Introduction to Subversion

Wendy Smoak

Rob Richardson

Wendy Smoak

- Sr. Systems Analyst,
  Arizona State University
  - Web application development
  - Systems and database administration

- Member, The Apache Software Foundation
  - Struts, MyFaces, Shale, Maven
  - Infrastructure, release management, documentation

- Contact me:
 - wsmoak@apache.org
  - http://www.wsmoak.net

Rob Richardson

- Principal, Richardson & Sons
  - Provides enterprise software development for small- to medium-sized businesses.
  - Has created software applications raging from enterprise-scale applications to PDA-based systems, web applications to embedded database synchronization and postscript processing systems.
  - Specializes in translating business requirements into technical solutions.
  - Our business is listening to your needs and building you effective tools.

- My Qualifications
  - Masters of Science in Computer Information Systems (MSCIS)
  - Bachelors of Fine Arts in Industrial Design (BFA ID), the study of human factors and human / technology interaction
  - Over 10 years of software development experience

- Contact me:
  - richardsonandsons.com/contact.aspx

What is source control?

- A central place to store your code
  - Backup and recovery

- Records a history of code changes

- Facilitates collaboration among development team members

- Easy to check out prior code, undo changes, version products
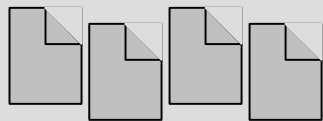
How does source control work?

User1

Create Project
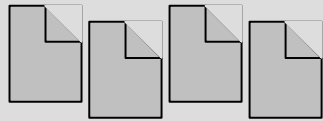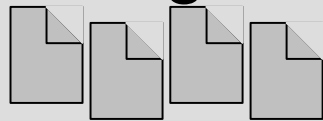
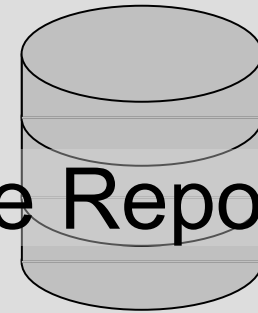Import
or Add

The Repository

Update

User2

Working Copy

Commit

Check Out

User3

Working Copy

Why should I use source control?

- Because the alternative is frightening beyond all reason

- Some motivation:

  "Well, I'd been working on the code, and got it to a state where I was prepared to share it. Then I thought I'd clean up the target directory and do a new build just to be sure. So I type rm -r *. But hold on, I'm not in the target directory, I'm in the project home directory, and I'd just deleted the entire project. Luckily I've got earlier versions lying around but I've lost a few hours work."

What is Subversion?

- Free/open source version control system

- Replacement for CVS
  - CVS was the de facto standard on Unix/Linux/BSD

- Started and still partially funded by CollabNet

- Tracks changes in any type of file, not just source code

Tradeoffs

- Disk space
  - Local control files require disk space

- Control
  - Must move and rename files through the tools

- User Maintenance Time
  - Users must be diligent in checking in and out files
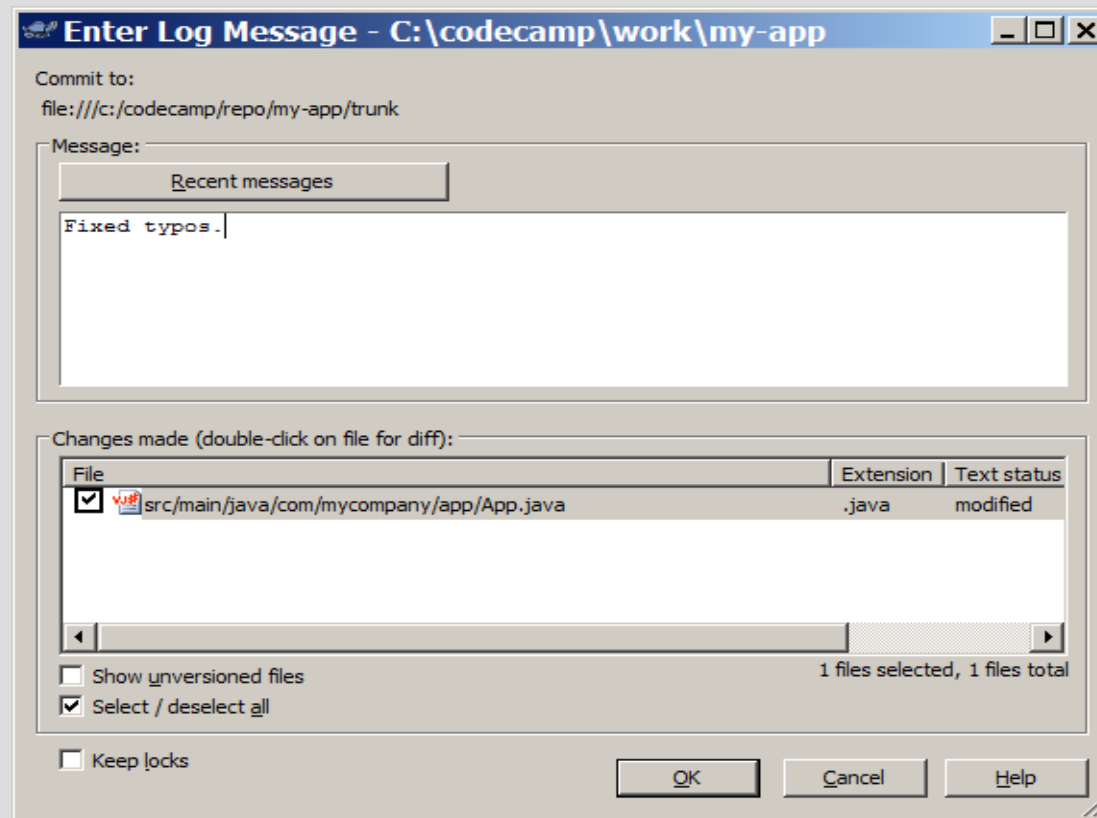
Choosing a Subversion Client

- Command line
  - Text commands, scriptable

- Graphical
  - TortoiseSVN

- IDE Integration

- Browser-based viewers

- Same functionality, a matter of preference

Demo
Check Out, Update, Commit

Commit Messages

- Explain what the change does and why you are making it

When to Commit

- When you've finished a feature

- Don't break the repository's build

- Don't diverge too far from the repository

- Be a good team player

- Work with your organization to develop practices that match your business culture

What to Commit:
If you can build it, don't commit it

- Commit these:
  - Source files
  - Saved database info:
    - stored procedures
    - table definitions
  - Resources
    - Images
    - resx files
  - Configuration files
  - Build notes
  - Test harnesses

- Not these:
  - User files
    - *.suo
    - *.csproj.user
  - Temp files
    - obj & bin folders
    - Debug & Release folders
  - Built files
    - *.dll, *.exe
    - *.cab, *.tar
    - *.jar, *.msi
  - The database
  - Absolute path references

Choosing a Subversion Client

- Command line
  - Text commands
  - Shell scripts

- Graphical
  - TortoiseSVN

Each tool has the same functionality Choose the tool that works for you

- IDE Integration
  - IntelliJ IDEA
  - Eclipse
  - Visual Studio

- Browser- based Viewers
  - ViewVC
  - FishEye
  - Repo Browser

IDE Integration

- IntelliJ IDEA

- Eclipse
  - http://subclipse.tigris.org/

- Visual Studio
  - Ankh: http://ankhsvn.tigris.org/
  - SvnLitePlugin:
    - http://cnicholson.net/content.php?id=61
  - VisualSVN: http://www.visualsvn.com
  - PushOK: http://www.pushok.com/soft_svn.php
  - TortoiseSVN + VS.NET:
    - http://garrys-brain.blogspot.com/2006/10/tortoisesvn-vsnet-2005-good-stuff.html

Viewing the contents of a repo

- ## TortoiseSVN Repo Browser

- ## ViewVC
  - formerly ViewCVS
  - http://www.viewvc.org/

- ## FishEye
  - Cenqua
  - Not free/open source

Demo
Viewing a Repository

Repository Design

- Multiple repositories?

- Convention, for each project
  - trunk
  - branches
  - tags

Creating a new repository

- TortoiseSVN "Create repository here"

- svnadmin at the command prompt

- Access the repository with file:// urls

Importing and Adding Code

- 'svn mkdir'
  - Creates a directory

- 'svn import'
  - Imports the <u>contents</u> of a directory

- 'svn add'
  - Adds to your working copy

Demo
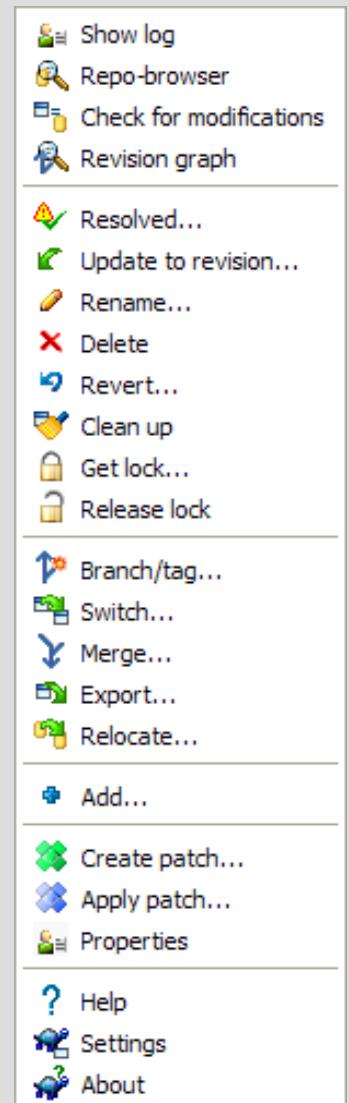## Create a repository and import a project

Moving and Renaming Files

- Use the tools, commit changes
  - 'svn move'
  - 'svn rename'

- Moving and renaming any other way will cause your working copy to get out of sync with the repository

- To recover, undo your changes, then re-do with svn commands.
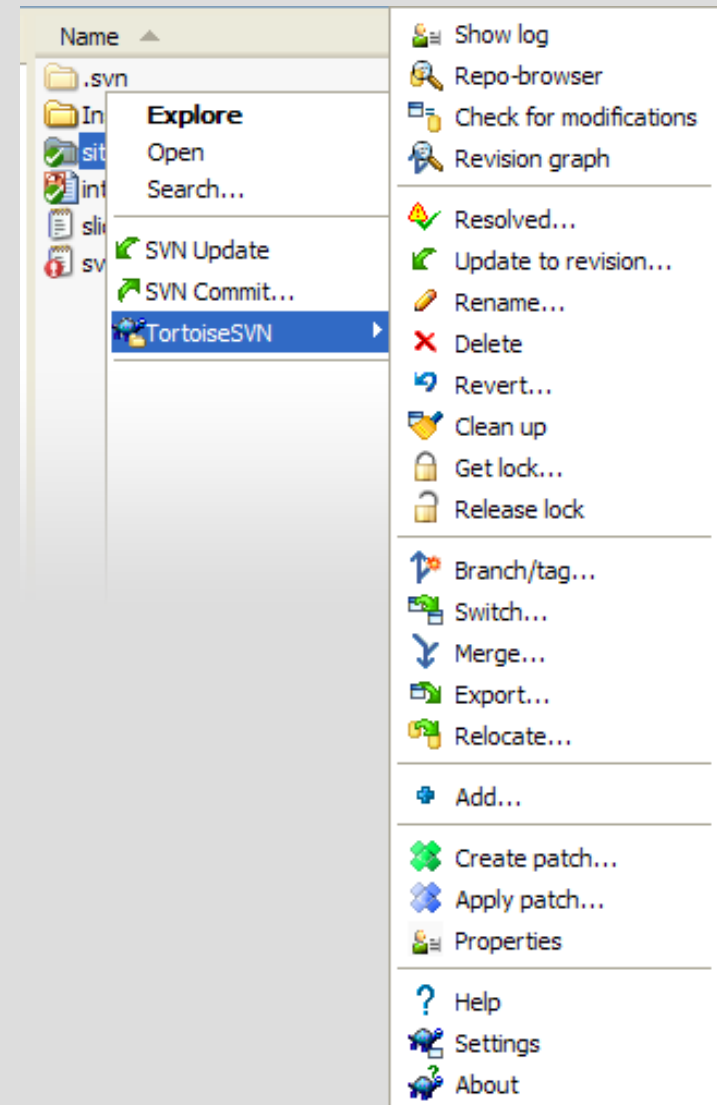
Removing and Reverting

- ## Nothing is ever truly gone
  - Be careful what you add!
  - Possible to remove with dump, filter, load
  - This is scary beyond measure

- ## 'svn rm' or 'svn delete'
  - Removes files from your working copy
  - Will remove files in project from now on
  - Will <u>not</u> remove the history of the file
    - (removing the file does <u>not</u> shrink repository)

- ## 'svn revert'
  - Local "undo" command
  - svn revert . (current directory)
  - svn revert -R * (recursive, everything)

Show log
Repo-browser
Check for modifications
Revision graph

Resolved...
Update to revision...
Rename...
Delete
Revert...
Clean up
Get lock...
Release lock

Branch/tag...
Switch...
Merge...
Export...
Relocate...

Add...

Create patch...
Apply patch...
Properties

Help
Settings
About

Subversion as a Server

- file://
  - OK for small applications
  - Not truly client / server
  - File system must provide security

- svn://
  - True client-server access mechanism
  - Run subversion as a windows service
  - Can integrate with ssh

- http:// or https://
  - Apache httpd webserver
  - Subversion comes with mod_*_svn.so
  - Various Apache authentication modules available
  - Apache can use with Windows authentication

Starting svnserve

- svnserve --help (for options)

- As a Windows Service
  - http://dark.clansoft.dk/~mbn/svnservice/SVNService.zip

- As a daemon
  - svnserve -d -r "c:\codecamp\repo"

- Listens on port 3690 by default
  - May need to configure firewall rules

- Now check out project via svn://
  - svn co svn://localhost/my-app/trunk my-app

Configuring svnserve

- Edit conf/svnserve.conf
  - anon-access = none
  - auth-access = write
  - password-db = passwd
  - authz-db = authz

- Add users and passwords to conf/passwd
  - userid = topsecret

- Add authorization rules to conf/authz
  - [/]
  - userid = rw
  - * =

Other topics

- Properties
  - svn:externals definitions
  - svn:ignore
  - svn:eol-style

- Converting repository to Subversion
  - cvs2svn
  - vss2svn

Resources

- Subversion
    - http://subversion.tigris.org

- TortoiseSVN
    - http://tortoisesvn.tigris.org

- CollabNet
    - http://www.collab.net

- Eric Sinc
    - http://software.ericsinc.com/scm/source-control.html

- SVN 1-click Setup
    - http://svn1clicksetup.tigris.org/

- Slides and Screenshots
    - http://www.wsmoak.net/subversion/demo

Books

- Version Control with Subversion
  - http://svnbook.red-bean.com

- Practical Subversion (Garrett Rooney)

- Pragmatic Version Control using Subversion (Mike Mason)

Questions?

fin