start crying!

Dangling commit

Open terminal and let's get advanced

You just need the SHA1 to get it back

- fb6ac4872fcbaec627dede1a21b626ecbaa89e00 [fb6ac48]



> git reflog

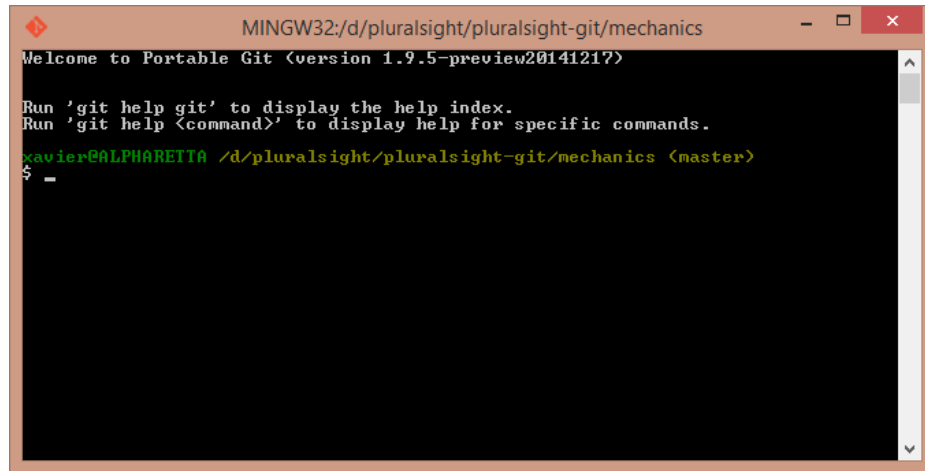
> git fsck --lost-found



Recover Deleted Commit

SourceTree Terminal

Works

A screenshot of a SourceTree terminal window. The title bar reads 'MINGW32:/d/pluralsight/pluralsight-git/mechanics'. The terminal text shows a welcome message for Portable Git (version 1.9.5-preview20141217), instructions on how to use 'git help', and a command prompt where the user 'xavier@ALPHARETTA' has entered a path and '[master]'.

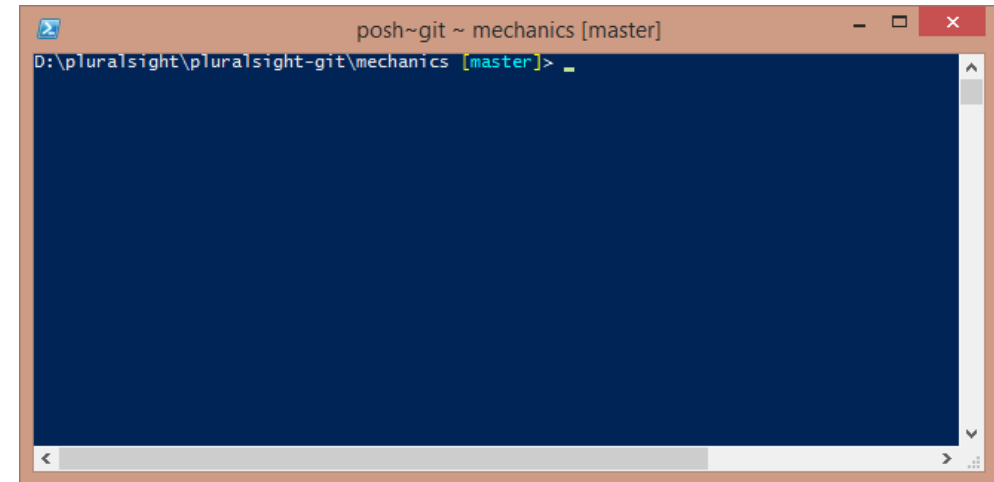
```
MINGW32:/d/pluralsight/pluralsight-git/mechanics
Welcome to Portable Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

xavier@ALPHARETTA /d/pluralsight/pluralsight-git/mechanics [master]
$ _
```

Git Shell

My personal preference (Powershell)

A screenshot of a Git Shell window. The title bar reads 'posh-git ~ mechanics [master]'. The terminal shows the path 'D:\pluralsight\pluralsight-git\mechanics' followed by '[master]' and a prompt character '>'.

```
posh-git ~ mechanics [master]
D:\pluralsight\pluralsight-git\mechanics [master]> _
```

<https://git-scm.com/>



Demo



Recovering Deleted Commits



Custom Actions

Not everything available in SourceTree

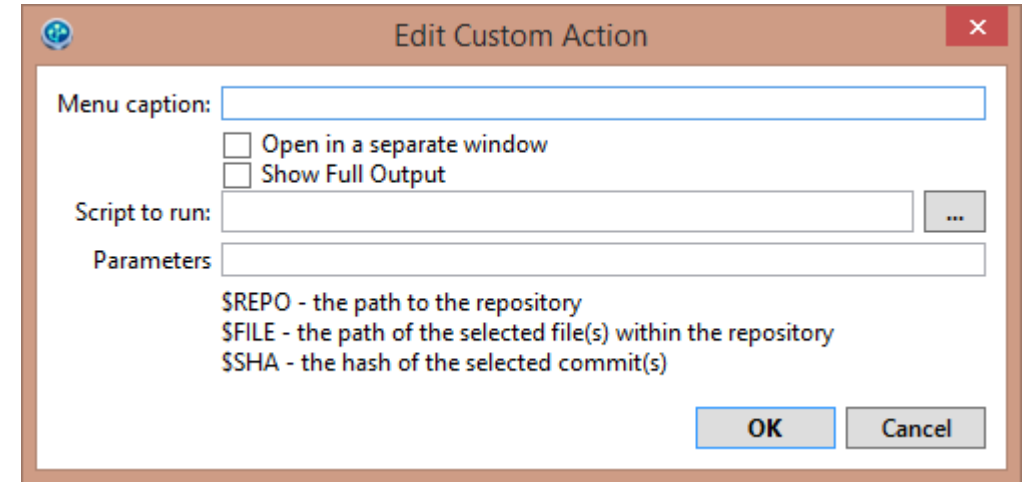
Extend range of actions

Creating scripts or commands

Defined variables

PowerShell, bash or Git command

Harness full power of Git



Demo



Custom Actions



Takeaway

Small Commits
Big Wins

Merging
Rebasing
Fast Forward Merge

History
Blame and Search

Tags

Rewriting History

Custom Actions

