

Subversion, Eclipse and java.net

Christelle Scharff and Guillaume Maillard

[This handout explains how to use Eclipse with Subversion. It also shows how to use the java.net Subversion capability. The code that will be developed during the course of the project will be made available in the source code repository of your java.net project. The code of the repository can be shared, browsed, compared with local code and accessed by committing code and checking out source code with a Subversion client in Eclipse.]

Eclipse Europa installation

See: <http://www.eclipse.org>

Subclipse installation

Subclipse is the Eclipse plug-in for Subversion.

See: <http://subclipse.tigris.org/install.html>

SVN code repository configuration

- Window / Show View / Other / SVN Repository
- Right click / New / New repository location
- Enter the URL of the code repository: <https://XXX.dev.java.net/svn/XXX/trunk> where XXX is the name of the java.net project you are using (e.g., multilib2008-usa1, multilib2008-usa2, multilib2008-india, multilib2008-thailand, and multilib2008-cambodia).
- Click finish
- Enter your java.net username and your password

Getting started

Create a shared project

- Create your Java project as usual
- Right click on it in the navigator view and choose Team / Share project / SVN
- Select the SVN code repository where you want to share your project

OR

Use a shared project

- Go to the SVN repository view and click on the repository to see the list of available shared projects, right click on the one you want to check out and choose “Checkout as project”. The project will then appear in your Java perspective / Explorer view.

Typical subversion lifecycle

The typical Subversion lifecycle is the following:

1. **Check out** a project from a repository. This operation copies the files from the repository to your eclipse workspace (called local copy).
2. In the project directory, **write code**.
3. **Update** (Team / Update) your local copy from the repository, picking up changes your team members may have made since your last update, and *go to step 2*. When you're ready to commit your additions and you have resolved the conflicts, click on Team / Mark Resolved and *go to step 4*.
4. **Commit** (Team / Commit) your changes to the repository. This operation sends the modifications you made to the repository. *Go to step 2*.

Comparing files

At any time, you can see the differences between the file you are considering and the current corresponding file of the repository by right clicking on a file (resource) in the navigator and select Compare / Compare With.

Team synchronization

Team / Synchronize With Repository permits you to see the differences between the local files and the current state of the repository.