

SETTING UP AND USING VERSION CONTROL

This document describes the steps for creating and using a version control repository for the term project of the Database Management Systems course. It is assumed that group members are using the Eclipse IDE. You can download the “Eclipse IDE for Java Developers” package from the Eclipse site. For Mercurial support, you need the MercurialEclipse plugin which can easily be installed from the Eclipse Marketplace (in Eclipse, select “Help → Eclipse Marketplace...”). Installing the “HTML Editor (standard)” plugin is also recommended.

The project repository will be kept on the Pikacode hosting site (<http://pikacode.com/>). Every group member needs to get an individual account from this site. The username for the account should be the same as the ITU username, if possible.

In addition to the individual accounts, a group will also create a group account which will be used for setting up the repository. The group account should not be used for later operations. The username for the group account has to be in the format “itudbYYNN” where YY is the last two digits of the current academic year and NN is the group number assigned in the course. So, for example, if we are in the Fall 2013 semester, the username for group 01 must be “itudb1301”.

Repository Setup

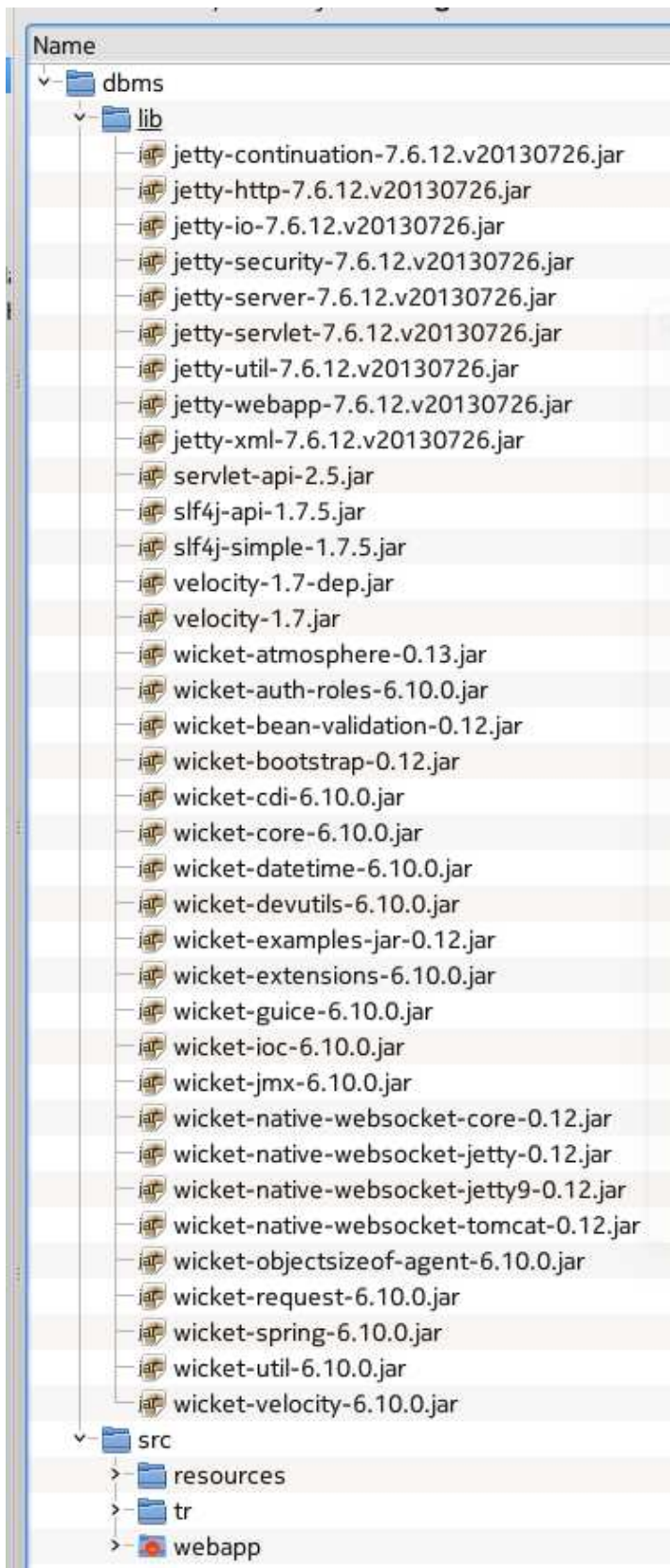
After all members of the group have created their individual accounts, one of the members has to perform the following steps:

On Pikacode:

- Create the group account. Log in to the site using this group account and create a repository. Type the short name of the project as the name of the repository and select “mercurial” as the source control management tool.
- Select the repository, go to the “Settings” page and select the “Members” tab. Add all group members to the repository (including your own individual account). Make sure that all members have push permission.

On your computer:

- Download the following files:
 - <https://dl.dropboxusercontent.com/u/61579540/dbms.zip>
 - <https://dl.dropboxusercontent.com/u/61579540/wicket.zip>
 - <https://dl.dropboxusercontent.com/u/61579540/slf4j.zip>
 - <https://dl.dropboxusercontent.com/u/61579540/jetty.zip>



- Extract the dbms.zip file anywhere on your computer. Extract the wicket.zip, slf4j.zip, and jetty.zip files under the folder where you extracted the dbms.zip file. The resulting folder hierarchy should look as in the figure.

In Eclipse:

- Select "File → Import... → General → Existing Projects into Workspace" and select the root directory of your project ("dbms") as the root directory.
- To initialize the repository, in the package explorer, right click on the project name ("dbms") and select "Team → Share Project... → Mercurial".
- To add the initial files to the repository, right click on the project name and select "Team → Commit". In the commit form, type "added initial files" into the top text box and check the "Select/unselect all" box to select all files.
- To push the changes to PikaCode, right click on the project name and select "Team → Push...". Give the HTTPS based repository address as URL (you can use the HTTPS tab on the PikaCode page to see what that URL is), along with the group username and password.

Setup for Group Members

- Download the wicket.zip, slf4j.zip, and jetty.zip files mentioned above.
- In Eclipse, select “File → Import... → Mercurial → Clone Existing Mercurial Repository”. Give the HTTPS based repository address as URL, along with the group username and password. This will create a local copy of the repository.
- Extract the wicket.zip, slf4j.zip, and jetty.zip files under the top folder of your project (“dbms”).
- In the Eclipse package explorer, right click on the project name and select “Refresh”.

Now the project should be ready. In the Eclipse package explorer, right click on the file “Start.java” under the “tr.edu.itu.cs.dbms” package and select “Run as... → Java Application”. If all went correctly, you should see the following message in the Eclipse console:

```
[main] INFO org.eclipse.jetty.server.AbstractConnector - Started  
SocketConnector@0.0.0.0:8081
```

In your browser, visit the address <http://localhost:8081/> and you should see the Wicket quickstart page.

Whenever you make changes to the project, you first have to commit them to your local repository. Right click on the project name and select “Team → Commit...”. Always write a short message describing the changes that you are committing. To send your changes to the main repository (on Pikacode), you have to push them (using “Team → Push...”). To get the changes pushed by other group members to your local repository, you have to pull them (using “Team → Pull...”).

Do not push your changes if they cause the project to fail, that is, if the project does not compile or run anymore. Pushing broken code will disrupt the work of your groupmates. Also, it is good practice to do a pull before a push so that you will receive the changes since your last pull and spot any conflicts that might arise.

Tips:

- After doing a commit, do not postpone the push too long. This will increase the risk of getting a conflict with the work of group members. If your code is working, directly do a push after the commit.
- For every work session, plan ahead what you will do and try to stick to that plan. When committing, if you can not find a meaningful commit message for the changes, it is very likely that you have lost your focus.