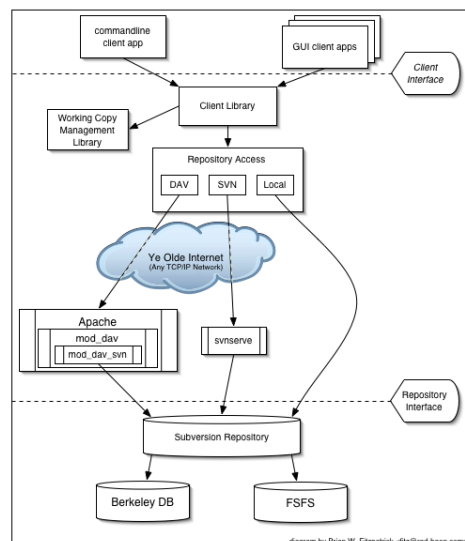


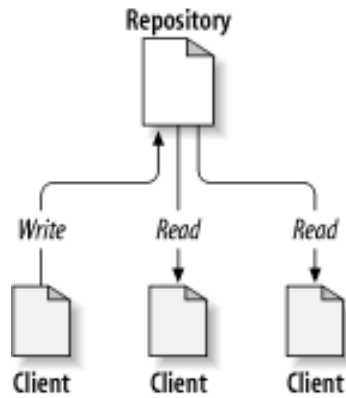
Subversion

- Fundamental Concepts
- Basic Usage

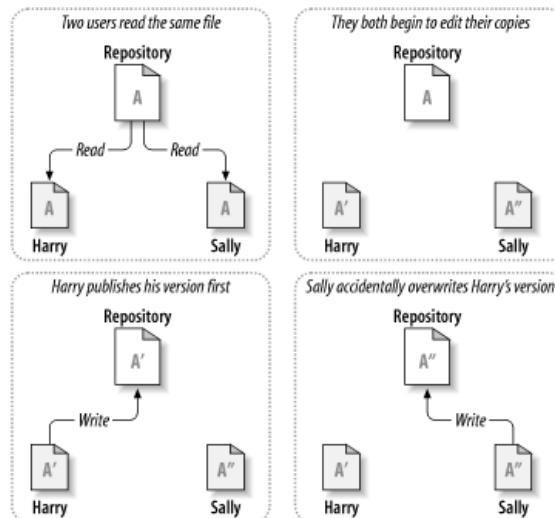
Architecture



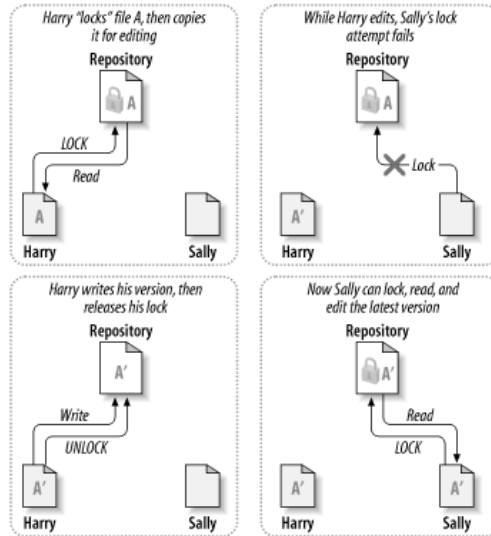
The Repository



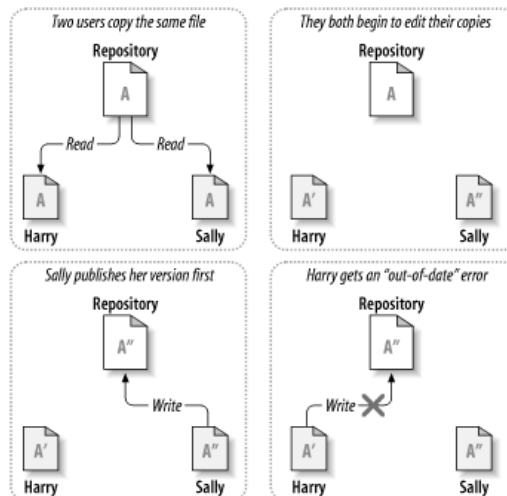
File Sharing



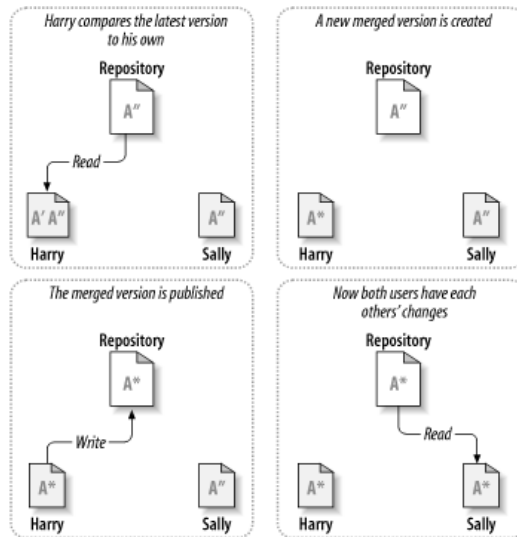
Lock-Modify-Unlock



Copy-Modify-Merge (1)



Copy-Modify-Merge (2)



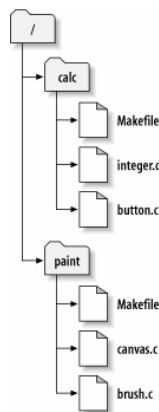
Repository URL

Schema	Access method
file:///	Direct repository access (on local disk)
http://	Access to Subversion-aware Apache server
https://	Same as http://, but with SSL encryption
svn://	Access via custom protocol to an svnserve server
svn+ssh://	Same as svn://, but through an SSH tunnel

Working Copy

- ❑ A private copy of a project in the repository, plus some administrative files
 - Administrative files are contained in a subdirectory `.svn`, and help Subversion to recognize which files have been changed, and which files have been out-of-date.
- ❑ Provides a private working space which is not interfered by other people, and changes are not visible to others
- ❑ After changes are done and verified in the working copy, they can be published by writing to the repository
- ❑ Commands are also provided to incorporate other people's changes into the working copy

Example (1)



Example (2)

```
$ svn checkout http://svn.example.com/repos/calc
A calc/Makefile
A calc/integer.c
A calc/button.c
Checked out revision 56.
$ ls -A calc
Makefile button.c integer.c .svn/
```

Example (3)

```
$ svn commit button.c -m "Fixed a typo in button.c."
Sending button.c
Transmitting file data .
Committed revision 57.
```

Example (4)

```
$ pwd
/home/sally/calc
$ ls -A
Makefile button.c integer.c .svn/
$ svn update
U button.c
Updated to revision 57.
```

Subversion

- Fundamental Concepts
- Basic Usage

Help

```
$ svn help import
import: Commit an unversioned file or tree into the repository.
usage: import [PATH] URL
Recursively commit a copy of PATH to URL.
If PATH is omitted '.' is assumed.
Parent directories are created as necessary in the repository.
If PATH is a directory, the contents of the directory are added
directly under URL.
Unversionable items such as device files and pipes are ignored
if --force is specified.
Valid options:
-q [--quiet] : print nothing, or only summary information
-N [--non-recursive] : obsolete; try --depth=files or --depth=immediates
--depth ARG : limit operation by depth ARG ('empty', 'files',
' immediates', or 'infinity')
...
```

Basic Work Cycle

1. Update your working copy: **svn update**
2. Make changes: **svn add**; **svn delete**; **svn copy**; **svn move**
3. Examine your changes: **svn status**; **svn diff**
4. Possibly undo some changes: **svn revert**
5. Resolve conflicts: **svn update**; **svn resolve**
6. Commit your changes: **svn commit**
7. Browse the repository: **svn cat**, **svn list**

Update Working Copy

```
$ svn update
U foo.c
U bar.c
Updated to revision 2.
```

File Change vs Tree Change

- ❑ **File change:** change an existing file in the working copy, do not need to notify Subversion
- ❑ **Tree change:** change a directory structure, e.g., add, delete, copy, or move a file, need to notify Subversion
 - **svn add, svn delete, svn copy, svn move, svn mkdir**

Tree Change Commands

- ❑ `svn add foo`: add `foo` into the repository
- ❑ `svn delete foo`: delete `foo` from the repository
 - If `foo` is a file or link, `foo` is immediately deleted from the working copy; if `foo` is a directory, `foo` will be removed on the next commit
- ❑ `svn copy foo bar`: create a duplicate of `foo` in the repository
- ❑ `svn move foo bar`: = `svn copy foo bar`; `svn delete foo`
- ❑ `svn mkdir blort`: = `mkdir blort`; `svn add blort`

Examine Your Changes (1)

```
$ svn status
? scratch.c          # file is not under version control
A stuff/loot/boo.h  # file is scheduled for addition
C stuff/loot/lump.c # file has textual conflicts from an update
D stuff/fish.c      # file is scheduled for deletion
M bar.c             # the content in bar.c has local modifications
```

Examine Your changes (2)

```
$ svn diff
Index: bar.c
-----
--- bar.c (revision 3)
+++ bar.c (working copy)
@@ -1,7 +1,12 @@
+#include <sys/types.h>
+#include <sys/stat.h>
+#include <unistd.h>
+
+#include <stdio.h>
int main(void) {
- printf("Sixty-four slices of American Cheese...\n");
+ printf("Sixty-five slices of American Cheese...\n");
return 0;
}
Index: README
-----
--- README (revision 3)
+++ README (working copy)
@@ -193,3 +193,4 @@
+Note to self: pick up laundry.
Index: stuff/fish.c
-----
--- stuff/fish.c (revision 1)
+++ stuff/fish.c (working copy)
-Welcome to the file known as 'fish'.
-Information on fish will be here soon.
...
```

Undo Working Changes

```
$ svn revert README
Reverted 'README'

$ svn status foo
? foo
$ svn add foo
A foo
$ svn revert foo
Reverted 'foo'
$ svn status foo
? foo
```

Resolving Conflicts (1)

```
$ svn update
U INSTALL
G README
Conflict discovered in 'bar.c'.
Select: (p) postpone, (df) diff-full, (e) edit,
(h) help for more options:
```

Resolving Conflicts (2)

```
$ cat sandwich.txt
Top piece of bread
Mayonnaise
Lettuce
Tomato
Provolone
<<<<<< .mine
Salami
Mortadella
Prosciutto
=====
Sauerkraut
Grilled Chicken
>>>>>> .r2
Creole Mustard
Bottom piece of bread
```

Resolving Conflicts (3)

```
$ svn resolve --accept working sandwich.txt  
Resolved conflicted state of 'sandwich.txt'  
$ svn commit -m "Go ahead and use my sandwich,  
discarding Sally's edits."
```

Commit Your Changes (1)

```
$ svn commit -m "Corrected number of cheese  
slices."  
Sending sandwich.txt  
Transmitting file data .  
Committed revision 3.
```

Commit Your Changes (2)

```
$ svn commit -m "Add another rule"  
Sending rules.txt  
svn: Commit failed (details follow):  
svn: File '/sandwich.txt' is out of date  
...
```

Examining History

- **svn log**: Shows you broad information: log messages with date and author information attached to revisions and which paths changed in each revision
- **svn diff**: Shows line-level details of a particular change
- **svn cat**: Retrieves a file as it existed in a particular revision number and displays it on your screen
- **svn list**: Displays the files in a directory for any given revision

svn log

```
$ svn log
-----
r3 | sally | 2008-05-15 23:09:28 -0500 (Thu, 15 May 2008) | 1 line
Added include lines and corrected # of cheese slices.
-----
r2 | harry | 2008-05-14 18:43:15 -0500 (Wed, 14 May 2008) | 1 line
Added main() methods.
-----
r1 | sally | 2008-05-10 19:50:31 -0500 (Sat, 10 May 2008) | 1 line

Initial import
-----
```

svn diff

```
$ svn diff -r 3 rules.txt
Index: rules.txt
=====
==
--- rules.txt (revision 3)
+++ rules.txt (working copy)
@@ -1,4 +1,5 @@
Be kind to others
Freedom = Responsibility
Everything in moderation
-Chew with your mouth open
+Chew with your mouth closed
+Listen when others are speaking
$
```

Browsing Repository

- ❑ `svn cat`: examine an earlier version of a file
- ❑ `svn list`: shows what files in the repository without downloading them

svn cat

```
$ svn cat -r 2 rules.txt  
Be kind to others  
Freedom = Chocolate Ice Cream  
Everything in moderation  
Chew with your mouth open  
$
```


svn list

```
$ svn list http://svn.collab.net/repos/svn  
README  
branches/  
clients/  
tags/  
trunk/
```