



Getting Started with Subversion 1.3.1

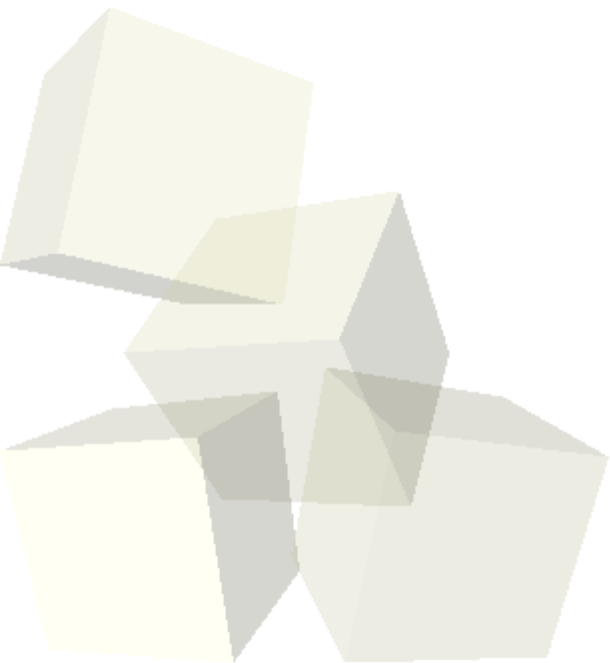
Gary Murphy
Hilbert Computing, Inc.
glm@hilbertinc.com
(913) 780-5051

21-May-2006





- Used to manage versioned copies of text and binary content, such as source code.
- Open-source, royalty-free.
- Heir-apparent to CVS and is quickly becoming the tool-of-choice for open-source development projects.
- Development began in 2000.



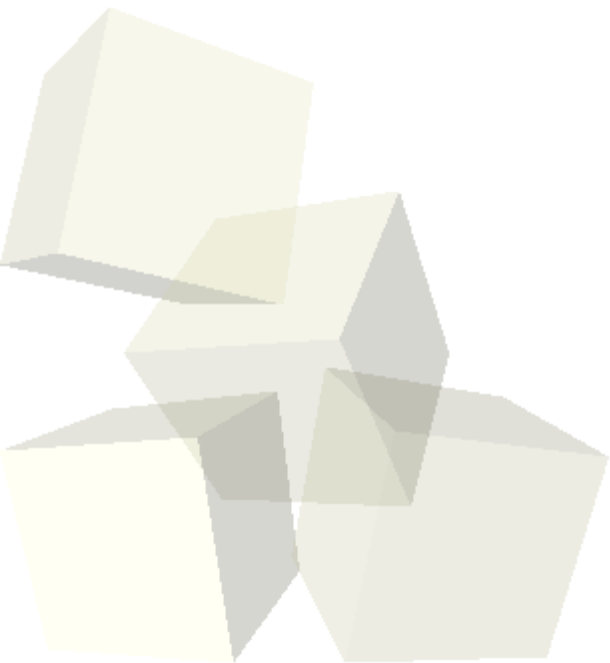


- Files and directories are versioned.
 - ◆ Enables delete, copy and rename to maintain version history.
 - ◆ Eliminates a design flaw in CVS where a same-named file could inherit the history of an old file that was deleted long ago, but has the same name
- Atomic commits
 - ◆ A collection of modifications are committed to the repository (or not) as a unit-of-work
- Versioned Metadata
 - ◆ Users can create a set of name/value pairs that are associated with any file or directory
- Binary and text differences are managed using the same algorithm
- Multiple Network Transports



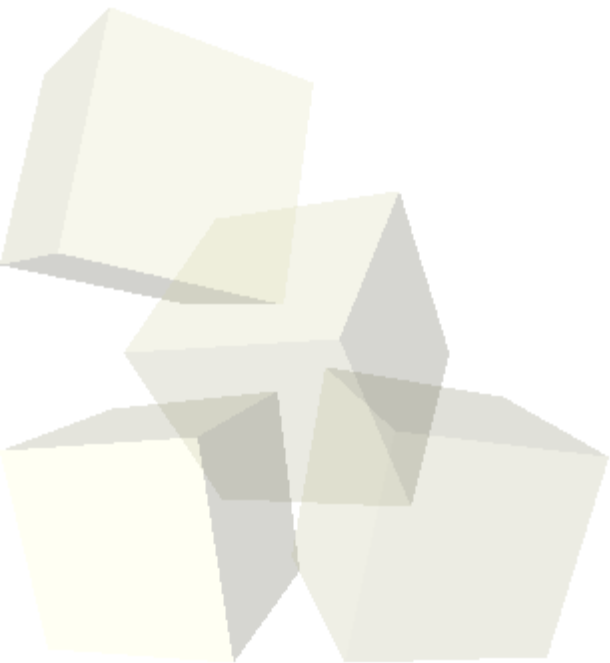
Access Methods

- Access method is determined by the URL
 - ◆ file:// - direct access to local disk
 - ◆ http:// - access via WebDAV. This requires the WebDAV Apache module to the Apache Web Server
 - ◆ https:// - same as http, but encrypted
 - ◆ svn:// - uses a custom protocol to the Subversion server (`svnserve`)
 - ◆ svn+ssh:// - same as svn, but through an SSH tunnel





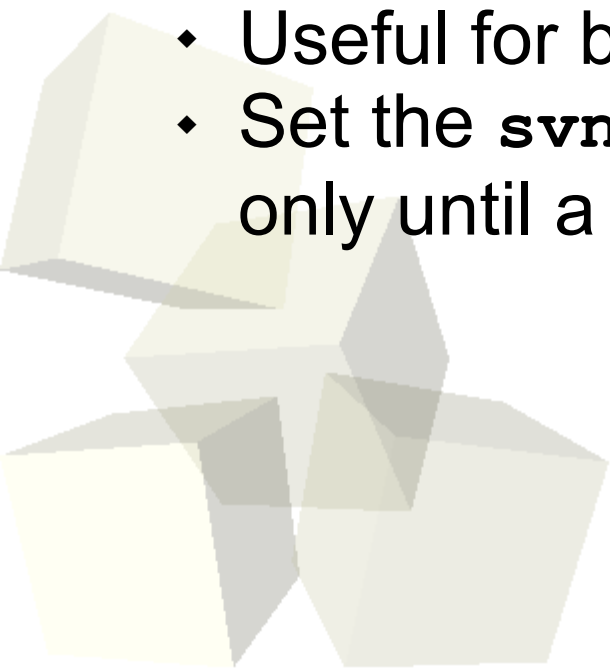
- Versioned files are cached in the administrative area of Subversion (.svn directory).
 - ◆ status, diff, revert can be accomplished while offline
- Allows more efficient commits
 - ◆ A compressed version of the delta is sent across the network.





File Management Strategy

- Uses copy-modify-merge over lock-modify-unlock
 - ◆ Potentially lessens administration. (e.g. developer locks files and goes on vacation.)
 - ◆ Locking approach may give a false sense of security. If two files are dependent, changes to both by different developer can still break code. This is more common with modern development (more, smaller parts).
- Subversion does support locking
 - ◆ Useful for binary files.
 - ◆ Set the `svn:needs-lock` property and it will be read-only until a lock is obtained.





- To create:
 - ◆ `svn create path`
- There is no “project” concept. It's just a virtual versioned filesystem.
- The data is stored in a database or the filesystem
- Supports event hooks
 - ◆ Scripts or executables can be run on events before and after commit, lock, unlock, properties changes and when a commit starts
 - ◆ Could be used to enforce business rules for source code management
- Supports hot backup of the data

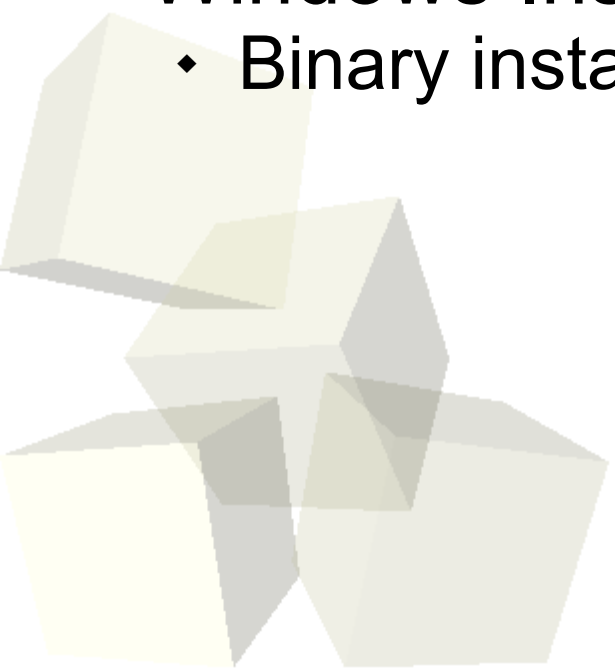


■ Linux Installation

- ◆ Build from source
 - ./configure
 - make
 - make install
- ◆ Some binary distributions available
 - RedHat Enterprise
 - No Fedora Core RPMs were available as of 20-May-2006

■ Windows Installation

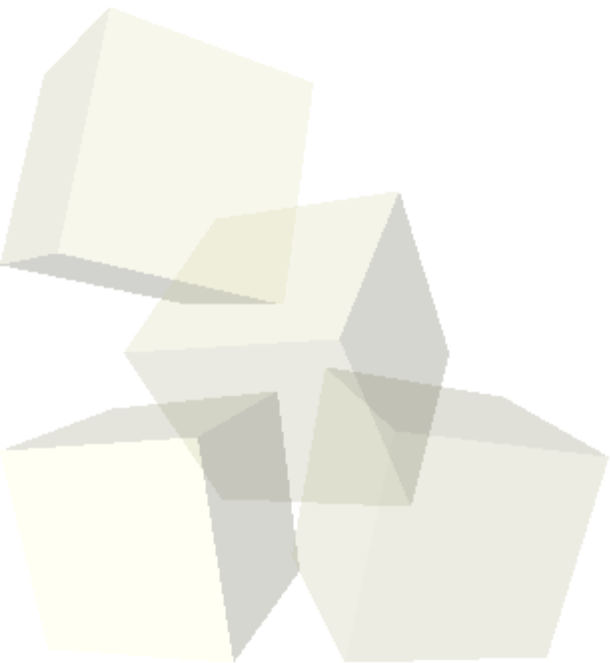
- ◆ Binary installation program available, but not tested





Server Configuration

- Supports
 - ◆ Direct file access (no server)
 - ◆ WebDAV (cleartext or encrypted)
 - ◆ svnserver (cleartext or encrypted)
- SSH Authentication and Access
 - ◆ There is no svnserver daemon running. SSH will invoke the command over the tunneled session.





■ IDE Integration

- ◆ AnkhSVN – addin for Microsoft Visual Studio .NET
- ◆ Subclipse – plugin for Eclipse
- ◆ JSVN – Subversion client and plugin for IDEA
- ◆ JDeveloper – plugin for Oracle's free JDeveloper IDE
- ◆ psvn.el – emacs integration. (Ok, that's not an IDE!)

■ Operating system integration

- ◆ TortoiseSVM – Windows file explorer integration
- ◆ SCPlugin – Subversion plugin for Mac OS X Finder

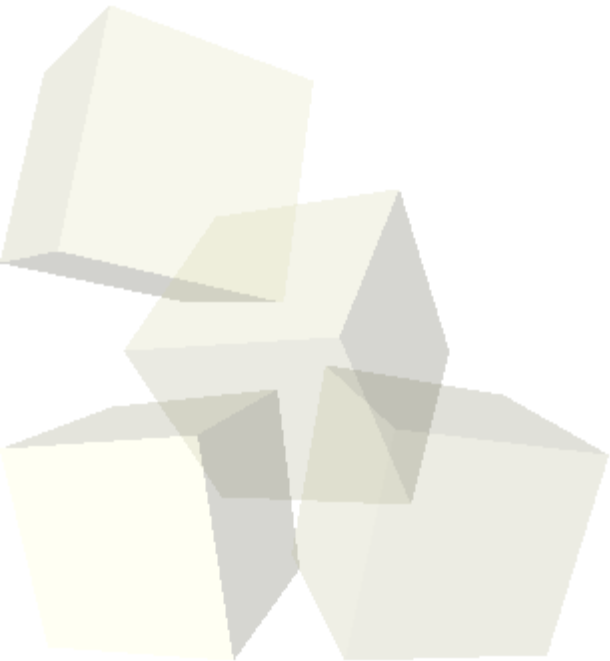
■ Language Bindings

- ◆ Core C API bindings for:
 - Python, Perl, Java, Ruby
- ◆ Client API. Object-oriented classes for:
 - Java, C#, C++, Python, PHP



Repository Conversion

- Conversion tools are available for:
 - ◆ CVS
 - ◆ Perforce
 - ◆ SCCS
 - ◆ RCS
 - ◆ Microsoft Visual Source Safe



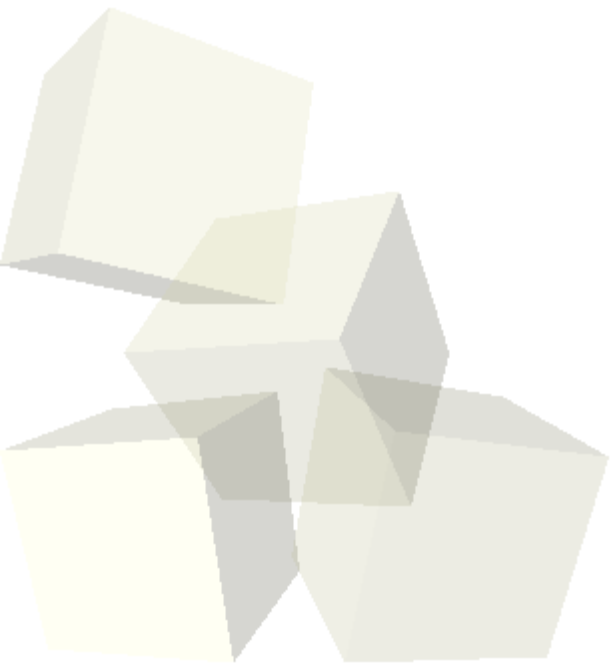


Eclipse Integration (General)

- Interface is similar to CVS plugin
- Plug-in doesn't implement the defacto standard for trunk/tags/branches. More flexible, but CVS developers must remember to do that on create.
- Current Subclipse uses JNI:
 - ◆ Experienced exceptions that crashed Eclipse when I corrupted the repository during test.
- Some “fit and finish” issues:
 - ◆ Have to manually select files on first check-in (one-click, but annoying)
 - ◆ Manually have to configure trunk/tag/branches
 - ◆ Tagging is directory creation. Interface is not intuitive.
 - Browse is broken during tagging

Eclipse Integration (SSH/svnserve)

- “Fit and Finish” Issues:
 - ◆ ssh window pops repeatedly





Advanced Features

- Properties – user-defined name/value pairs that can be versioned with the project files
- Peg revisions – When a project (really directory) name is reused, it can be told that it is unrelated to the previous project with the same name
- Externals – contains references to information in different areas of the same or different repositories within organization
- Vendor branches – Copy vendor code into our project, but maintain linkages to original code so that our mods can be integrated. Valuable to mods to open source projects
- Internationalization support – All repository names are converted to UTF-8