



# SUBVERSION

---

# SVNTest

- Download the SVNTest project from the class web site.
- Un-tar the project.
- Import the Project into Eclipse.
- Build, test, make sure the project works.

# Subversion

- What is source code version control?
  - <http://svnbook.red-bean.com/>
  - allow multiple people to modify the same source code
- allow one person to manage multiple versions of their source code
  - move from computer to computer to develop
  - track all changes

# Repository



zeus.cs.pacificu.edu  
/home/shereen/SVNROOT/

Store your source code on zeus  
check it out and edit it on any  
other machine and upload your  
changes back to zeus.

# Client



moe.cs.pacificu.edu  
/home/shereen/workspace/HelloWorld

# Client

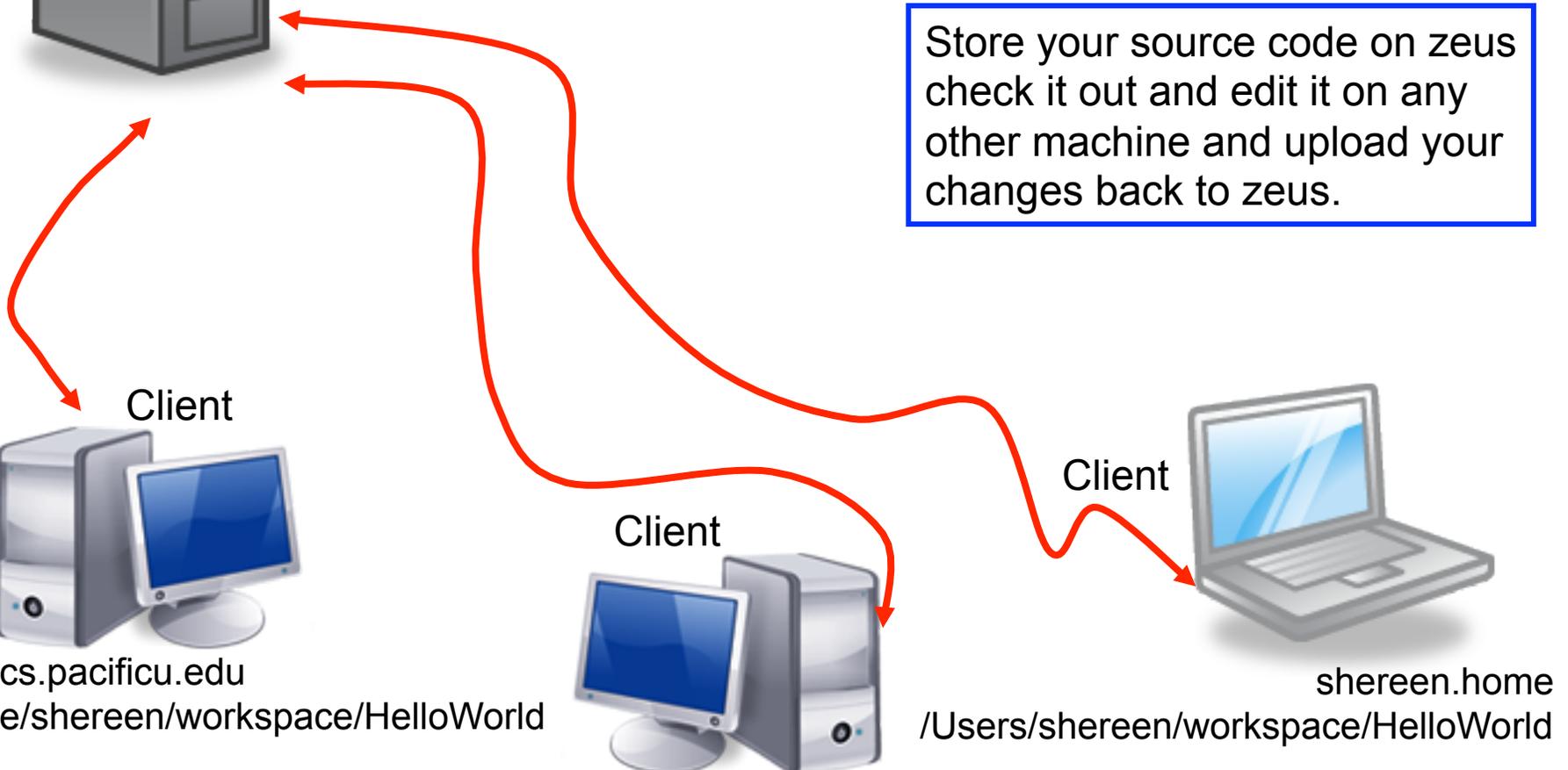


lisa.cs.pacificu.edu  
/home/shereen/workspace/HelloWorld

# Client



shereen.home  
/Users/shereen/workspace/HelloWorld



# Topics

- Subversion
  - Source Control
  - Check in
  - Check out
  - Update
  - Commit
  - Merge Conflict
  - Revert a file

# Version Control

- Each change you make to the source code is a **revision** stored in the repository
  - can annotate your change with a note
    - why did I do that?
  - you can browse back through the repository to find old revisions of file
    - changed a data structure and it did not work
    - rewrote an algorithm and it got slower!
  - check out the old (working) revision from the repository

# Hmmm....

- How often should I *update* and *commit*?
  - every major change
  - once every 15 minutes
  - right before you do something you think may be a bad idea
  - be sure to update and commit before you log off of a lab machine!
    - Or before you leave the lab
    - Someone may reboot your machine!

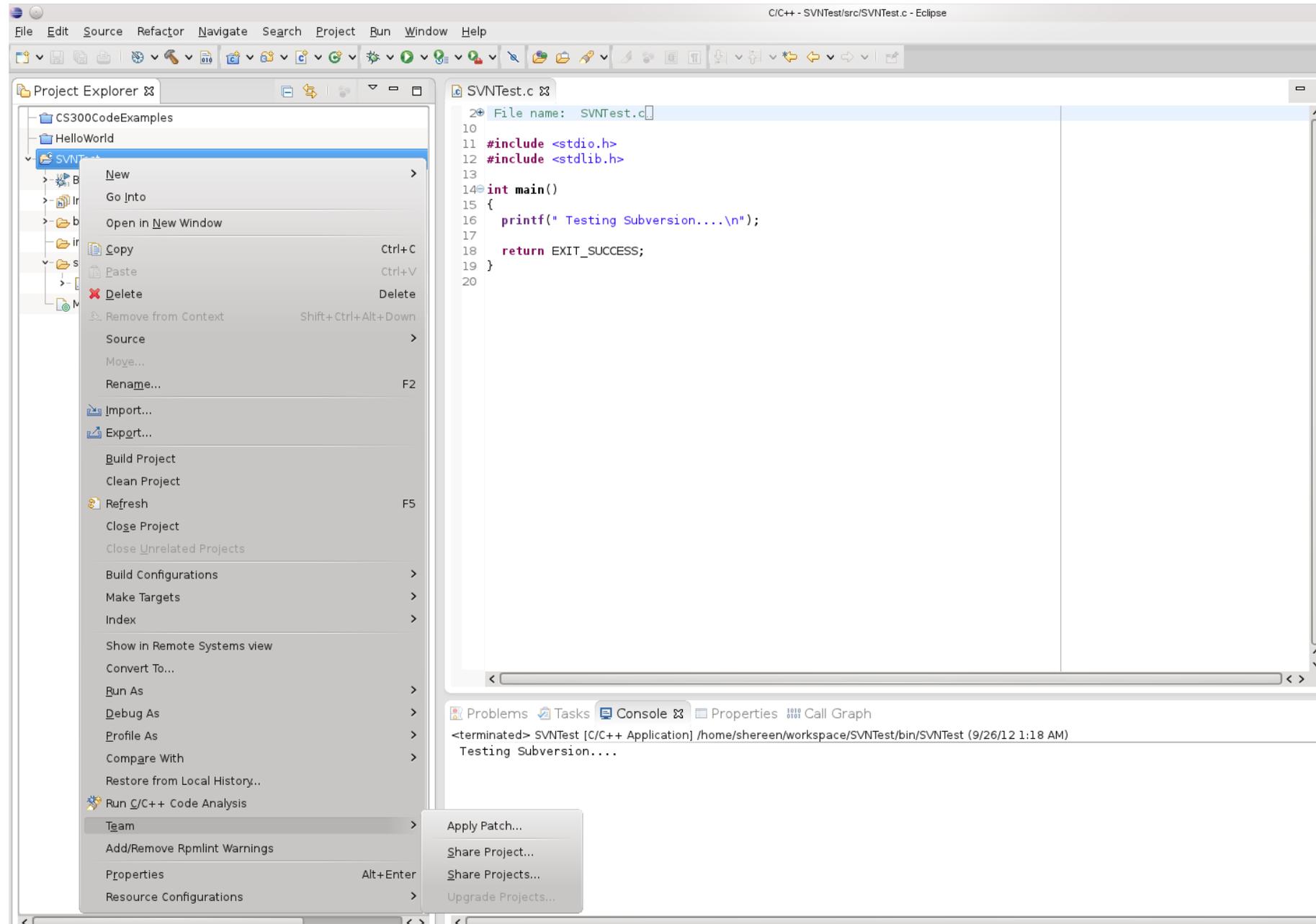
# How to get this to work

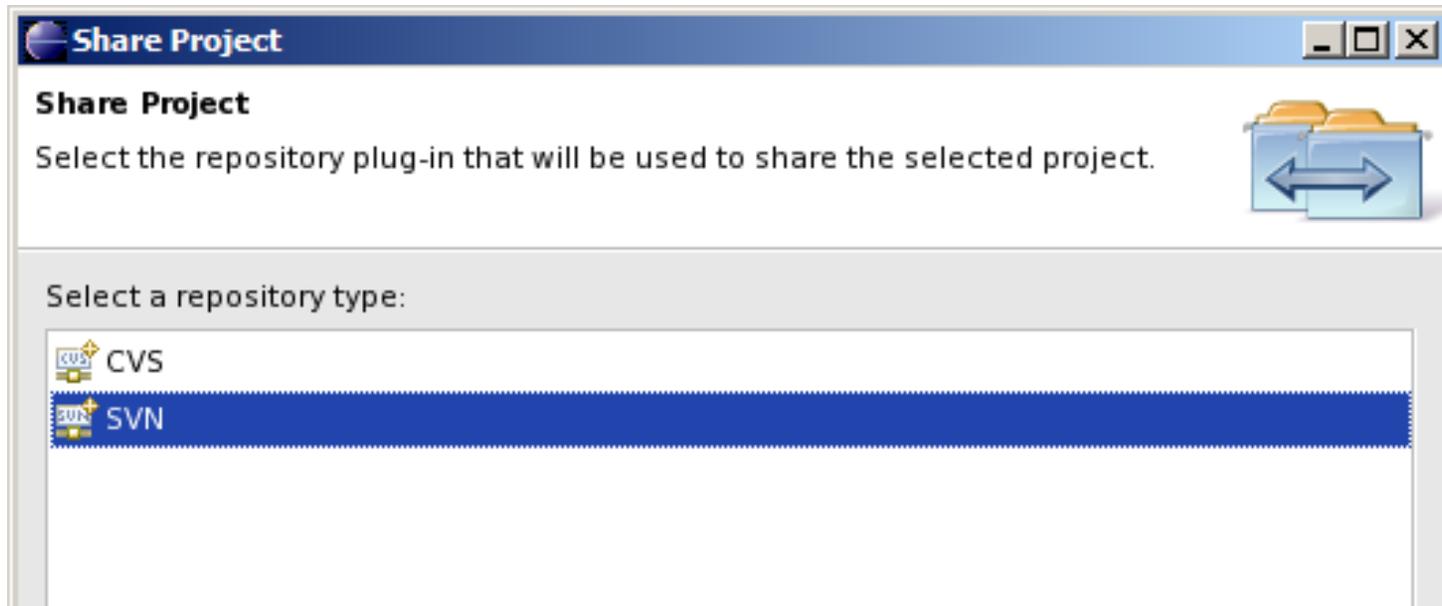
- Create a repository on zeus
  - do this exactly once
  - use this one repository for all your projects
- login to zeus

```
zeus$ svnadmin create /home/shereen/SVNROOT
```

# Check in SVNTest

- Problem: Let's check this project into the repository.
- How?
- Right click on the SVNTest project, then Team, then Share Project.
  - See next slide





Share Project Wizard

### Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.



**General** | **Advanced** | **SSH Settings** | **SSL Settings**

URL:

Label

Use the repository URL as the label

Use a custom label:

Authentication

User:

Password:

Save authentication (could trigger secure storage login)

To manage your security data, please see ["Secure Storage"](#)

Show Credentials For:

Validate Repository Location on finish

Share Project Wizard

### Specify the project(s) location

Specify the project(s) location in the SVN repository



Simple Mode:

URL:

Advanced Mode:

Name on Repository

Use project name

Use empty name

Use specified name:

Project Repository Layout

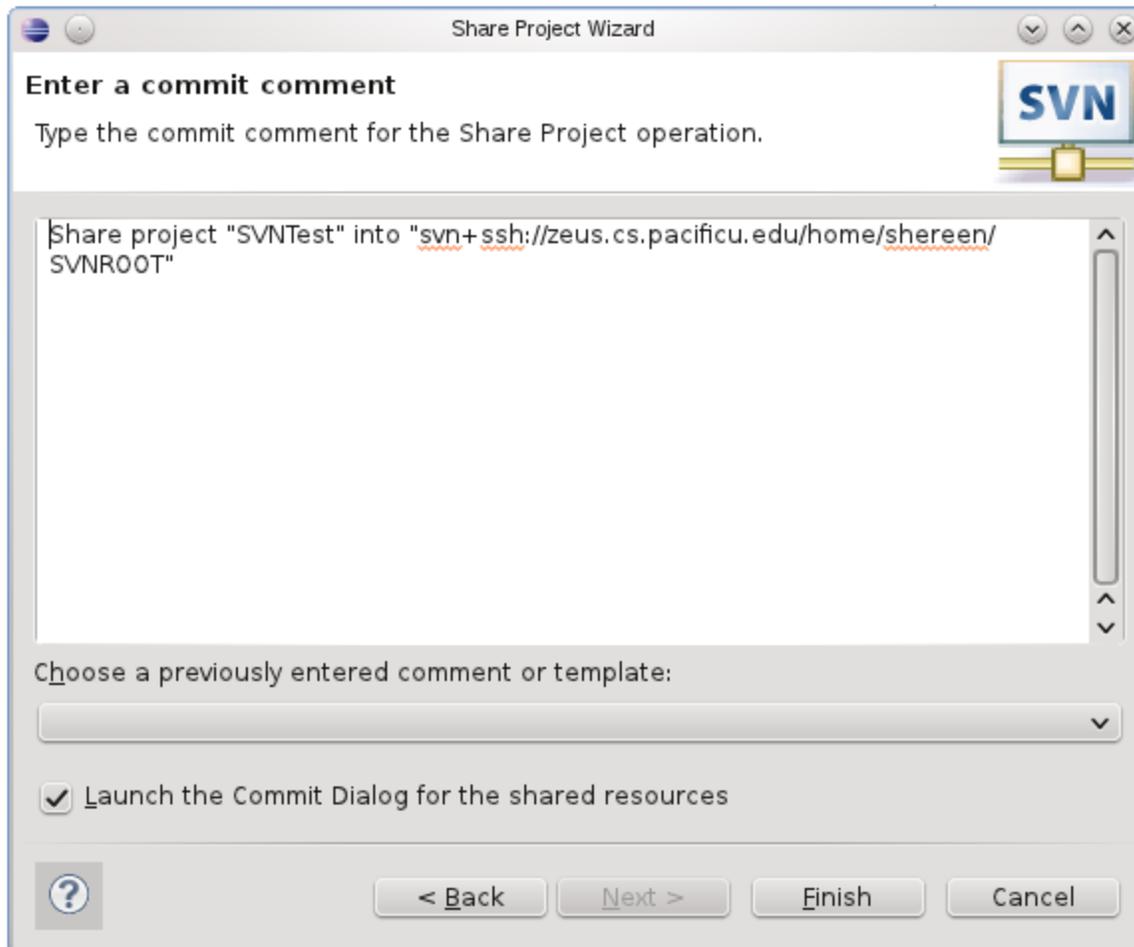
Use Repository Location layout

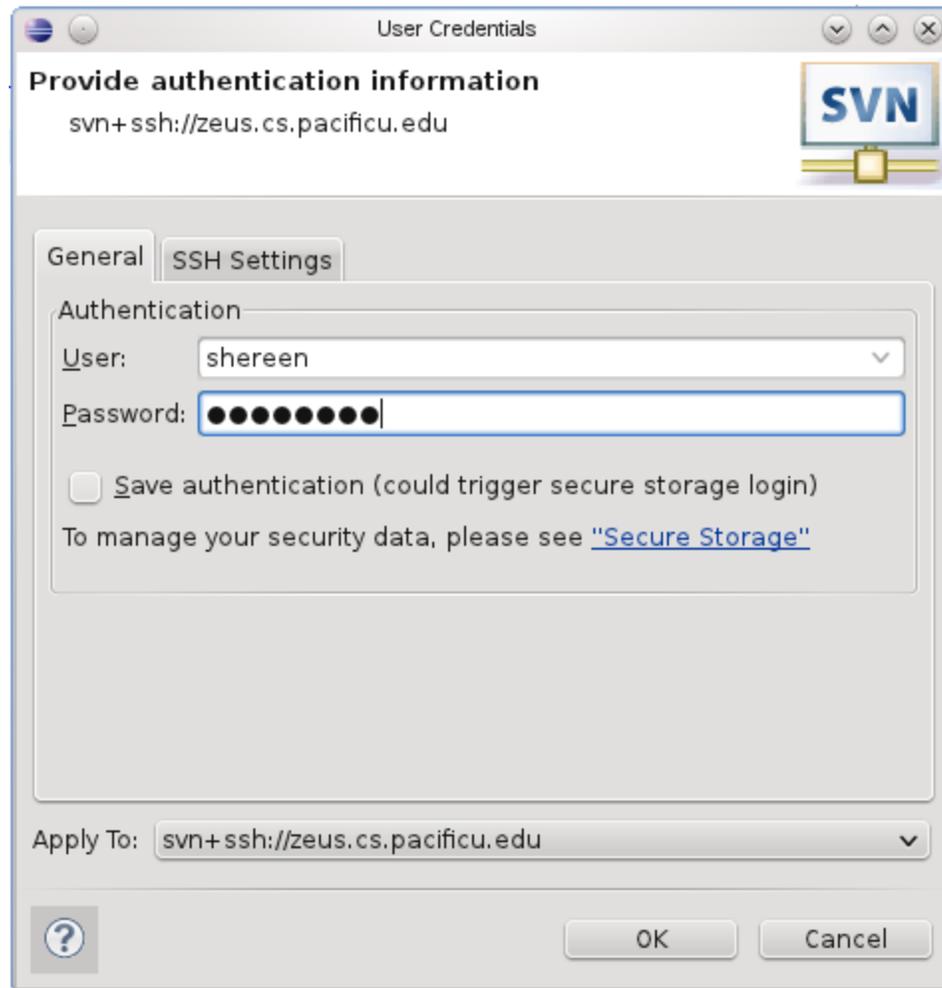
Use single project layout

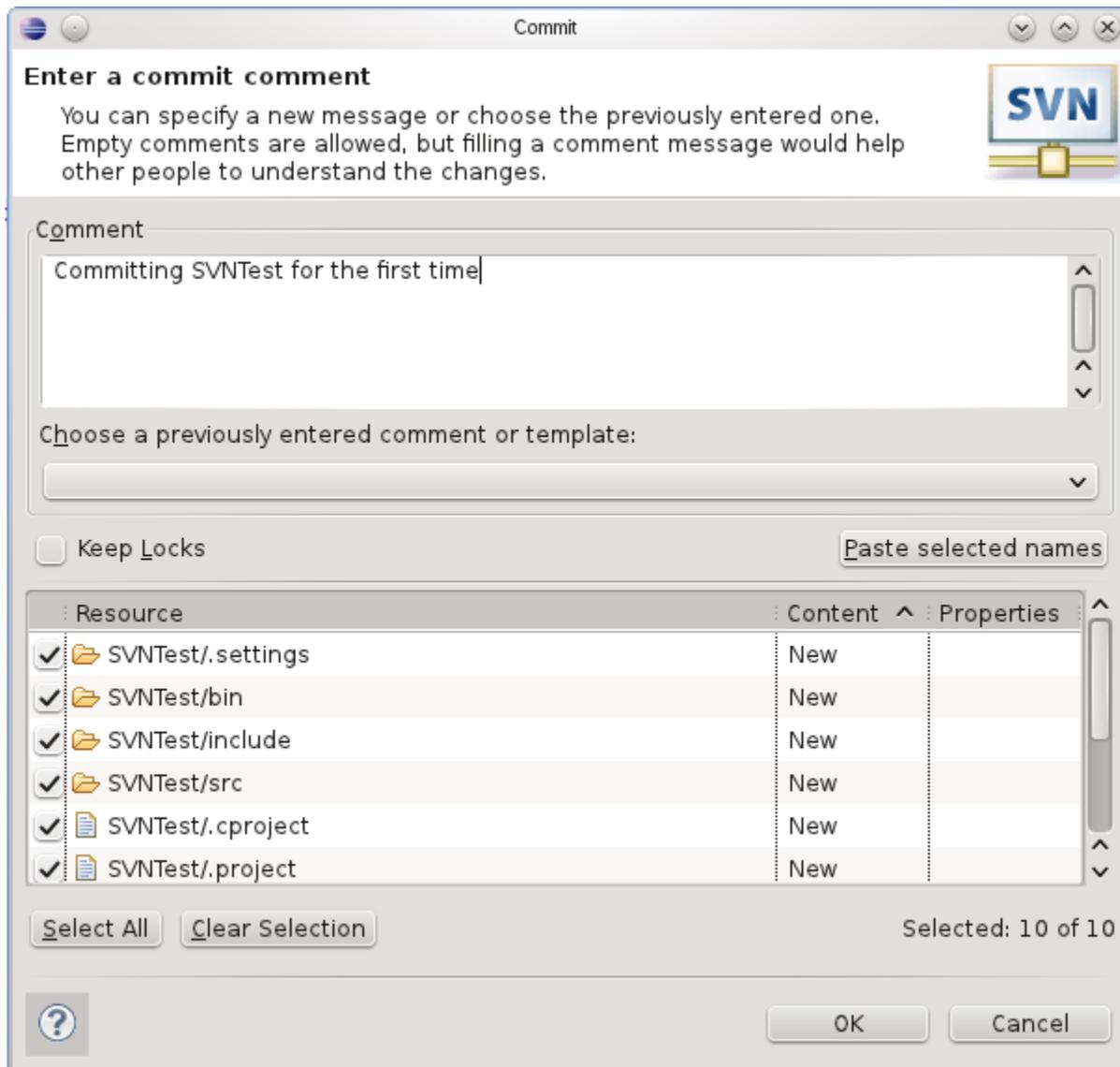
Use multiple projects layout with the specified root name:

Use Subversion recommended layout ('trunk', 'branches' and 'tags')

Project files location on the repository will be different depending on the selected layout type. You can see future files location below:







## How to do a code commit

To commit a project, right click on the project folder → Team → Commit

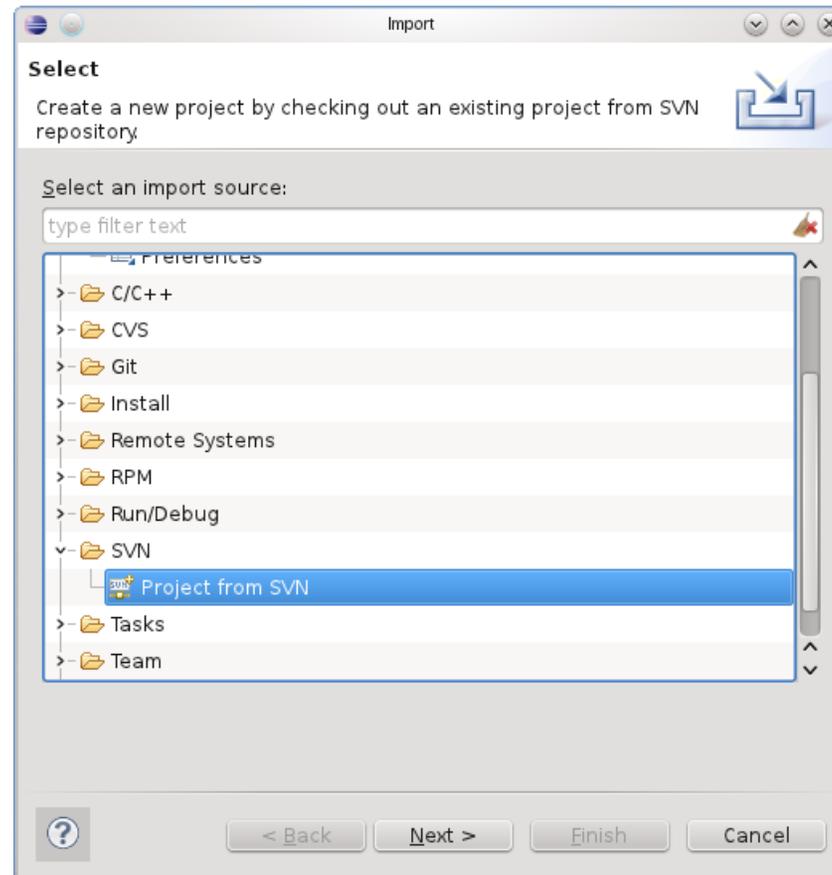
Add very descriptive comments for EACH code commit. You will not be sorry.

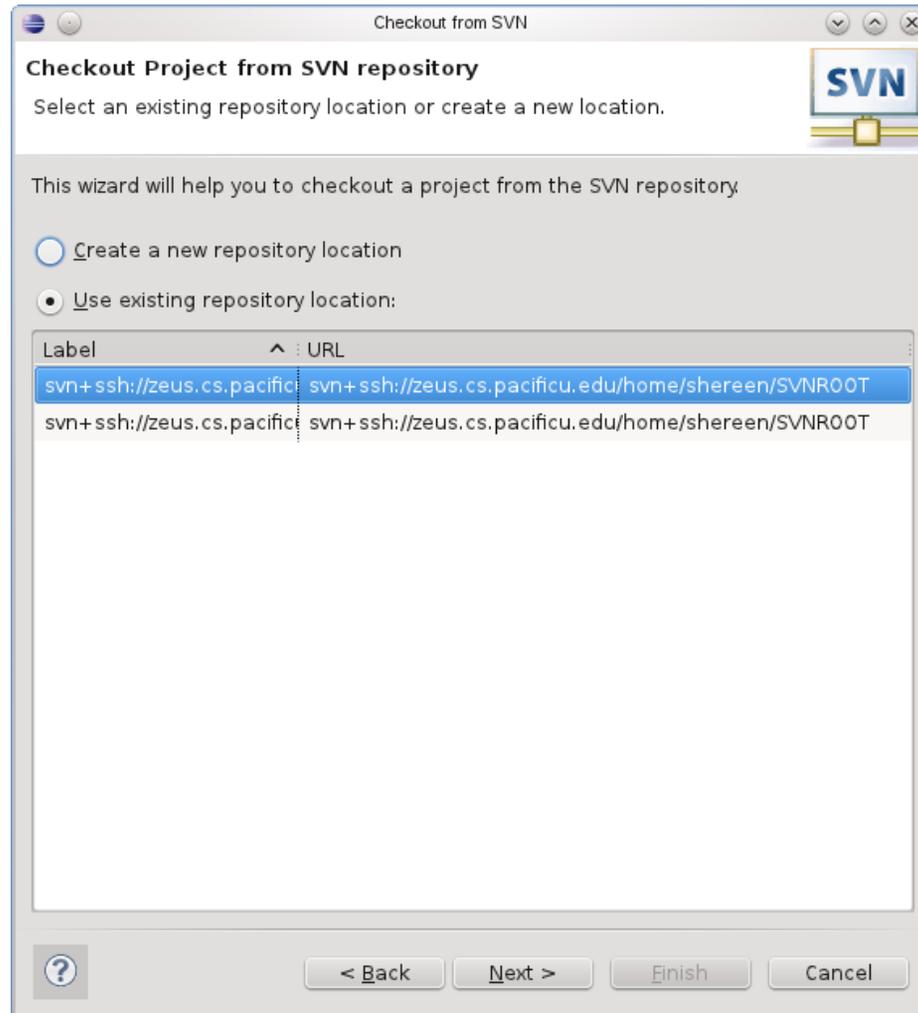
# Let's Delete SVNTest

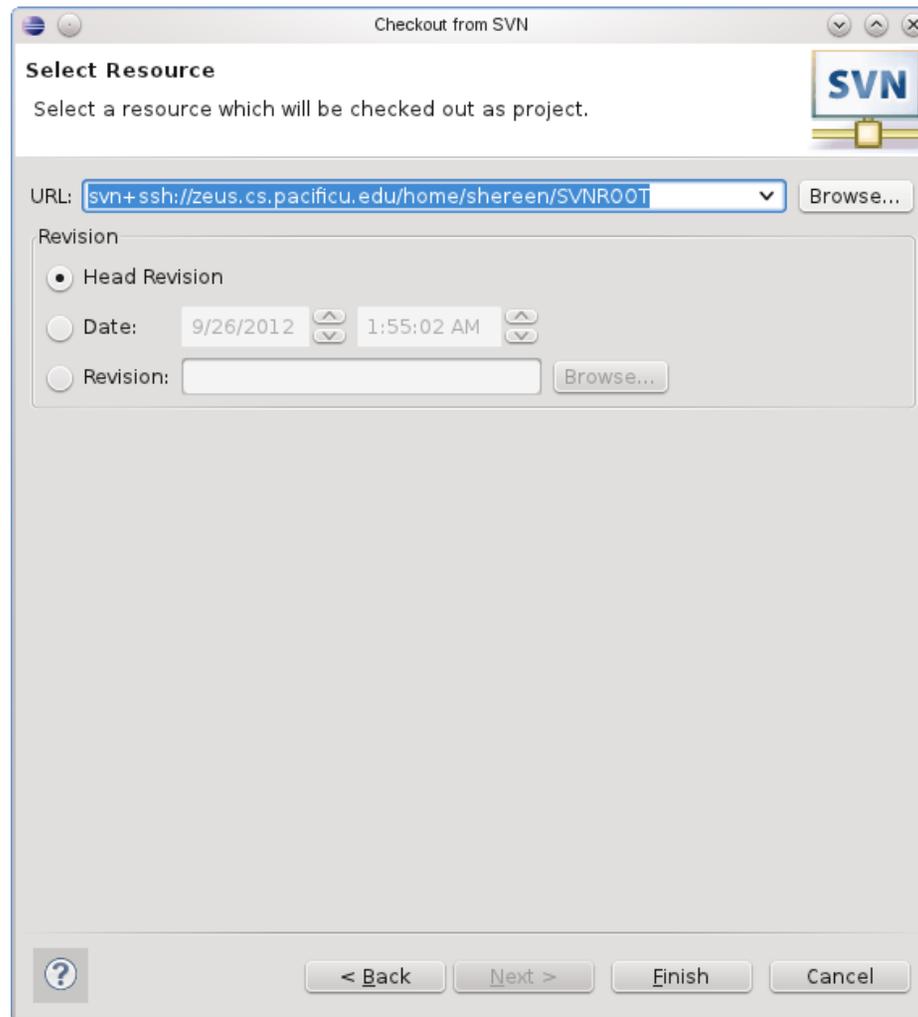
- Right Click the SVNTest project -> Delete
- CHECK → Delete project contents on disk

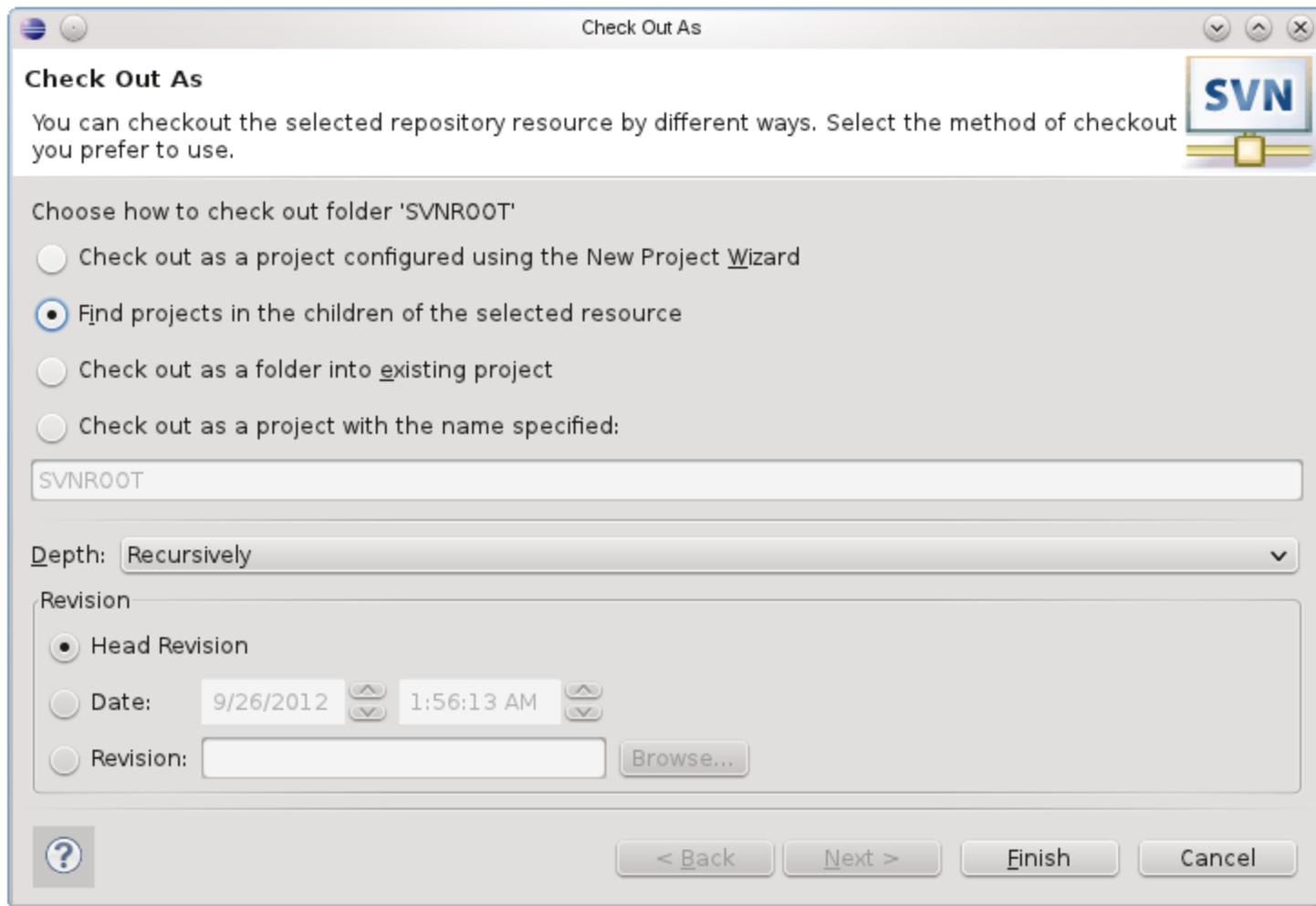
# How to checkout

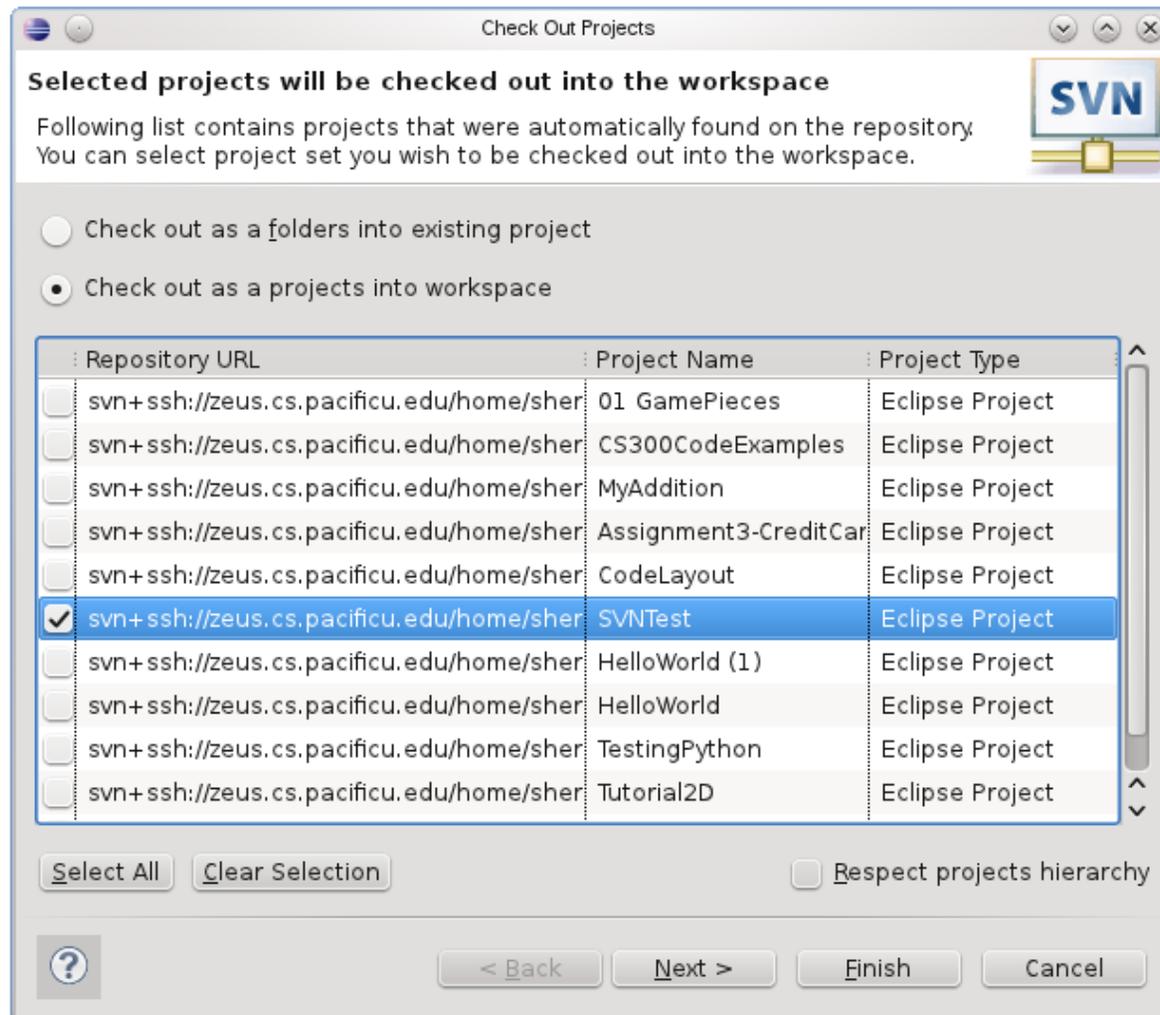
To checkout a project, File → Import, then

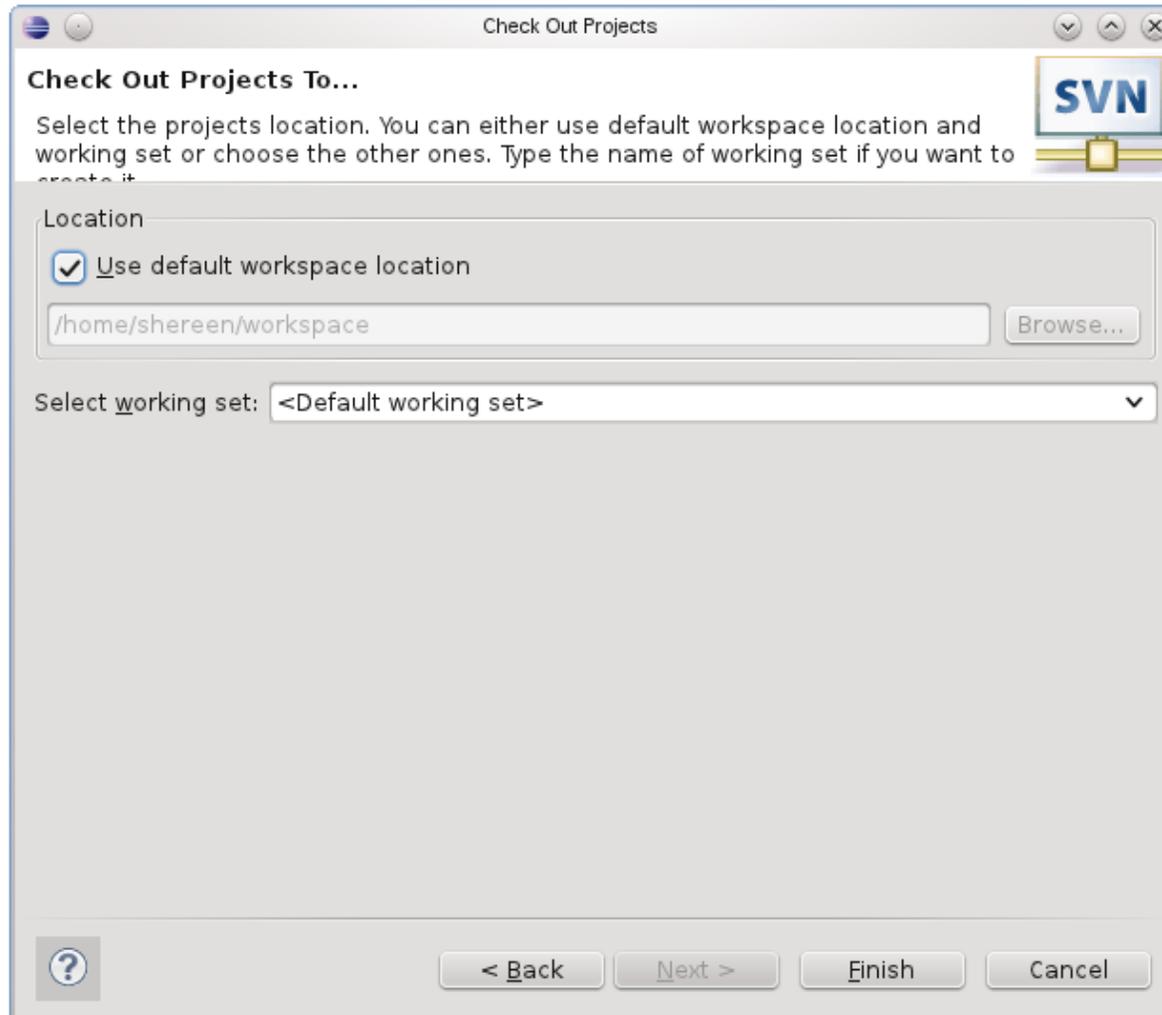












# Make a change in Eclipse

- Add `printf("I love CS 300!\n");` to `main()`
- Build and run (just to be sure)
- Commit to SVN:
  - Right Click SVNTest | Team | Commit
- Do NOT commit .o or executable files!

# Show History

- In Eclipse
- Right Click a File
  - Team | Show History

# When things go bad...

- Let's revert ONE FILE back to before the last change
- Right Click the project to revert
- Replace With | Revision
  - Revisions listed with comments
  - Double-click a revision

# Using Subversion by hand

- Open a single shell prompt
- Create a folder called Junk and change into it
- Check out SVNTest project in Junk directory
- Type

```
$ svn checkout svn+ssh://zeus/home/shereen/  
SVNROOT/SVNTest
```
- Using Geany, add a printf to main().

# Using Subversion by hand

- From a command line, find the Makefile and re-make the project and run it.
- Now commit the changes to the repository by hand.

```
$ svn commit -m "add second printf"
```

- In Eclipse and do an update on SVNTest. Your changes should show up.  
Right Click SVNTest | Team | Update to HEAD

# Check out on Zeus

ssh to zeus.

```
zeus~> mkdir cs300
```

```
zeus~> cd cs300
```

```
zeus~> svn checkout svn+ssh://zeus/home/shereen/SVNROOT/  
SVNTest
```

```
zeus~> cd SVNTest
```

```
zeus~> make clean
```

```
zeus~> make
```

This is how you should test on zeus from now on.