

## Professional Development with Java

### Hands-On Sessions #3 and #4

The goals of this hands-on session are to (1) set up and use a RCS repository and to (2) set up and use a continuous integration server. This CI server will use the RCS repository.

The application that will be added to the RCS repository and, then, which will be continuously integrated is the one that was developed during hands-on session #2.

The following material is required for this session:

- Subversion Edge 1.3.1 (CollabNetSubversionEdge-1.3.1\_linux-x86.tar.gz)
- Java 6 (jdk-6u23-linux-i586.bin)
- Eclipse Helios Service Release 1 (eclipse-java-helios-SR1-linux-gtk.tar.gz)
- Subclipse 1.6.17 (site-1.6.17.zip), which is an Eclipse plugin providing Subversion support
- Ant 1.8.2 (apache-ant-1.8.2-bin.zip)
- Hudson 1.348 (hudson.war)

### Subversion

**Note:** In a production environment, Subversion would run on a dedicated server which is backed-up. Also, Subversion would run behind an Apache HTTP (or similar) front-end. To keep this session simple, Subversion will be run on local machines and its file system will be accessed directly, which is clearly one of the worst bad practices you can think about.

### Java, Subversion, Eclipse and Subclipse installation

Install these software packages as follow:

- Install Subversion by uncompressing CollabNetSubversionEdge-1.3.1\_linux-x86.tar.gz in your home folder. This creates the ~/csvn folder. Add ~/csvn/bin to your PATH.
- Install Java by copying jdk-6u23-linux-i586.bin to your home folder and by running the following command: ./jdk-6u23-linux-i586.bin. This creates the jdk1.6.0\_23 folder. Add ~/jdk1.6.0\_23/bin to your PATH and define a new JAVA\_HOME environment variable which points to ~/jdk1.6.0\_23.
- Install Eclipse by uncompressing eclipse-java-helios-SR1-linux-gtk.tar.gz in your home folder. This creates the ~/eclipse folder. Add it to your PATH.
- Install Subclipse from within Eclipse:
  - Start Eclipse, go to **Window > Preferences > Install/Update > Available Software Sites** and disable all software sites.
  - Then, go to **Help > Install New Software... > Add... > Archive...** and select the site-1.6.17.zip archive you have been provided with. Set the **Name** to **Subclipse 1.6.17** and run the installation. When asked, pick the **Core SVNKit Library** and **Subclipse** features.

### Creation of a local Subversion repository

Create a local Subversion repository as follow:

- Create a new Subversion repository in your home folder by running the following command:  

```
svnadmin create ~/axe-repository
```
- Create the following folders: ~/axe, ~/axe/branches, ~/axe/tags and ~/axe/trunk. This

represents a standard Subversion repository structure. *To which RCS concepts do these folders refer to?*

- Import this structure into the repository (and, once done, delete ~/axe which is of no use anymore) by running a command similar to this one:

```
svn import ~/axe file:///home/users/hcallahan/axe-repository -m "svn repository setup"
```

Notice the file:///home/users/hcallahan/axe-repository URL which is used to locally access your Subversion repository. This is, again, a worst practice.

- You can now import your cleaned (no .class files, etc.) Java projects (the ones from hands-on session #2) by running commands similar to the following one (note that this could have been done during the previous step):

```
svn import ~/eclipse-workspaces/axe/axeModel file:///home/users/hcallahan/axe-repository/trunk/axeModel -m "initial import"
```

- The repository is now created and loaded, it can then be used from Eclipse.

## Using the Subversion repository from Eclipse

For this part of the hands-on session, you'll need to use the two following Eclipse perspectives:

- **Java** is the perspective you'll use to play with your application, as you previously did in the other hands-on sessions.
- **SVN Repository Exploring** is the perspective you'll use to play with your Subversion repository.

Do the following:

- Start Eclipse with a new workspace (that is, with no projects)
- From the **SVN Repository Exploring** perspective, add the Subversion repository you've just created using its location (e.g. file:///home/users/hcallahan/axe-repository)
- Browse the repository to ensure everything is at the place it should be. If it's not the case, fix it, possibly by recreating the repository from scratch.
- From the **Java** perspective, import the projects that are in the Subversion repository: Go to **File > Import... > SVN > Checkout Projects from SVN**. Once the projects are imported, notice the badges that are displayed on top of the standard icons...
- The goal is now to play with RCS features: Right-click on the various resources in the **Package Explorer** view, go to the **Team** menu and explore the actions available.

Particularly, you are requested to:

- Bring changes to the source code and commit these changes;
- Create new classes, add them to the repository (and commit them);
- Revert changes to an earlier revision;
- Create a branch (in file:///home/users/hcallahan/axe-repository/branches/) and bring changes to the source code;
- Bring other changes to the trunk;
- Merge the branch back into the updated trunk.

## Hudson

**Note:** In a production environment, Hudson is normally intended to be run on top of a servlet container, such as Apache Tomcat. To keep this session simple, Hudson will be run using its embedded servlet container (Winstone).

## Ant installation

Install Ant by unzipping apache-ant-1.8.2-bin.zip into your home folder. This creates a ~/apache-ant-1.8.2 folder. Add ~/apache-ant-1.8.2/bin to your PATH.

## Using Hudson

If you need help regarding Hudson, refer to the Hudson wiki at <http://wiki.hudson-ci.org/display/HUDSON>.

Before starting Hudson, you need to define the `HUDSON_HOME` environment variable:

```
export HUDSON_HOME=~/.hudson
```

- Start Hudson by running the following command:  

```
java -jar hudson.war
```
- Using a Web browser, go to <http://localhost:8080/>  
If the port is already bound, use the `--httpPort` option to define another port (e.g. `--httpPort=9080`)
- Take time to discover the Hudson UI, its apparent functionalities and the `HUDSON_HOME` folder.
- Go to **Manage Hudson > Configure System** to set:
  - A JDK installation;
  - An Ant installation.
- Go to **Manage Hudson > Manage Plugins > Advanced > Upload Plugin** to install the Cobertura plugin (`cobertura.hpi`). Once the plugin is installed, you need to restart Hudson.
- Create a new free-style job. This job will:
  - Checkout/update the projects from the Subversion repository  
(`file:///home/users/hcallahan/axe-repository/trunk`).  
Tip: Set the **Local module directory** field to `'.'`.
  - The Subversion repository will be polled every minute.
  - Use an Ant build step to build, test and run the test coverage (that is, what you did in the previous hands-on session).
  - Publish JUnit test result reports so that they get aggregated by Hudson.
  - Publish Cobertura coverage reports the same way.
  - Once the job is fully set up and works fine, bring changes to your source code from Eclipse and observe that Hudson runs builds as expected.