Freenet: Setup Eclipse and Debug Environment

University of Hawaii at Manoa

Project Members
Dr. Yingyei Dong
Todd Baumeister
Dwayne Yuen

Author
Todd Baumeister

Date
2/21/2011

1. Purpose

The purpose of this document is to guide a developer through the process of setting up Freenet on a machine, so it can be debugged with Eclipse. The process of retrieving the source control, setting up Eclipse, and getting Freenet to run locally will be covered.

2. Paths

This section lists out the paths that are used for various tasks in this document. Each path is assigned a short name to make reading the document easier for the user. Also additional information about each path will be list here.

BaseDir := /home/[username]/   (changeable)
This is the directory where you choose to place the git source controlled folder. All other paths will have *baseDir* as there prefix in this document. The location (value) of this directory is at the users discretion, and for simplicity we will assume that the home directory(~) was chosen.

ProjectDir := [*baseDir*]/Freenet-Analysis
This is the root folder of the Freenet analysis project. [*ProjectDir*] contains the eclipse project with the Freenet source code, and other files used to support the analysis of the Freenet protocol.

DocumentsDir := [*baseDir*]/Freenet-Analysis/Documents
This is the folder where all of the documentation used in the analysis project is stored. [*DocumentsDir*] contains documents such as this one.

EclipseWorkDir := [*baseDir*]/Freenet-Analysis/eclipse_workspace
This is the directory that Eclipse uses as base directory for its software projects. In [*eclipseWorkDir*], you will find a FreeNet-Source folder that contains all of the source code for the Freenet project.

FreeNetAppDir := [*baseDir*]/Freenet/
This is the directory where Freenet is installed by default. This directory can be changed.

3. Source Code

- Install Git
- Retrieve source code
- Update source code
- Commit source code
- Install qgit (optional)

The first thing that needs to be done is install Git. Open the command shell and type the command "sudo apt-get install git". Git is the source control used to manage the project's files.

Next, you will need to retrieve the source code from github.com. This step only needs to be performed once to create a local copy of the repository. You will use git to copy the public repository to your [*baseDir*]. In command shell, navigate to the [*baseDir*] folder, and type one of the following two commands that are in double quotes.

Read-only access: "git clone git://github.com/tbaumeist/FreeNet-Analysis.git"
or
Read-write access: "git clone git@github.com:tbaumeist/FreeNet-Analysis.git"
(requires additional configuration at github.com)

This will create a copy of the repository on your local machine. After the operation has completed there should be a folder [*baseDir*]/Freenet-Analysis that has files in it. Note that [*baseDir*]/Freenet-Analysis is the same as [*projectDir*].

The update step will retrieve the latest changes from github.com, and should be performed periodically. This step should be performed periodically to update your local repository with the public repository. In the command shell navigate to the [*projectDir*] and type "git pull".

After a source code change has been locally committed, then it needs to be pushed to github.com. In command shell navigate to the [*projectDir*] and type "git push". You need to consult the repository owner before commits can be pushed. The repository will need to be configured to allow you write access.

A recommended GUI interface that can be used for code check-ins is qgit. It can usually be installed by going to the Ubuntu Software Center and searching for it. This GUI is a nice tool for visualizing code changes, and view revision histories.

4. Eclipse

- Install Eclipse
- Create Java project

It is assumed that the previous steps from the source code section have already been completed. Install Eclipse from the Ubuntu Software Center. Eclipse version 3.5.2 was used while this document was created.

After installing Eclipse, open Eclipse. Eclipse should ask you to select a workspace. Select [*eclipseWorkDir*] as your working directory. If Eclipse does not ask for your workspace, and it is not on the correct workspace. Then got to File->Switch Workspace. Now go to File->New->Project, and select Java Project from the list. On the dialog box that pops up

choose "Create project from existing source", and navigate to [*eclipseWorkDir*]/FreeNet-Source/. Click Next->Finish. Most of the project setting are stored in the git repository, and you should be able to build and debug at this point.

Note: After you build your project successfully the first time, you need to select the project folder in Eclipse and right click->refresh. This will cause Eclipse to recognize the new files and folders generated by the build. This must be done before debugging can be done the first time.

5. Building

At this point there are two options for building the Eclipse project. 1) use eclipse GUI. 2) use ant. If you are building from the Eclipse GUI try to do a project->Clean when you build. This will resolve most build related issues. If using ant, open the bash shell, navigate to [*eclipseWorkDir*]/FreeNet-Source/, and type "ant clean" then "ant". This is done for the same reasons as a project->clean in the GUI.

6. Debug

After you have successfully built Freenet, you will want to debug it.   This step may not be required as it will sometime be imported in with the eclipse project files. To set up debugging, go to Project->Properties, while the FreeNet-Analysis project is selected in Eclipse. Then go to Run/debug settings, and click New->Java Application. The main class is "freenet.node.Node". On the Classpath tab Add JARs, and select the freenet.jar file in [*eclipseWorkDir*]/FreeNet-Source/dist/. Note, the project must have built successfully for the jar to be there. If you cannot find the jar file, and it exists follow the italic instructions from section 4 of this document.

Now you must choose the correct ini settings to ensure your Freenet runs properly. There are already two freenet.ini file checked into the source control in [*eclipseWorkDir*]/FreeNet-Source/. One is freenet.ini.seednode and the other is freenet.ini.noneseednode. To fix an issue with the connection phase in the testbed, code was added to prevent seed nodes from sending out connection requests to other seed nodes. The flag that turns this on and off is in the ini file. If node.opennet.acceptSeedConnections=false, then the node will attempt to connect to the seed nodes. If node.opennet.acceptSeedConnections=true, then the node is a seed node, and it will not try to connect to other seed nodes. The other basic ini file setting are in the files checked in source control.

7. Freenet

- Install Freenet from http://freenetproject.org/

- Turn off Freenet autostart
- Set up development folder

Install Freenet from http://freenetproject.org/. There are several installers, and I recommend the web-start installer. Note, at this point you will have two different Freenet related directories on your machine. The Freenet install directory is [*freeNetAppDir*] by default.

Freenet will be set up as an autostart service by default. This will cause problems if you try to run several freenet processes at once on one machine. To turn off auto start go to [*freeNetAppDir*]/bin/ and run "remove_cronjob.sh". If Freenet is already running it can be stopped by going to [*freeNetAppDir*] and running "./run.sh stop" in the command shell.

If you use Eclipse to launch a debug instance of Freenet, then it will be using the Freenet.ini file in [*eclipseWorkDir*]/FreeNet-Source/. This ini file still has several properties that still point to the [*freeNetAppDir*], so it will save any data in that folder instead of [*eclipseWorkDir*] /FreeNet-Source/. You may have to slightly alter you [*eclipseWorkDir*]/FreeNet-Source/ freenet.ini so that it has your proper folder paths.

8. Conclusion

If all of the configuration and data files remain unchanged from the initial freenet install. Then when you debug freenet with the above steps, it will attempt to connect to the publicly available freenet network. Further documents will be released that will guide a developer through configuring multiple Freenet nodes, so that they can run in an isolated environment.