

Subversion Obliterate, the forgotten feature

by Benny Bottema - Saturday, March 01, 2008

<http://www.bennybottema.com/2008/03/01/subversion-obliterate-the-forgotten-feature/>

Sometimes it is best to know forehand some of the more obscure features and limitations of a software product, before diving in. One such obscure limitation is that of the infamous Obliterate feature of the Version Control software, Subversion, that people only learn of when they discover it is not there. Will you need it? Most likely yes, at some point. Especially if you are dealing with a large codebase, you will want to be wary of Subversion, however great its existing feature are. Read on.

In a nutshell, Subversion is Open Source, therefor is free, it is time tested and is actively being developed for. For many, this alone is enough reason to use Subversion for their version control needs. It is the supposed successor to CVS and therefor boasts most of the features CVS had, better and more. According to the [feature list](#) on the Subversion homepage:

Most current CVS features.

- Subversion is meant to be a better CVS, so it has most of CVS's features. Generally, Subversion's interface to a particular feature is similar to CVS's, except where there's a compelling reason to do otherwise.

This sounds really good, and this is only one item of the list. Yet still, one feature is missing. Can you guess it?

Contents

- [1 The need to Obliterate](#)
- [2 The broken workaround](#)
- [3 So, after 712 years, where the hell is it?](#)
- [4 What to do now](#)

The need to Obliterate

Obliterate means 'really removing stuff from the repository, seriously'. When you check in files and at some point decide you need it removed, normally you would go and simply use the *delete* command for

Subversion. Either manually by command line, or through some automated tool such as [TortoiseSVN](#). However, you don't really delete it from the repository, it is only *marked* as deleted. Now why would you want to completely obliterate anything from a repository including version history?

There are a couple of situations that might arise:

- You accidentally [checked in sensitive information](#) and need it removed
- You are [running a large codebase](#) and need to archive old repository entries to [clean up to save drivespace](#)
- You need to split up a repository in several others for some reason, maybe a project needs to be split up
- You [accidentally checked in large chunks of \(unused\) information](#) (perhaps some .iso was somehow included), which is sitting there bloating your repository for no good reason

All these cases have already happened. Most notably, the first case [happened with the Apache codebase](#), where third-party code that Apache did not have intellectual rights of was checked into the Subversion repository. They simply couldn't get it out, because there was no command like Obliterate. Deleting them the normal way wasn't enough, since you could still get to the pre-deletion version. There is no way, at all, to accomplish this... save for one workaround that is not bulletproof (we'll come to that).

These are not use cases I came up with. They are suggested many times already and the subversion dev team knows about these. In fact, and here's something that will blow your mind, they have known about these problems for 7 years. Say what? Take a look at the [Obliterate feature request](#) in the Subversion issue tracker. All these use cases have been discussed and summarized there, and are acknowledged as valid ground for feature implementation. Notice the first post of October 2001. It was then when it was first 'officially' reported as an issue.

Also take a look at the feature requests on the [mailing lists](#).

The broken workaround

There exists a feature in Subversion called 'dump'. It allows you to dump the entire repository into a single file. Then there is the feature 'load', that can fill an existing (empty) repository from such a dump file. The workaround revolves around a third feature called 'dumpfilter'. With this, you can filter a dump file and so remove files with a pathbased matching pattern.

This works up to a certain point:

- It is cumbersome, and works inconsistent in various environments (related to filesystems and trailing slashes)
- It is very slow (can end up in tens of hours for very large codebases)
- It does not always work (the infamous ['Invalid copy source path' error](#))

So while it is very nice to have at least some workaround to fall back to, it sometimes simply isn't a viable option. At least with CVS you could dive into the repository itself and remove the files manually.

In Subversion not so (I'm not implying though CVS is a better alternative).

So, after 712 years, where the hell is it?

That's the billion dollar question. Or to be more accurate, that's the X dollar question, where X is the bounty that some suggested should be offered for whoever gets the feature in ([cit.](#)). Apparently the problem lies in the nature of Subversion's filesystem; working in an Obliterate command supposedly requires tons of requirement specifications, design reports and even more debates on these before even a capable programmer could be put on the matter, with a salary from a sponsor. All this may be true, but... let me reiterate:

Open source. ~~Seven~~ Twelve years.

Apparently Subversion's codebase is so complex that it simply isn't accessible enough for the public to come up with a patch all this time. Remember Subversion has been one of the accepted de facto standards for Revision Control for many years, it should be one of the more prolific Open Source projects running to date. Yet after ~~seven~~ twelve years there has been no activity for an Obliterate command, save a whole lot of brain activity. In fact it apparently is so inaccessible, the Subversion development team itself doesn't feel confident it can pull it off without rewriting the filesystem ([cit.](#)). I'm not trying to trivialize the situation, it's no small task and documentation certainly is important. The truth is that the issue has been dead in the water for many years, while question marks keep popping up around the web.

How is it possible that such a widely acknowledged feature is still missing after all these years. Perhaps planned already? Let's see, what is Subversion's (official) standpoint on this:

- It is not on the [roadmap](#), medium or long term
- the [Subversion guide \(note 35\)](#) ([SVN Guide](#) by red-bean) mentions this feature is planned, though posts on the web suggest it has been in there since the beginning of time
- The [issue tracker](#) promises no timeframe and in fact states the development team is reluctant to do anything until design documents have been produced

Finally there is the ubiquitous [FAQ entry](#) about version history removal that states:

How do I completely remove a file from the repository's history?

There are special cases where you might want to destroy all evidence of a file or commit. (Perhaps somebody accidentally committed a confidential document.) This isn't so easy, *because Subversion is deliberately designed to never lose information*. Revisions are immutable trees which build upon one another. Removing a revision from history would cause a domino effect, creating chaos in all subsequent revisions and possibly invalidating all working copies. The project has plans, however, to someday implement an `svnadmin obliterate` command which would accomplish the task of permanently deleting information. (See issue tracker, issue 516.)

In the meantime, your only recourse is to `svnadmin dump` your repository, then pipe the dumpfile through `svndumpfilter` (excluding the bad path) into an `svnadmin load` command. See chapter 5 of

the Subversion book for details about this.

It all seems a case of a design flaw that is almost irreparable due to a complex filesystem implementation and ~~seven~~twelve years of labor on this implementation.

Recently (around 2010) There has been [an effort by the svn team](#) to spark such a command into existence, but it died a silent death as it proved too difficult to implement.

It appears Obliterate is still on the [roadmap](#) as one of the “Most Wanted” features.

What to do now

There seems little we can do. Try downloading the [sourcecode](#) and see if you can make sense of it all. Be aware, even the core developers are afraid to touch it without refined documentation backup them up. Maybe go along with the bounty idea and attract [Cowboy Coders](#) that lack the compulsive need for documentation.

Okay, that was a little bit harsh. But what's the common man to do, but wait for more than a decade? Meanwhile, take a look at the comparison list of [revision Control software](#).

Finally, here's a [semi-interview](#) from 27 February 2008 I found, where Karl Fogel states what needs to be done to get the ball rolling.