

Subversion Repositories: Creating New Projects

Table of contents

1 About standard project layouts.....	2
2 Creating a new project in a repository.....	3

1. About standard project layouts

The convention IST-AS recommends for a project's directory layout, appropriate for importation using `svn import`, resembles the following:

```

/path/to/repository/myproject/
/path/to/repository/myproject/branches/
/path/to/repository/myproject/tags/
/path/to/repository/myproject/trunk/
/path/to/repository/myproject/trunk/my-project-dir/
/path/to/repository/myproject/trunk/my-project-dir/my-project-file.txt
/path/to/repository/myproject/trunk/my-project-other-dir/
/path/to/repository/myproject/trunk/my-project-other-dir/my-project-other-file.xml

```

Note that the `trunk/` directory under the project's root directory is where the project's directories and files should be located.

This project layout is not a technical requirement of the repository, but is a conventional Subversion practice recommended by the IST-AS Architecture Group; and, for the sake of uniformity, it is therefore very strongly recommended for *all* projects in UC Berkeley's Subversion repositories. While described in [Planning Your Repository Organization](#) in the [Subversion Manual](#), the following may be a more succinct and helpful discussion of this convention [quoted from the [Subclipse](#) (Subversion client for Eclipse) [documentation](#) (navigate to Tasks : Maintaining Branches):

Unlike CVS (which supports tags/versions and branches as a direct mechanism) the support for branches and tags are a "by-product" of the SVN repository model, where copying is cheap and where the addressing of specific resources is simplified. The usual SVN repository setup groups the repository into three areas:

trunk

the main line of development, the leading edge, where developers usually commit their regular changes.

tags

a collection of snapshots of the trunk (or a branch) at a user-defined point in time, for instance release day.

branches

active variations of the project compared to the trunk (or even another branch). This is useful when working out a (big) change that needs to be coordinated in a team or when using a single working-copy for the work with the changeproject is not feasible.

While this convention roughly matches the equivalent CVS constructs used for tagging (versioning) and branching, the interpretation is entirely up to you (and your team), and the

process of defining and setting up your repository should take this into account.

2. Creating a new project in a repository

Creating a new project in a repository can be accomplished from the command line (on a Windows box, this is likely to be a Cygwin command line) or from a GUI client, such as Subclipse. (You may wish to refer to [local instructions](#) for Cygwin installation and configuration.)

Using Subclipse or other GUI clients to create projects in a repository (per the instructions given in the plug-in's [documentation](#)) requires additional steps to conform to the conventional directory layout described below. This is not recommended. For those who must use the Subclipse tool to create new projects, follow the plug-in documentation; then, after importing the project files to the `myproject/trunk/` directory as described, add `myproject/tags/` and `myproject/branches/` directories from the SVN Repository Exploring perspective (right click on the `myproject/` directory and choose `New : New remote folder`).

The reference for command-line instructions that allow addition of projects are described in the [Getting Data into your Repository](#) section of the [Subversion Manual](#). The simplest way to create a new project is to create the project's directory structure on a filesystem, then use the [svn import](#) command. **Simplified, recommended instructions for performing these steps to create new repositories follow...**

- Create a set of "template" directories with a README.txt file in the `trunk/` subdirectory. If the template is a directory called `commonspj`, its contents should look like this:

```
$ ls -R commonspj
commonspj:
branches  tags  trunk

commonspj/branches:

commonspj/tags:

commonspj/trunk:
README.txt
```

The README.txt file might contain text like this: This is a UCB Commons project.

- From the directory containing the `commonspj/` directory, issue the `svn import` command. Example (all on one line!):

```
svn import commonspj
```

```
svn+ssh://mylogin@svn.berkeley.edu/svn/berkeley/projects/commons/my/new/project/path  
-m "original commit"
```

The new project should now exist in the repository
/svn/berkeley/projects/commons.