

# Version Control With Subversion

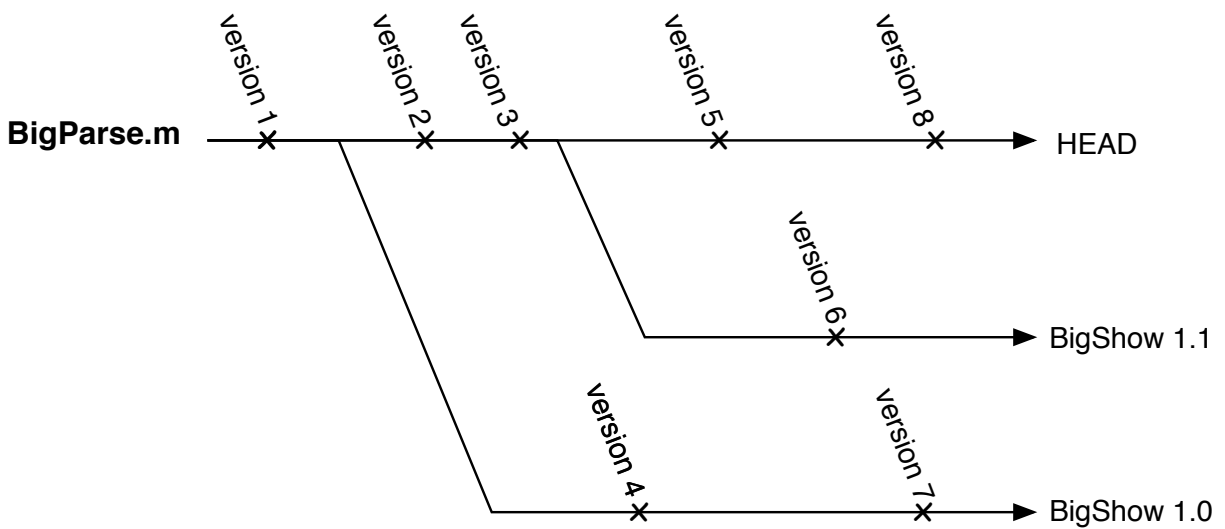
## **Subversion**

- What is version control?
- What is subversion?
- How do you use it?
- Getting more info

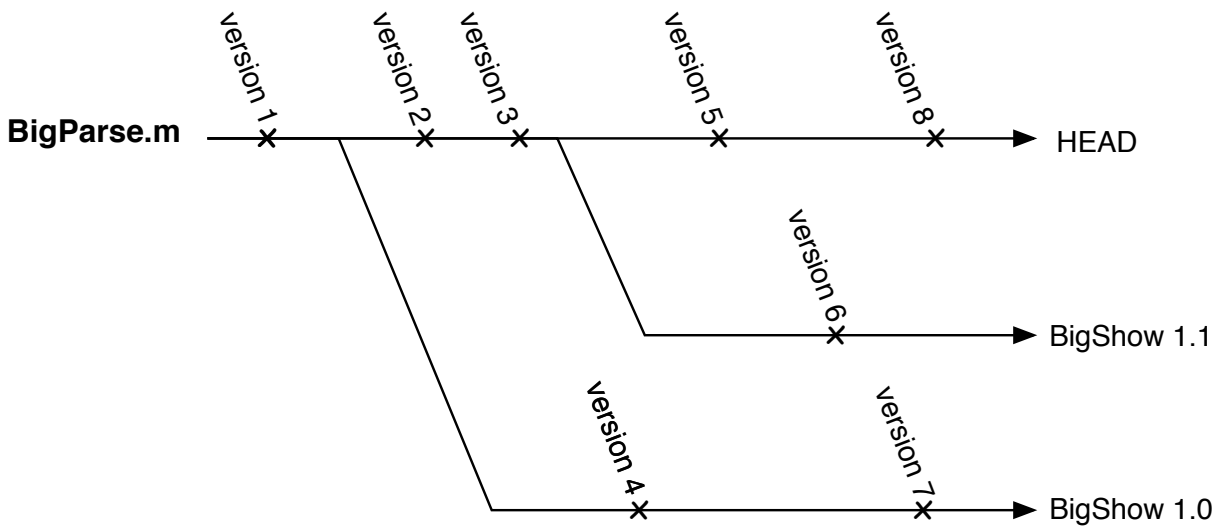
## What is version control?

- Revision Control
- Version Control
- Version Management
- Source Code Control
- Source Code Management

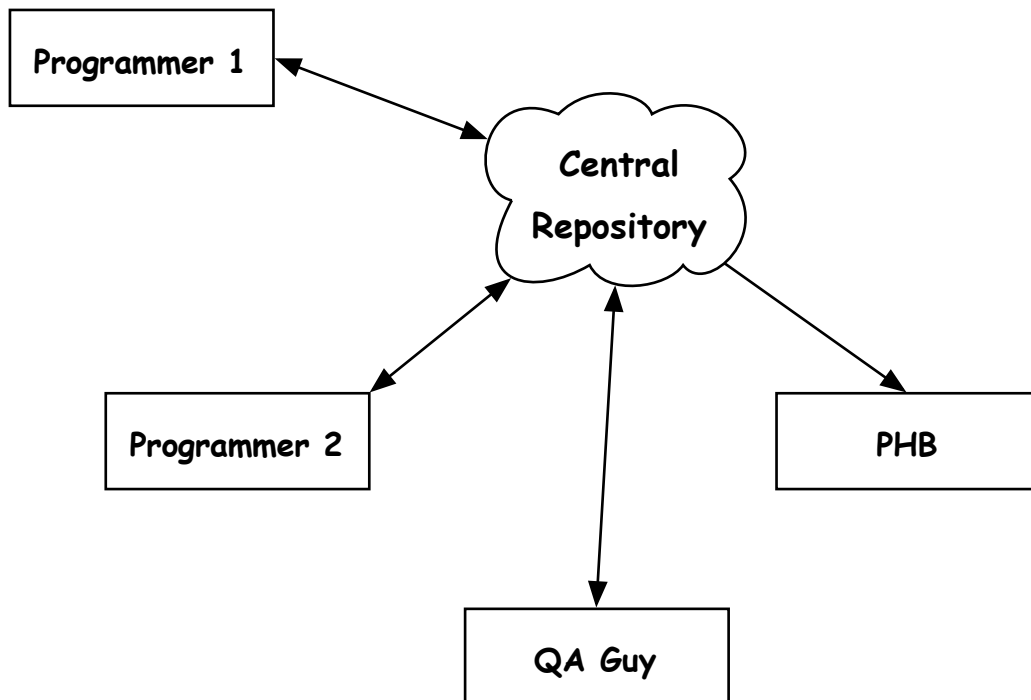
## It Is History



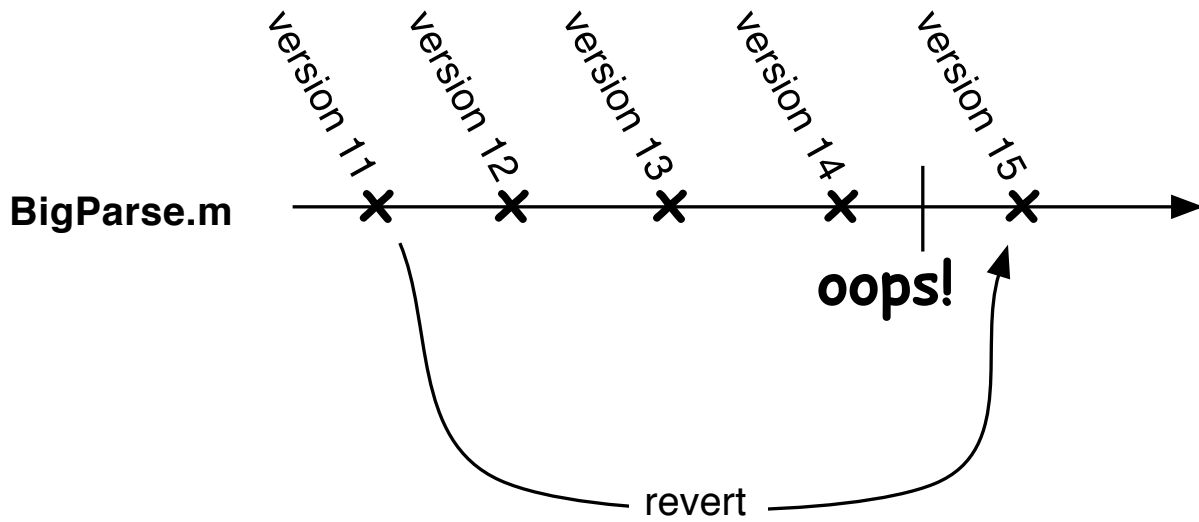
## It Is Parallel Development



## It Is Collaboration



# It Is Undo



## Version Control Systems

### Centralized Repository:

RCS → CVS → Subversion (Free)  
Perforce (Commercial)

### Distributed Repository:

GNU Arch (Free)  
BitKeeper (Commercial)

## **Lock, Edit, Unlock**

- Locking prevents other from editing the file
- No conflicts
- Serialized Modifications to files
- Doesn't Scale
- The Aruba Problem

## **Edit, Merge**

- Edit any file at any time
- Concurrent modifications to files
- Conflicts can happen
- Easier for remote development

## **Finally! On to Subversion**

- Created to address limitations in CVS
- Up to version 1.2.0
- Used by a number of high-profile projects
- Concepts and commands similar to CVS

## **CVS Limitations Addressed**

- Can't version directories
- Can't version symbolic links
- Can't easily move / rename files
- Commits aren't atomic
- Remote Access Limitations
- Inefficient handling of binary files
- Branching is byzantine
- Tagging can be time consuming
- Costs proportional to data size

## Creating a Respository

```
$ svnadmin create --fs-type fsfs /usr/local/svnroot/
```

or

```
$ svnadmin create --fs-type bdb /usr/local/svnroot/
```

## Subversion Commands

```
svn <command> <command-options> <file-name(s)>
```

### Commands:

- add
- checkout
- commit
- diff
- import
- log
- propget
- propset
- status
- update

### Options:

- dry-run
- verbose / -v
- quiet / -q
- message / -m
- revision / -r

## Importing Projects

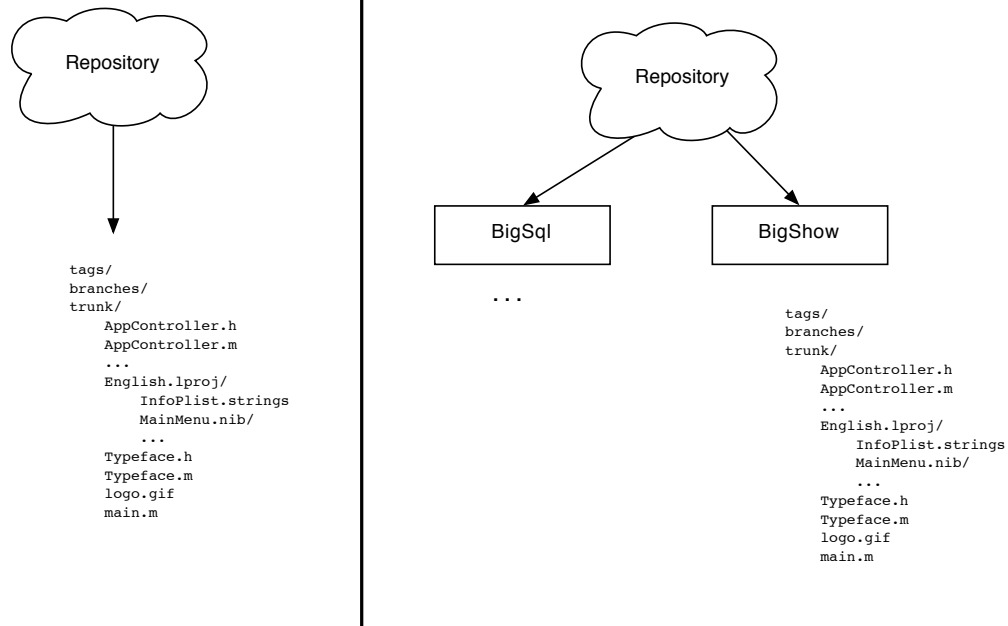
```
BigShow/  
  ApplicationController.h  
  ApplicationController.m  
  ...  
  English.lproj/  
    InfoPlist.strings  
    MainMenu.nib/  
    ...  
  Typeface.h  
  Typeface.m  
  logo.gif  
  main.m
```

## Subversion Project Layout

```
BigShow/  
  tags/  
  branches/  
  trunk/  
    ApplicationController.h  
    ApplicationController.m  
    ...  
    English.lproj/  
      InfoPlist.strings  
      MainMenu.nib/  
      ...  
    Typeface.h  
    Typeface.m  
    logo.gif  
    main.m
```



# Repository Decision Time



## Multi-project Repository

```
$ svn mkdir -m "initial revision" \  
  file:///usr/local/svnroot/bigshow
```

```
$ svn mkdir -m "initial revision" \  
  file:///usr/local/svnroot/bigsql
```

## Import

```
$ cd/Path/To/BigShow
$ svn import -m "initial import" \
    . file:///usr/local/svnroot/bigshow
```

```
Adding trunk
Adding trunk/MyDocument.m
Adding trunk/main.m
Adding trunk/BigWindow.h
...
Adding trunk/PrefsController.m
Adding branches
Adding tags
Committed revision 2.
```

## Check It Out

```
$ cd ~/Development/Projects
$ svn checkout \
    file:///usr/local/svnroot/bigshow/trunkBigShow
```

```
A BigShow/MyDocument.m
A BigShow/BigWindow.h
A BigShow/main.m
...
A BigShow/Image.m
A BigShow/PrefsController.m
Checked out revision 2.
```

# Making Commitments

```
$ svn commit -m \  
  "Fixed dereference of NULL in para tag handler" \  
  BigParse.m
```

```
Sending BigParse.m  
Transmitting file data .  
Committed revision 3.
```

# Differences

```
$ svn diff BigParse.m  
  
=====
```

```
--- BigParse.m (revision 9)  
+++ BigParse.m (working copy)  
@@ -49,17 +49,17 @@  
     case kCFXMLNodeTypeEntityReference:  
         theText = CFXMLNodeGetString(node);  
         if ([NSString *)theText isEqual:@"lt"]  
+         return @"&lt;";  
+         if ([NSString *)theText isEqual:@"gt"]  
         return @"&gt;";  
-         if ([NSString *)theText isEqual:@"gt"]  
-         return @"&lt;";  
         if ([NSString *)theText isEqual:@"t"]  
         return @"\t";  
-         if ([NSString *)theText isEqual:@"bork"]  
-         return @":-";  
+         if ([NSString *)theText isEqual:@"spc"]  
+         return @" ";  
  
         NSLog(@"Unknown entity:%@", theText);  
         return NULL;
```

## Adding New Files

```
$ svnadd BigParse2.[hm]
```

```
A          BigParse2.h
```

```
A          BigParse2.m
```

```
$ svn commit -m "Split BigParse into two parts."
```

```
Adding          BigParse2.h
```

```
Adding          BigParse2.m
```

```
Transmitting file data ..
```

```
Committed revision 4.
```

## Removing Files

```
$ svn rm BigParse2.[hm]
```

```
D          BigParse2.h
```

```
D          BigParse2.m
```

```
$ svn commit -m "Didn't really need these files anyway."
```

```
Deleting          BigParse2.h
```

```
Deleting          BigParse2.m
```

```
Committed revision 5.
```

## Looking at Logs

```
$ svn log BigParse.m
-----
r13 | markd | 2005-03-27 23:44:42 -0500 (Sun, 27 Mar 2005) | 3 lines
Revised node type processing
-----
r12 | bork | 2005-03-27 23:09:35 -0500 (Sun, 27 Mar 2005) | 6 lines
Added new logic to node type processing
-----
r4 | markd | 2005-03-26 16:48:15 -0500 (Sun, 27 Mar 2005) | 87 lines
Split BigParse into two parts.
-----
r3 | markd | 2005-03-26 16:48:15 -0500 (Sat, 26 Mar 2005) | 1 line
Fixed dereference of NULL in para tag handler
-----
r2 | markd | 2005-03-26 15:38:31 -0500 (Sat, 26 Mar 2005) | 1 line
initial import
-----
```

## Seeing Status

```
$ svn status
M      BigParse.h
M      BigParse.m
A      BigParse2.h
A      BigParse2.m
D      test-case23.txt

$ svn status --show-updates
*      11      BigParse.m
*      11      Slide.m
```

## Updating Stuff

```
$ svn update
```

```
U  BigParse.m
```

```
U  Slide.h
```

```
Updated to revision 11.
```

## Conflicts

```
$ svn update
```

```
C  BigParse.m
```

```
Updated to revision 12.
```

```
        break;
    case kCFXMLNodeTypeProcessingInstruction:
<<<<<<< .mine
        theClass = [Show classForElementName:
                    (NSString *)CFXMLNodeGetString(node)];
=====
        theText = CFXMLNodeGetString(node);
        return (void*)theText;
>>>>>>> .r12
        break;
    case kCFXMLNodeTypeComment:
        return NULL;
```

## Conflict File Generation

```
$ ls BigParse.m*
```

```
BigParse.m.mine  
BigParse.m.r11  
BigParse.m.r12
```

```
$ svn resolved BigParse.m
```

```
Resolved conflicted state of 'BigParse.m'
```

```
$ svn commit -m "Revised node type processing" BigParse.m
```

```
Sending          BigParse.m  
Transmitting file data .  
Committed revision 13.
```

## Placing Blame

```
$ svn annotate BigParse.m > BigParse.m-annotated
```

```
...  
2   markd          break;  
2   markd          case kCFXMLNodeTypeProcessingInstruction:  
13  markd          theClass = [Show classForElementName:  
13  markd          (NSString *)CFXMLNodeGetString(node)];  
2   markd          break;  
6   bork           case kCFXMLNodeTypeComment:  
6   bork           return NULL;  
6   bork           break;  
2   markd          case kCFXMLNodeTypeText:
```

## Specifying Revisions

```
$ svn log -r 12 BigParse.m
```

```
-----  
r12 | bork | 2005-03-27 23:09:35 -0500 (Sun, 27 Mar 2005) | 6 lines
```

```
Added new logic to node type processing  
-----
```

```
$ svn log -r 5 BigParse.m
```

## Specifying Revision Ranges

```
$ svn log -r2:4 BigParse.m
```

```
-----  
r4 | markd | 2005-03-26 16:48:15 -0500 (Sun, 27 Mar 2005) | 87 lines
```

```
Split BigParse into two parts.  
-----
```

```
r3 | markd | 2005-03-26 16:48:15 -0500 (Sat, 26 Mar 2005) | 1 line
```

```
Fixed dereference of NULL in para tag handler  
-----
```

```
r2 | markd | 2005-03-26 15:38:31 -0500 (Sat, 26 Mar 2005) | 1 line
```

```
initial import  
-----
```



## Revisions By Date

```
$ svn log -r {2005-03-27} BigParse.m
```

```
-----  
r4 | markd | 2005-03-26 16:48:15 -0500 (Sun, 27 Mar 2005) | 87 lines
```

```
Split BigParse into two parts.  
-----
```

```
$ svn log -r { "2005-03-27 16:48:00" } BigParse.m
```

## Tagging

```
$ svn copy \  
    file:///usr/local/svnroot/bigshow/trunk \  
    file:///usr/local/svnroot/bigshow/tags/release-1.0 \  
    -m "Tag for BigShow 1.0"
```

```
Committed revision 20.
```

```
$ svn checkout \  
    file:///usr/local/svnroot/BigShow/tags/release-1.0 \  
    bigshow-1.0
```

```
A bigshow-1.0/MyDocument.m  
A bigshow-1.0/BigWindow.h  
A bigshow-1.0/main.m  
...  
A bigshow-1.0/Image.m  
A bigshow-1.0/PrefsController.m  
Checked out revision 27.
```

## Branching

```
$ svn copy \  
  file:///usr/local/svnroot/bigshow/tags/release-1.0 \  
  file:///usr/local/svnroot/bigshow/branches/branch-1.0 \  
  -m "Branch for BigShow 1.0"
```

```
Committed revision 22.
```

## Properties

```
$ svn propset "todo" \  
  "need to reflux the frobulator" BigParse.m
```

```
property 'todo' set on 'BigParse.m'
```

```
$ svn status
```

```
M      BigParse.m
```

```
$ svn commit -m "updated todo" BigParse.m
```

```
Sending      BigParse.m  
Committed revision 25.
```

```
$ svn propget todo BigParse.m
```

```
need to reflux the frobulator
```

## **Subversion Properties**

- svn:executable
- svn:mime-type
- svn:keywords (\$Id: \$, \$Date: \$, etc)
- svn:eol-style
- svn:externals

## **System Administration issues**

- .svn directories
- Making backups

## Remote Repositories

- Apache WebDAV (`http://` / `https://`)
- svnserv (`svn://`)
- svn over ssh (`svn+ssh://`)

## Mac OS X Weirdness

- File bundles and `.svn` directory
- Xcode
- Conflicts with Projects and Nib Files

## Summary

- Version Control is good
- Subversion is good

## Resources

- <http://subversion.org> / <http://subversion.tigris.org>
- <http://svnbook.org> / <http://svnbook.red-bean.com>
- #svn on Feenode IRC