

---

# Using Subversion to Lock a DataFax Database or Version Control Study Setup

Phil Kirsch  
*Statistical Center for HIV/AIDS Research and Prevention*  
Fred Hutchinson Cancer Research Center  
Seattle, WA, USA



---

## The Problem

How do I preserve previous  
versions of a file without making  
multiple copies?

---

## The Solution

Version Control

### Examples of Version Control software include

---

- Microsoft Source Safe
- IBM Clear Case
- CVS
- GIT

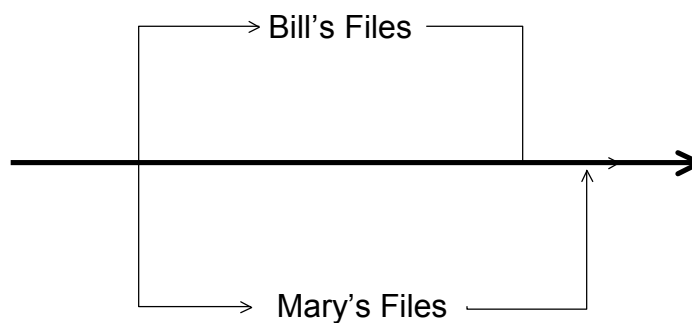
## Theory

---

- Assign a unique number to every version
- Store only the differences between the current and any previous version.
- Allow each individual to keep a unique set of versions during development phases.
- Merge these various branches to create a new test or production instance.

## Branches

---



## Should not be confused with

---

- Document Management
- Configuration Management

## Uses

---

- Most commonly to support software development
- Works on any file
- Simply a way to go back to a previous state
- Best for files that change over time

## Not recommended for

---

- Pictures
- Music
- Or other situations where you tend to add files rather than change them.

## Why Subversion?

---

- Open Source
- Multi-platform
- Web-based or stand-alone
- GUIs available
  - SmartSVN
  - Tortoise SVN
- Paid Support Available

## Why Subversion in DataFax

---

- Access to previous versions of
  - Study Setup
  - Visit and page maps
  - Edit checks
- For
  - Configuration management
  - Data archive (It may not be the most efficient form of archive, but it can easily transfer between various storage media.)
- Saving Test Data
- 'Freezing' data

## How would you get back to an old setup, visit map or edit check?

---

- Backup?
  - Index files are large enough that it's hard to keep incremental backups more than a few months.
- Refuse to change?
  - When is the last time you completed a study without a single protocol amendment?
- Hope no one will ask?

## An Example at SCHARP

---

- Perl code uses
  - svn status to identify new/changed files
  - svn add when new files are encountered
  - svn commit to send any differences to the repository
- And produces email to inform administrators about
  - What changed
  - Any problems encountered

## The end result looks like this

---

```
svn log DFsetup | more
r26943 | datafax | 2008-11-20 04:03:16 -0800 (Thu, 20 Nov 2008) | 1 line
Automatic Update
r26882 | datafax | 2008-11-18 04:03:34 -0800 (Tue, 18 Nov 2008) | 1 line
Automatic Update
r26293 | datafax | 2008-10-29 04:03:41 -0700 (Wed, 29 Oct 2008) | 1 line
Automatic Update
r25851 | datafax | 2008-10-16 04:03:11 -0700 (Thu, 16 Oct 2008) | 1 line
Automatic Update
r25805 | datafax | 2008-10-15 04:03:16 -0700 (Wed, 15 Oct 2008) | 1 line
Automatic Update
r24797 | datafax | 2008-09-12 04:03:04 -0700 (Fri, 12 Sep 2008) | 1 line
Automatic Update
```

## Pitfalls

---

- Single 'owner' approach
- Takes over some functions of the Operating System
- Increased complexity
  - Adds must be followed by commits
  - People who copy directories sometimes get more than they wanted.

## Special Issues when cronning Subversion for DataFax

---

- "Tilde" files, e.g. DFsetup~ go away without notifying subversion
- So do other temporary files, i.e. DFsetup.autosave
- Users who delete things using the Operating System
- The first commit after a password change must be done manually.



## What about Validation?

---

- Like other open source projects, Subversion has a defined user community
- With verifiable development practices
- Publically available
  - Requirements
  - Bug reports

---

A good version control system is essential to DataFax configuration.

Subversion is an effective, well-documented, low-cost alternative.