

Troubleshooting FAQ [DRAFT]: UC Berkeley Subversion Repository Service [PILOT]

Table of contents

1	Questions.....	2
1.1	1. Client connections to the repository.....	2
1.2	2. Fixing and re-organizing repository contents.....	4
1.3	3. Importing and exporting projects.....	6
1.4	4. Terms of use.....	7

Questions

1. Client connections to the repository

1.1. I can't connect to my repository (I have never been able to connect).

Here's a checklist to run through, roughly in order of most frequently encountered problem:

- Has an account been set up for you on Subversion, and are you using the correct login id for that account (generally, it's your "friendly" CalNet ID)? If not, please contact the person in your department/group authorized to approve access to the repository you wish to access.
- Have you provided a public key to be forwarded to the Subversion server's system administrators, and has your group received notice that the public key has been associated with your account? If not, please complete this required step of the account-setup process.
- Have you confirmed that your SSH key exchange was successful? A successful key exchange is a prerequisite for connecting to UC Berkeley's Subversion repositories. More information here: <http://ist.berkeley.edu/as/ag/tools/howto/ssh-key-exchange-win.html>
- Is the public key you or your repository's authorizer forwarded the one that is paired with the private key used by your SSH client, and/or specified in your Subversion client? If not, your login won't work. Please make sure that you're using a matched private-public key-pair (i.e., that the Subversion server's system administrators have been provided a public key that is paired with the private key you are specifying in your SSH client).
- Are you using the correct repository URL? Usually it looks something like this: `svn+ssh://your_user_id@svn.berkeley.edu/svn/ist-svn/berkeley/path/to/your/repository`
- Have others in your group connected successfully to the repository?
 - If so, did they use the same repository URL that you tried? If they succeeded with a different URL, can you succeed using the URL they did?
 - If not, continue to the next item in the list
- Are you using a Subversion client that has been properly configured?
 - If you are using the Subversion client recommended by IST (currently Subclipse, running in the Eclipse IDE), you can find some local setup documentation at <http://ist.berkeley.edu/as/ag/tools/howto/subclipse-setup.html>
 - If you are using a different client than the one IST recommends, have others in your group had success using the client? If so, have you confirmed that you configured your client the same way a successful user configured hers/his?
 - If you are having configuration problems with a client you've chosen that is not recommended by IST, perhaps trying a different client -- whether recommended by

IST or not -- is worth a try. If you have a *nix system (including OS X), the command line svn client may be the easiest to use for a preliminary test of your ability to connect to the repository using public key access. If you are using a Windows client, even though these clients are not the ones recommended for general use by IST, we have test-driven them successfully and without a lot of configuration problems; the Cygwin (command line) client is one that some advanced Subversion users in IST use regularly:

- [Cygwin's](#) SVN client (also cf. [local documentation](#)). If public key login is set up correctly, this client should work "out of the box."
- [TortiseSVN](#). To configure using public key login as described in [local documentation](#), navigate to the Network pane in the TortiseSVN Settings dialog and specify the location of the SSH client, `ssh2.exe`. If your private key and identification file are in a non-standard location, you may be required to specify where to find the identification file (which points to your private key) in addition to specifying the SSH client location, using the `-i` option, e.g. (all on one line),

```
C:\Program Files\SSH Communications Security\SSH Secure
Shell\ssh2.exe -i
"C:/Documents and Settings/<user-id>/Application
Data/SSH/UserKeys/identification"
```

- If none of the above steps have resolved your problem, please contact svn-support@lists.berkeley.edu.

1.2. I can't connect to my repository (I was able to connect before)!

If others in your group can connect to (reach and read) the repository but you cannot, the problem probably has to do with your account, and most likely with the key(s) used for authentication. Did you change your local keys? If you are using a different private key, you must have your public key on the Subversion server changed as well - the two must correspond. Cf. [local documentation](#) for guidelines on checking whether your key-exchange is set up properly. If you have eliminated key-exchange problems, please determine whether others in your group (if there are others) are having the same problem you are experiencing, or if you are the only one; then please contact svn-support@lists.berkeley.edu for assistance.

1.3. I can't commit to a project in my repository!

Commit permissions are governed by a member of the IT group that owns the repository (sometimes this responsibility is shared). The configuration file (readable by all users of the repository) is `SVN/commit-permissions/commit-access-control.cfg`, and one or more member of the group is responsible for editing the file (cf. [Subversion](#)

[Repositories: Governing Commit Permissions](#) for more info). A common mistake is to attempt to commit to the root of a project: `svn.berkeley.edu/svn/path/to/repo/myproj/`. The standard and recommended project setup (including standard permissions based on the examples given in the previously-cited page) does not allow this; new and ongoing work is intended to be committed to the project's `trunk/` directory, e.g., `svn.berkeley.edu/svn/path/to/repo/myproj/trunk/`. Cf. [Subversion Repositories: Creating New Projects](#) for more information.

1.4. Subclipse commits fail when using my full name instead of my svn login id

The Subclipse client asks (and optionally saves for re-use without prompting) a user's authentication information (e.g., the location of a private key used for public key authentication) *and* the name of the "author" of changes committed to the repository. The name of the "author" must be the same as the login id. If the Author Name is set to a different value, the standard governance scripts used in UC Berkeley repos will cause a commit to fail with an error such as this (where the Author Name was set to Ghengis Kahn, which is not the login ID on the svn host):

```
[...]  
/svn/berkeley/projects/my/repo/path/hooks/commit-access-control.pl:  
user `Ghengis Khan' does not have permission to commit to  
these paths: [...]
```

If you have not asked Eclipse to remember the Author Name, simply exiting and re-starting Eclipse, then attempting the commit again (supplying the login id as the Author Name) should solve the problem. If you have asked Eclipse to remember the Author Name, you must delete the file `configuration/org.eclipse.core.runtime/.keyring`, located in the Eclipse installation directory. Erasing this file (and re-starting Eclipse) will require you to respecify key file location and Author Name.

1.5. I committed a repository-permissions configuration file, and now no one in our group can access the repository: I can't even commit a "last good version" to the head of our repo. How can I fix this?

Please contact svn-support@lists.berkeley.edu, and identify the revision number of the "last good" version of the configuration file. A repository administrator with shell-login access to the Subversion server will restore the working-copy on the server to the version you specify; please be sure to review the contents of the configuration file carefully for proper syntax before you commit a new version.

2. Fixing and re-organizing repository contents

2.1. I made an error in committing files/directories to a project in my the repository and I need to restore the project to a good working state.

Check out the last "good" copy of your project to a clean directory, then check it in (commit it) to the head of your project. You may need to reconcile differences between the older, "good" copy and the "bad" files at the head of the repository before you will be allowed to commit. Please consult your Subversion client software's documentation, or Subversion's [on-line manual](#).

2.2. I committed a file to the repository that must be completely erased from the repository's history. Is there a way to do this?

This is not a trivial operation in the current release of Subversion, and must be performed by an administrator. If the file is "sensitive" due to inclusion of a password, it is recommended that the password be changed rather than that the data be expunged from the repository. If there is no user-implementable workaround of this nature, and the file(s) must truly be expunged, access to the repository must be shut down while the work is performed. In order to perform this work, you will need to provide either a list of repository paths to keep, or a list of repository paths to expunge; generally, the latter is preferred as it is usually more concise. The Subversion project's [FAQ on removing a file from the repository's history](#) describes and links to further detail on how this operation is accomplished. The person authorized to make changes in the repository must initiate this work by contacting svn-support@lists.berkeley.edu, identifying the required list of repository paths and a time-period during which repository access may be disrupted. Depending on the scope of the work that must be performed by a system administrator, this may be a recharge service.

2.3. I arranged my project(s) in the repository using a directory structure that I want to change. How can I do this?

If you are using a command-line Subversion client, cf. documentation on the svn copy, svn delete, and svn move commands in Subversion's [on-line manual](#). If you are using a different client, please consult its documentation to discover the procedures that effect copy+delete or move functionality.

2.4. I want to tag / branch / merge code in my repository, but I don't know how. What should I do?

Please consult published documentation, either on-line or in hard-copy books. The [Subversion on-line manual](#) is the best reference if you are using the command-line client. Your Subversion client's documentation is the most likely source of useful information if you

are using something other than command-line `svn`. Other resources - some of which are much "friendlier" to GUI-client oriented Subversion users - are linked from [Subclipse usage](#) pages on the [tools page](#) on the IST-AS Architecture website.

3. Importing and exporting projects

3.1. I want to check a project out of my repository without all the `.svn` directories. How can I do that?

You can use the command-line `svn export` command to perform this task. Note that `svn export` does NOT create a "working copy" and should be used only in cases where you do NOT need or want to modify files and check them back into the repository (e.g., this command is often appropriate for deployment of files from a repository to a web or application server where a full, fresh checkout is performed as part of an automated/scripted deployment task). Windows users can install Cygwin (Unix shell for Windows) and its command-line Subversion client by following [local instructions](#). Command-line Subversion clients are available for most Unix operating systems; in many cases the binaries are included in standard installations (`svn`).

3.2. I want to move one or more projects from one repository to another on UC Berkeley's Subversion host. How can I do this?

You will need to supply the repository path for the old and new project(s)' location(s). This task requires intervention of an administrator who has shell access to the Subversion server. Please contact svn-support@lists.berkeley.edu to schedule this work. Depending on the scope of the work that must be performed by a system administrator, this may be a recharge service.

3.3. I want to move one or more projects - including history - from a Subversion repository somewhere other than on UC Berkeley's Subversion host to one that is hosted on `svn.berkeley.edu`. How can I do this?

You will need to produce a dump file (using `svnadmin dump`) from your old repository that will be used as input to `svnadmin load` on the UC Berkeley Subversion Host. You will need to provide a repository URL to which the data should be loaded (e.g., `svn.berkeley.edu/svn/berkeley/path/to/my/repo/my-migrated-project/`). The load task must be performed by an administrator who has shell access to the Subversion server. Please contact svn-support@lists.berkeley.edu to arrange for file transfer and to schedule this work. Depending on the scope of the work that must be performed by a system administrator, this may be a recharge service.

3.4. I want to move one or more projects - including history - from a CVS repository to a Subversion repository on UC Berkeley's Subversion host. How can I do this?

There are a couple of ways to accomplish this. One way is for you to produce an SVN dump file by running [cvs2svn](#) (if the CVS repository is on a server running both SVN and CVS). Then, see FAQ regarding moving projects from one SVN repository to another, above. Another way is to supply a tarball of the CVS repository (which may be a pruned copy of the repo, including only the projects that are to be migrated) to IST, which will allow an IST Subversion administrator to run [cvs2svn](#) using the command line options or options-file that you supply. If you are not confident of your ability to run [cvs2svn](#) or specify how to run the utility to obtain the results you want, IST can work through the conversion with you. In either case, please contact svn-support@lists.berkeley.edu to schedule this work. Depending on the scope of the work that must be performed by a system administrator or other IST staff, this may be a recharge service.

3.5. I want to move one or more projects from another repository to a Subversion repository on UC Berkeley's Subversion host, but I don't care about history: I just want to put the current state of the project(s) into Subversion - i.e., the "head" of the repository trunk. What should I do?

Use an exported copy of the project(s) - one without control files/directories, such as `/CVS` or `/.svn`. Then import the project to your repository as you would any other collection of new files. Local documentation on [creating new projects](#) in the centrally hosted Subversion repositories may prove helpful.

4. Terms of use

4.1. How may UC Berkeley community members use Subversion repositories?

See the Terms of Service document, at ...

Note:

Insert Link here...

4.2. May I put software (executable code and/or source) that is licensed for a single person or department's use on a network accessible to others?

Generally not, unless the terms of the license allow such publication. Practically speaking, this means that licensed software may not be put into a Subversion repository provided as a service by IST unless access to that repository is restricted to licensees and/or system

administrators who understand and abide by license terms when checking-out and/or installing the software.

4.3. Once I have purchased software, why can't I do whatever I want with it?

Making copies of copyrighted materials is one of the exclusive rights of the copyright owner. When purchasing commercial software, the end-user generally purchases the right to use the software, not the copyright itself. You may only acquire rights in the owner's copyright by permission or in a license agreement. Copyright rights in software are rarely sold to end users like tangible property; they exist in the copyright owner, not in the individual tangible copies of the software. The software can only be used in ways that do not infringe upon the rights that the owner of the copyright has retained.

4.4. What about open-source software?

Open-source software is generally made available under the terms of a license (e.g., the GNU General Public License or the Apache Public License). Such licenses often allow copies of code to be freely made. Usage of the software (source or executable) published under such licenses must, however, fully comply with the licenses' terms.

4.5. If this FAQ doesn't answer my questions, who can provide more information about technology use policy for UC Berkeley community members?

Send inquiries about information technology use policy at the University of California, Berkeley to: itpolicy@berkeley.edu.