

***Obliterate***

*or*

*Let's Pretend I Didn't Do That*

Julian Foad

WANdisco



# Overview

- Why obliterate?

Reasons fall into three categories
- How to obliterate?

Delete files or undo changes
- Alternatives

How it's done in CVS and P4  
How it's done in Svn now



# *Why?*

- Requested...
  - since pre-1.0
  - by many on mailing list over the years
  - by big companies & organisations
- Reasons stated...
  - hide secrets
  - undo accidental large additions
  - remove old data to reduce repos size
  - work flow needs only recent version(s)



# *Why?*

- Alternatives...

  - can do in CVS (crudely)

  - Perforce has “p4 obliterate”

  - Use authz to hide files

- Counter-arguments...

  - is not pure version control (find other ways)

  - “dangerous” if mis-used (audit trail)



# *Why? Hide a Secret*

- What happened?
  - committed a customer's private data in a company-wide repository,
  - committed copyrighted material to a public repository.
- What's needed?
  - Hide** that file first,  
(may require **swift** action),  
then plan the permanent fix.



# *Why? Recover Space*

- What happened?
  - added large, generated files,
  - imported to wrong repository,
  - split the repos,
  - a large project became redundant.
- What's needed?
  - Reclaim** server disk **space**.
  - Can be a **planned** event.



# Why? Latest Version

- Project structure:

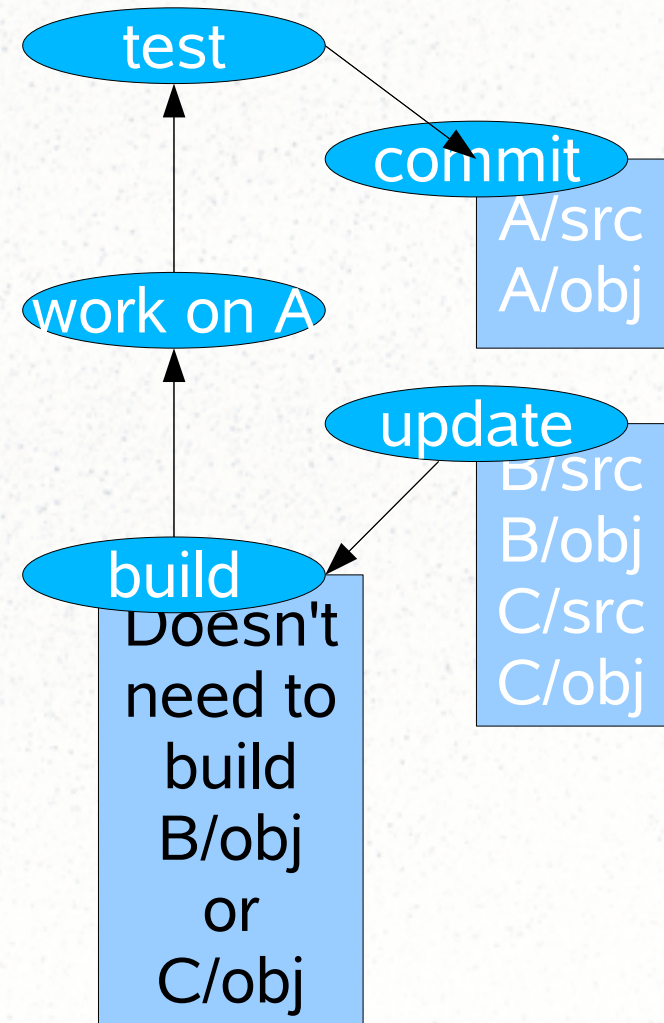
moduleA/

- src/...

- **obj/...**

moduleB/...

moduleC/...





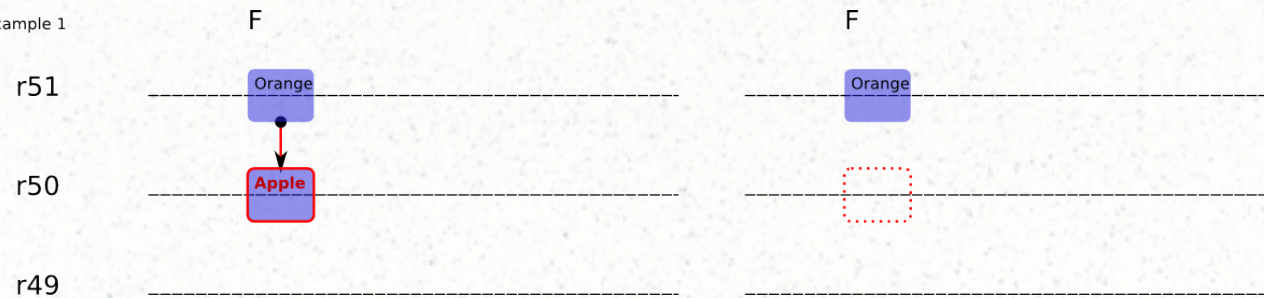
# *Why? Latest Version*

- What's happening?
  - frequently revising a large file
  - never need old versions of it
- What's needed?
  - configure to store only latest version
  - make “update” work without deltas
- Why?
  - convenience (could do outside svn)

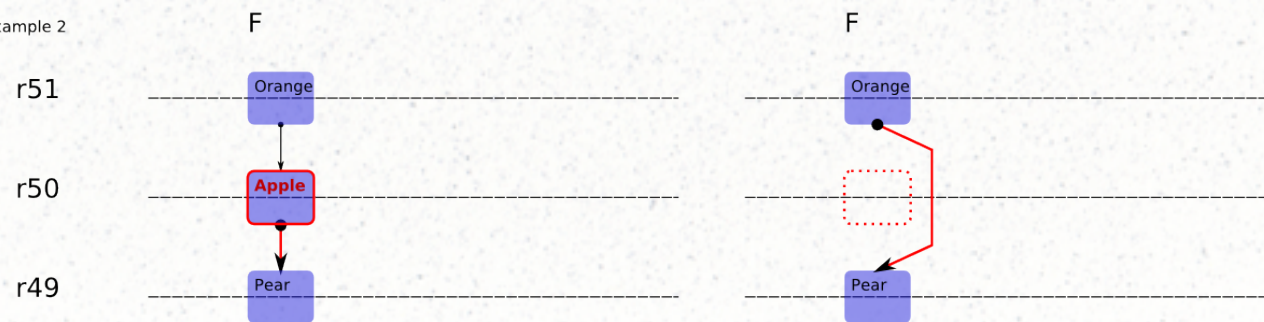


# How? Delete Files

Example 1



Example 2



Before Obliterating

After Obliterating

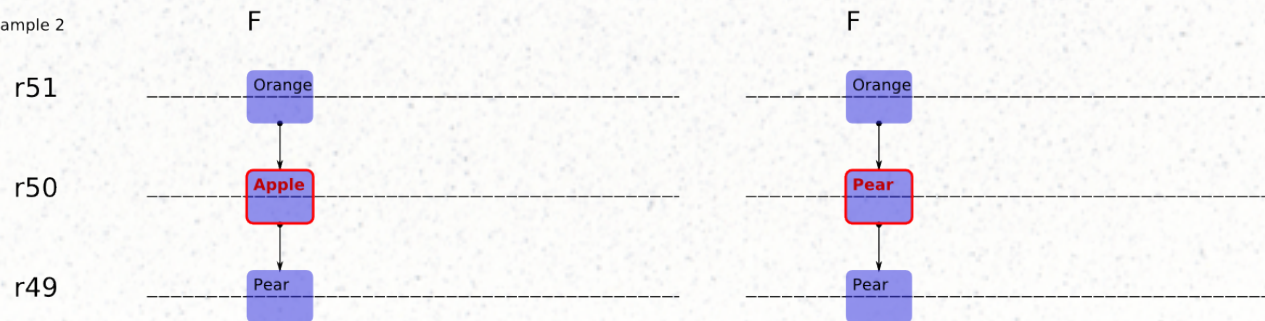


# How? Undo Changes

Example 1



Example 2



Before Obliterating

After Obliterating



## *What CVS/P4 Users Do*

- CVS: Delete the “,v” file
- Perforce: “p4 protect”
- Perforce: “p4 obliterate”
- Perforce: “p4 filetype” for latest N revs



# *What Svn Users Do*

- dump | svndumpfilter | load
- authz can hide paths



# *Non-reasons*

- These possible reasons to obliterate are not goals
  - to undo (roll back) a recent commit
  - to tidy up history after an unwanted change has been done and undone
- Documentation should steer users to best practices for these tasks



# ***Obliterate***

*or*

*Let's Pretend I Didn't Do That*

Julian Foad

WANDisco



## *Overview*

- Why obliterate?  
Reasons fall into three categories
- How to obliterate?  
Delete files or undo changes
- Alternatives  
How it's done in CVS and P4  
How it's done in Svn now



## *Why?*

- Requested...
  - since pre-1.0
  - by many on mailing list over the years
  - by big companies & organisations
- Reasons stated...
  - hide secrets
  - undo accidental large additions
  - remove old data to reduce repos size
  - work flow needs only recent version(s)<sup>3</sup>



## *Why?*

- Alternatives...
  - can do in CVS (crudely)
  - Perforce has “p4 obliterate”
  - Use authz to hide files
- Counter-arguments...
  - is not pure version control (find other ways)
  - “dangerous” if mis-used (audit trail)



## *Why? Hide a Secret*

- What happened?

committed a customer's private data in  
a company-wide repository,

committed copyrighted material to a  
public repository.

- What's needed?

**Hide** that file first,

(may require **swift** action),

then plan the permanent fix.

## *Why? Recover Space*

- What happened?
  - added large, generated files,
  - imported to wrong repository,
  - split the repos,
  - a large project became redundant.
- What's needed?
  - Reclaim** server disk **space**.
  - Can be a **planned** event.

6

- Even when correcting a mistake just made, can't assume that it's sufficient to delete just the most recent N revs of the whole repository, because the need arises most commonly in a busy repository.



## Why? Latest Version

- Project structure:

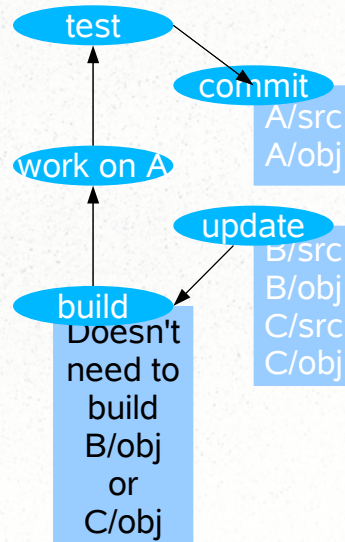
moduleA/

- src/...

- **obj/...**

moduleB/...

moduleC/...

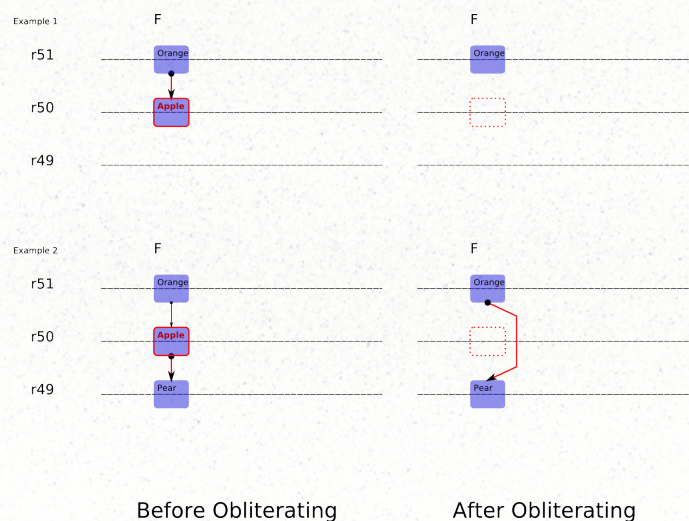


## ***Why? Latest Version***

- What's happening?
  - frequently revising a large file
  - never need old versions of it
- What's needed?
  - configure to store only latest version
  - make “update” work without deltas
- Why?
  - convenience (could do outside svn)



## How? Delete Files



9

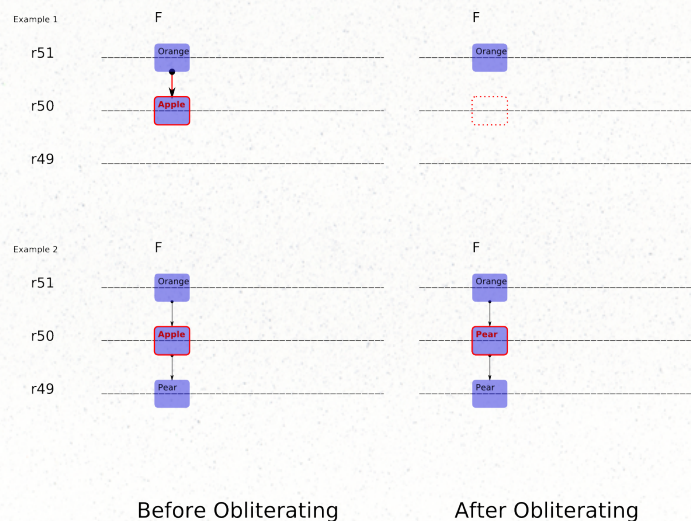
There are many possible ways in which history could be changed. This is one way, and on the next slide is another way. Both of these ways satisfy all of the use cases.

This scheme, here labelled “delete files”, is in fact illustrating deletion of an individual node-rev. The user can obliterate individual files (or directory trees) from individual revisions (or ranges of revisions).

This scheme shows history being re-connected from F@51 to F@49, whereas another option is not to re-connect the history in this way.

See [<notes/obliterate/fspec-dd1/>](#) in the source tree for details of this scheme.

## How? Undo Changes



10

This scheme, here labelled “undo changes”, differs from the “delete files” scheme in that it replaces an individual node-rev with the previous revision of that node, or deletes it if it did not exist in the previous revision.

See [<notes/obliterate/fspec-cc1/>](#) in the source tree for details of this scheme.



## *What CVS/P4 Users Do*

- CVS: Delete the “,v” file
- Perforce: “p4 protect”
- Perforce: “p4 obliterate”
- Perforce: “p4 filetype” for latest N revs

11

CVS administrators can remove an archive “,v” file to effectively delete the whole history of a particular file. (Recall that in CVS each file has its own history, version numbers and log messages.) It is not so easy to remove selected revisions or changes from a CVS archive.

Perforce administrators can hide files with “p4 protect”, delete file-revisions with “p4 obliterate”, and enable “last N versions only” on selected files with a “p4 filetype” setting.

## *What Svn Users Do*

- dump | svndumpfilter | load
- authz can hide paths

12

Subversion administrators can dump, filter and load a repository. Functionally, that satisfies most needs, but on a large repository it is much too slow for anything except a planned space-recovery. Authorization settings can be used to hide specific paths. (But not at specific revs? Need to check.)



## *Non-reasons*

- These possible reasons to obliterate are not goals
  - to undo (roll back) a recent commit
  - to tidy up history after an unwanted change has been done and undone
- Documentation should steer users to best practices for these tasks