

GALORE: ENV SETUP GUIDE

1. SVN (ECLIPSE) SETUP

This is some what complex procedure to setup the eclipse with svn. If you do not want eclipse to be configured with svn(skip eclipse+subversion part), you can download the tortoise svn client for windows and get the sources to the local windows box. Then you should know that the svn checkout/checkin diff happens with tortoise client, but not inside eclipse. I find it more convenient to enable eclipse to do those things. Anyway, here is the setup that I compiled some of it from the internet and some my own experiences while configuring it for myself.

0. download eclipse plugin for svn here:

http://subclipse.tigris.org/update_1.6.x

1. Set up ssh keys.

- i. download putty; but we should not use the puttygen that comes with the putty installation.
- ii. <http://tartarus.org/~simon/putty-snapshots/x86/puttygen.exe>; use the puttygen from this location as opposed to the one you get from putty installation
- iii. Try the following on galore.clockreplay.com server(you should have your account created already on the server):

```
ssh-keygen -t dsa
accept the file names it wants to use
do not enter any pass phrase, this makes it easy.
```

```
ssh to your server using good old user name and password
do check permissions on your ~/.ssh folder and make sure to
chmod 700 .ssh
```

if they are wrong

```
do check permissions on your ~/.ssh/authorized_keys file and make sure to
chmod 600 authorized_keys (if you dont have authorized_keys file, create a
new one with touch authorized_keys
```

```
add the pub key to the authorized_keys file with something like
```

```
cat id_dsa.pub >> .ssh/authorized_keys
```

```
copy the content of the id_dsa file as is to your local windows machine (use
winscp or sftp or some such tool or copy/paste using mouse) and name the
file "id_dsa.ppk".
```

NOW open puttygen.exe(not the one that comes with the installation)
under actions select "load" and load the id_dsa.ppk file that you just copied
from linux box. In the parameters section below, select SSH 2-DSA option and

click "Save private key" button.

Puttygen will now convert the key to something that putty will understand save that file to something like

privatekey.ppk

NOW change your putty settings under "connection > SSH > auth" to use privatekey.ppk; now enter the server name galore.clockreplay.com and save the putty session information by giving a name say GALORE. remember that the same session name should be given while checking out the files from eclipse.

NOW try and connect using putty(select session GALORE); should not ask for the password, if you do everything right.

2. Download and installed the excellent [TortoiseSVN](http://tortoisesvn.tigris.org/) (<http://tortoisesvn.tigris.org/>) client for Windows.
3. Set the following environment variable (by right-clicking on My Computer, Properties, Advanced, Environment Variables, System variables, New):-
4. Variable name: SVN_SSH
5. Variable value: C:\\Program Files\\TortoiseSVN\\bin\\TortoisePlink.exe
6. (The "\\" is very important, otherwise it won't work. Equally, you cannot use the plink.exe that comes with putty as that fires up a command shell window which is really annoying. The TortoisePlink.exe is a windows implementation of plink that doesn't bring up any UI)
7. Configure the Subclipse plugin to use JavaHL (JNI)
8. Restart Eclipse
9. while checking out from svn thru eclipse, the trick is that you should give the location of svn root like below

svn+ssh://haranadh@galore/svnroot

please note that @galore, here is the name of the putty session that you saved previously with privatekey configured.

Now, you can open a project in eclipse of type svn checkout. then directly checkout the files into a new/existing project of eclipse.

2. Jira/confluence/fisheye links

/opt/software/jira/install/atlassian-jira-4.3.2-standalone/bin\$

jira : <http://galore.clockreplay.com:8088/secure/Dashboard.jspa>

crowd: <http://galore.clockreplay.com:8095/crowd/console/login.action>

confluence: <http://galore.clockreplay.com:8090>

fisheye/crucible: <http://galore.clockreplay.com:8060>

Enable Asserts in Eclipse

debug/configurations/vm arguments add `-enableassertions`

Galore eclipse env setup

- install checkstyle and pmd and enable them by default to all the projects
 - <http://eclipse-cs.sourceforge.net/downloads.html>
 - <http://pmd.sourceforge.net/eclipse/>
- create ENV variable in eclipse(you can go to the debug configurations to do this) CRP_ROOT and set it to the source tree root directory
- create a system property in eclipse(debug configurations java vm args) for log.home, pointing to CRP_ROOT directory where you can have logs.

Here are my settings for debug config vm in eclipse

-Dlog.home="c:\eclipse\gm" -enableassertions

Galore Before Checkin Dos and Donts

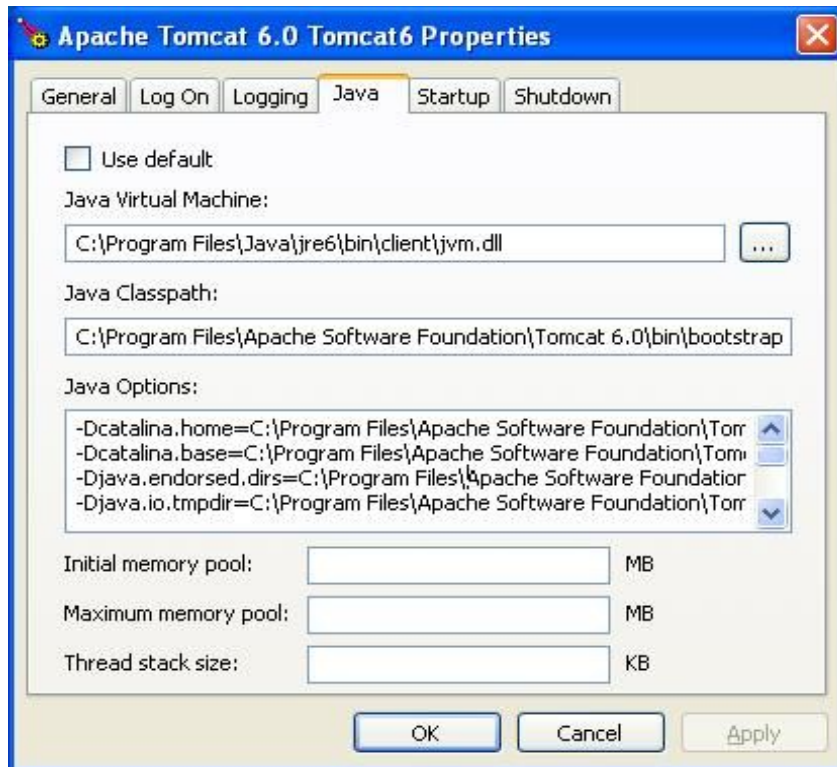
- make sure you have a test file that tests your changes.
- tortoise svn update first, make sure you have latest changes
- **look for any conflicts and resolve them ; this is very important.**
- create patch file from tortoise svn
- go to fisheye (galore.clockreplay.com:8060) , create a new request, edit details add reviewers to the review and upload the patch file.
- get it reviewed by at least one person before proceeding for checkin.

Command Line Running Options

```
C:\eclipse\gm\lib\ext>java -Dlog.home="c:\eclipse\gm" -Xbootclasspath/a:c:\eclipse\gm\lib\ext\log4j-1.2.8.jar;c:\eclipse\gm\lib\ext\protobuf.jar;c:\eclipse\gm\lib\json.jar -jar crp.jar ProcessMain
```

```
C:\eclipse\gm\lib\ext>java -Dlog.home="c:\eclipse\gm" -Xbootclasspath/a:c:\eclipse\gm\lib\ext\log4j-1.2.8.jar;c:\eclipse\gm\lib\ext\protobuf.jar;c:\eclipse\gm\lib\json.jar -jar crp.jar ProcessMain
```

How can you set java vm arguments for tomcat in windows.



Add jnetpcap library to eclipse

for jnetpcap to work, first download and install the winpcap library for windows 32 or 64.
download jnetpcap.
add the jnetpcap jar to eclipse setup as below.

1) Select the project you want to add jnetpcap library to and open its properties:

Project->Properties->Java Build Path
or right click on the project in "Package Explorer" and select
Build Path->Configure Build Path

2) Click the "Add External JARs..." button (or "Add JARs..." if you installed jnetpcap within the Eclipse workspace)

3) Go to your installation directory (c:\libs\jnetpcap-1.2 for example) and select the jnetpcap.jar file and click "Open" button to add the jar file to the project's build path.

4) A "jnetpcap.jar" file should show up in the "build path" tree. Expand it.

5) Select "Native library location" and click "Edit..." button.

6) Select the directory where the native file resides (c:\libs\netpcap-1.2 for example).

Thats it!! Now you are good to go.