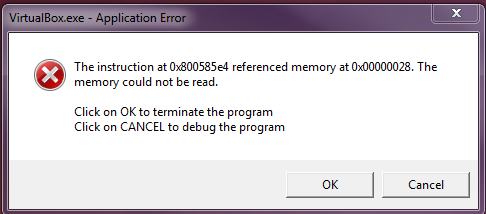
# SOFTWARE INSTALLED, AND ISSUES DISCOVERED & RESOLVED

STS Eclipse workspace folders  
C:\jbdissertation\scm\mscdissertation\workspace  
C:\Users\johnwarde\Documents\workspace-sts-3.6.3.RELEASE

#### Maven commands/shortcuts

* To build entire project (from same folder as pom.xml)  
  mvn package
* Same as above but without the automated unit test  
  mvn package -Dmaven.test.skip=true

## VirtualBox: UbuntuDesktop:

* If attempting to start a VM in a saved state and then the **guest is unresponsive** then try View->Auto-resize Guest Display (Right-Ctrl + G), if not then try Devices‑>Insert Guest Additions CD image and then try View‑>Auto-resize Guest Display (Right-Ctrl + G) again.
* After a Ubuntu Software Update, the screen may not be sized correctly, do the following:
  + Install Additions CD (from VB menus)
  + Restart Ubuntu VM
  + Flick through full screen mode and other view options and should get it back
  + If that doesn’t down then you need to “power down” (“send shutdown signal will not work”). **Do not choose “restore to current snapshot”** as it will reset the machine back to the day you took the snapshot”
* Crash after logging into a Ubuntu VM and sharp exit:
  + Also get the following:  
    
  + Go into the Settings of the VM and then go to storage
  + Browse to the Guest Addition CD icon/iso
  + Right-click and chose Remove Attachment
  + Save the settings and start the VM again, should be okay now
* To share a folder between Linux and Windows:
  + First ensure that files will not be in contention when accessing them, i.e. if you have a PDF open in Windows and then you try to open the same PDF in Linux you will have problems and be especially careful when writing to a file – if you have to do this then it should via git!
  + In VirtualBox menus, choose Devices->Shared Folders Settings …
  + Enter **name**, folder and access level
  + mkdir -p ~/shared/**books** (i.e. ensure the folder is created before next command)
  + sudo mount -t vboxsf **books** ~/shared/**books**/
* Use the following to locate package names for using with Chef  
  apt-cache search git | grep git  
  The grep part is only used to highlight the searched word

Windows: When a Java application cannot load (i.e. cannot create a JVM or find a JRE/JDK then it usually means that the application is 32 bit or 64 bit and it cannot find the required JRE/JDK in 32 bit or 64 bit)

## Amazon Web Services Info

* SSH client -> use PuTTY  
  <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

## Useful Linux Commands

* Confirm open and listening on Linux  
  sudo netstat -a | egrep 'Proto|LISTEN'

## GIT COMMANDS

* GitHub for Windows is very useful and easy to setup and includes a PowerShell command line.
* Git on Linux: you need to set up SSH access – good instructions here : <https://help.github.com/articles/generating-ssh-keys/>
* This explains the foundation of GIT – graph theory  
  <http://think-like-a-git.net/>  
  Still to read
* Branching  
  <http://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

### GIT – initial config

(maybe put these into devenv Chef recipe eventually)

git config --global user.name "John Warde"  
git config --global user.email [john.warde@gmail.com](mailto:john.warde@gmail.com)  
mkdir ~/.ssh  
cd ~/.ssh  
ssh-keygen -t rsa -C [john.warde@gmail.com](mailto:john.warde@gmail.com)  
ls -l /c/Users/johnwarde/.ssh/id\_rsa  
ssh -T [git@github.com](mailto:git@github.com)  
cd ~/scm  
git clone <https://github.com/johnwarde/library.git> (would clone into a folder names library)

git clone https://github.com/johnwarde/mscdissertation.git mscdbranching (clones remote repository into a specific folder name)

xclip -sel clip < ~/.ssh/id\_rsa.pub (copies contents of file to clip board for pasting into a new SSH key on gi hub, see settings->SSH Keys)  
ssh -T [git@github.com](mailto:git@github.com) (again)

### GIT – initialize new repository

(can always init a repository on github and clone down)

cd C:/Users/johnwarde/Downloads/johnwarde/library  
git init  
git status  
git status –s  
git add . (add everything in current directory)  
git commit -m 'First commit of project'  
git remote add origin git@github.com:johnwarde/library.git (links to repository at github.com)  
git push -u origin master

### GIT – normal workflow

git status  
git add src/library/Book.java  
git add src/library/Library.java  
git add . (you need to do a git add on already tracked/modified files too to put them into staging for commit)  
git commit -m "Added Library.java, modified others to work with it."  
git push origin master (push changes to remote repository)  
git status  
git remote show origin (indicates whether there are updates available from the remote repository)  
git pull origin (pull updates from remote repository and merges with local repository)

**NOTE: careful when renaming files** – you need to rename and commit immediately (without changing the contents of the file – git uses file content to detect a rename!!!)

* <http://blog.bauermann.eng.br/2008/01/03/take-care-when-renaming-files-in-git/>

### GIT – other useful commands

git log

git –help

git rm --cached ../docs/~$nNotes.docx (supposed to remove tracked file but didn’t work)

### GIT – tagging

git tag (lists locally available tags)

git tag -a v1.4 -m 'my version 1.4' (Creates an annotated tag) in local repository

git push origin [tagname] (pushes the tag to the server so others can use)

git push origin --tags (Pushes all defined tags to the remote)

git pull origin --tags (Pulls all remote defined tags into local repository)

### GIT – branching related commands

git branch -vv (lists your branches and how they are connected to remotes (remote repository))

### GIT - Create a new branch

* NOTE: if you are creating(/copying to) a new branch then you need to ensure that the .project and .classpath files in Eclipse projects exist in the git repository otherwise these projects will not be accessible in a new workspace. You’ll still need to import an existing project, see last point below.
* Ensure that your working folder is clean, i.e. no edited files
* git branch <NewBranchName>
* git push origin <NewBranchName>  
  (This pushes the newly created repository to the remote origin, the only reason you would not do this is if you wanted this branch to only live in locally, you would need to issue this command if you haddir another working folder that you wanted to use this branch)
* If you switch to another working folder or another user wants to pull down the new branch then you/they need to …  
  git fetch origin <NewBranchName>  
  (This fetches the new branch to your local repository, you should not use a ‘git pull’ command as that command does a git fetch and then a got merge, you don’t want to merge the new branch into the branch you are currently on – not sure if a git pull will go ahead and just do a git fetch and pull in the local <NewBranchName> of your local repository.
* git checkout <NewBranchName>
* git pull origin <NewBranchName>  
  (nothing should happen because you have already pulled changes to the <NewBranchName> branch with the ‘git fetch origin <NewBranchName>’ command but just to this to be sure your local <NewBranchName> branch is up-to-date )
* NOTE: if you are pulling a new branch from a remote repository and you have an Eclipse project then you will need to re-import the project back into the workspace by …
  + File->Import …
  + Locate ‘Import Existing Projects into Workspace …’ in the General folder and click OK
  + Choose the workspace folder and the wizard should be able to locate the projects if the .project files are located within the project folders, see first point above.
  + Generally you can ignore the “Some projects cannot be imported because they already exist in the workspace” for Eclipse generated project i.e. ‘Servers’

Done!

**18/10/2014 16:13**

* Upgraded VirtualBox to 4.3.18 which resolved the “make sure the kernel module has been loaded successfully” (also bad dll message on opening of VirtualBox). All VMs are now working
* Issues with Java which I had fixed before, still need to fix, I should refrain from upgrading Java until the end of the project.
* In UbuntuDesktopX32 VM:
  + Maven has been installed but never used it
  + Disk space is low, need to free up if upgrading from Ubuntu 12 -> 14
  + Accidently deleted the hard drive image of this VM, have now deleted this
* Created UbuntuServerX64 (may have only called it UbuntuX64 in places/set-up i.e. hostname)
* Created UbuntuDesktopX64 using C:\Users\johnwarde\Downloads\ubuntu-14.04.1-desktop-amd64.iso

**27/10/2014 21:01**

* After learning some Puppet from tutorials at puppetlabs.com, decided to switch to Chef after reading a number of articles stating that Chef is more suited to multi-cloud environments because it more customisable and better integration with external tools which may be needed for this project.
  + <https://www.computenext.com/blog/opscode-chef-vs-puppet-labs-for-multi-cloud-orchestration/>
  + <http://devopsanywhere.blogspot.ie/2011/10/puppet-vs-chef-fight.html>
* Installed Ubuntu version of Chef-DK on UbuntuDesktopX64 from <http://downloads.getchef.com/chef-dk> and let the Ubuntu Software Center open the .deb file

**02/11/2014**

* Attempt to get stuff going on the macbook …
* Was able to reset my account (name) using steps in the “Resetting a password without an additional account” from here:
* Then needed to reset/delete the default keychain:  
  <http://support.apple.com/en-us/HT1631>

**01/11/2014**

* On Unbuntu
* Installed Oracle JDK using the following, based on the “Install Oracle Java 7 in Ubuntu or Linux Mint via PPA” section of <http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>  
  sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java7-installer

java -version

javac -version

sudo apt-get install oracle-java7-set-default

**02/11/2014**

* Continuing on with Chef (already installed on TODO) ...
* Followed tutorial starting at https://learn.getchef.com/ubuntu
* Then followed this which required ...
* Signing up for a free trial account at opscode.com
* Downloading a Starter Kit, a zip file containing certificates, links to account just set up etc. : <https://learn.getchef.com/ubuntu/get-ready-to-add-another-server/>

**02/11/2014**

* On MacBook ...
* Downloaded the .dmg file for ChefDK however I am getting an “Illegal Instruction 4” or “Segmention fault: 11” when I try to run any chef commands
* Decided to uninstall ChefDK and upgrade Mac OSX

**06/11/2014**

* Mackbook …
* Successfully upgraded Mac OSX to Yosmite
* Successfully installed ChefDK:
  + Downloaded the .DMG file from <http://downloads.getchef.com/chef-dk/mac/#/>
  + Opened the .DMG file on the Mac and it contained a .pkg file, double clicked on this and it installed successfully
  + Typed 'chef verify' in a terminal session and all was verified with no illegal instruction/segmentation fault errors
  + More info here for un-installing etc.: https://docs.getchef.com/install\_dk.html
  + Saving this line although this may only be for installing chef client tools and not the ChefDK  
    curl -L http://www.getchef.com/chef/install.sh | sudo
* Need start creating a Chef recipe for the applications I install on the Mac, start with git then Java. Also need to start thing about but these into the git repository

**08/11/2014**

* Learned that Mac OSX is a derivative of BSD Unix and not Linux and have decided against using Mac OS as there may be incompatibilities with Linux packages I want to install and in terms of Chef, what would work on the Macbook may not Linux cloud instances.
* Sticking to Linux/Windows environment, hope the VM holds out.

**09/11/2014**

* Learned that Mac OSX is a derivative of BSD Unix and not Linux and have decided against

**10/11/2014**

* Started adding packages to Chef recipe file UbuntuDesktopX64.rb, got Eclipse and git set up

**13/11/2014**

* Was able to sync down old Java library projects , load in Eclipse, debug and run from command line to confirm that Java is working on the Ubuntu guest VM
* Copy & paste steps to run Java project and git commands summary from the library repository (OwnNotes.docx)

**14/11/2014**

* Synced down mscdissertation repository from github and set it up so that I can make changes from Ubuntu, see “GIT COMMANDS” section above.

**27/12/2014**

* Started learning the Spring Framework while …
* Trying to get Spring +Maven working on Ubuntu

**27/12/2014**

* Attempted to get Maven & Eclipse working on Ubuntu however after much trial and error (Eclipse was also not able to update successfully even after installing a new instance) abandoned it in favour of getting them to work on Windows.

**01/01/2015**

* Installed Spring Tools Suite (STS) from <http://spring.io/tools/sts/all> on Windows using 64 bit zip, just unzipped the files into C:\apps\sts-bundle. However, it did not work straight away, the error message suggesting that there was a mismatch between 32 bit versus 64 bit. So ..
* Installed JDK using <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>) , choose jdk-8u25-windows-x64.exe

**07/01/2015**

* Installed Windows version of Maven
  + Downloaded latest version of file ending in “bin.zip”
  + Extracted zip to C:\apps to get a versioned folder
  + Followed the steps described at <http://maven.apache.org/download.cgi#Windows> along with …
    - Couldn’t add %M2% to Path had to explicitly call out the path
    - Moved M2 from user variable to system variable
    - Had to set JAVA\_HOME to the JDK home of the currently installed JRE  
      C:\Program Files\Java\jdk1.8.0\_25`
* Cleaned up Path environment variables: had empty entries, duplicates, folders that no longer existed and finally need to delete some java\*.exe in %SYSTEMROOT%\System32 which were version 1.6.

**08/01/2015**

* Installed Erlang 64bit prerequisite for RabbitMQ, accepted defaults except for destination folder which defaulted to erl6.3 for Erlang OTP 17 (2 numbers are confusing) so chose C:\Program Files\erlang
* Installed what looks like the 32bit of RabbitMQ (did not appear to have a 64 bit version). Installs just as a service don’t see a front end maybe a web interface. More info here: <http://www.rabbitmq.com/install-windows-manual.html> (used the info but didn’t follow it as the exe did all this)
* Was able to put together a sample project containing a web server and AMQP (RabbitMQ) working together and simultaneously!
  + <https://github.com/spring-projects/spring-boot/tree/master/spring-boot-samples/spring-boot-sample-amqp/src/main/java/sample/amqp>
  + <https://github.com/spring-projects/spring-boot/tree/master/spring-boot-samples#readme>
* To import a spring boot maven project (or any maven project) into Spring Tool Suite (STS)
  + File->Import …
  + In Maven Folder choose ‘Existing Maven Projects’
  + Choose the root directory where the pom.xml exists
  + Click ‘Finish’ button
  + Project folder will have the M symbol
  + As Eclipse/STS attempts to build the maven project it will detect the Spring Framework and will eventually attain the S symbol for Spring

**09/01/2015**

* Successfully extracted ‘spring-boot-sample-web-secure-jdbc’ from Spring Boot samples
* Got a clean Maven build and run at <http://localhost:8080>
* Successfully imported maven project into STS Eclipse

17/01/2015

* Got ThemeLeaf templating working with security, see CL
* When I get a blank screen render out usually that means that there was an error in the templating engine ThymeLeaf and should look at the Console in STS for error messages

18/01/2015

* Installed the H2 database, from <http://www.h2database.com/>, which is compatible with the Hibernate Persistence framework.
* WebApp does not log out properly, trying things
  + Added the following to the end of “protected void configure(HttpSecurity http) throws Exception { “ (does not work!!)  
    .and().logout().logoutUrl("/logout").invalidateHttpSession(true).logoutSuccessUrl("/")

20/01/2015

* For the image folder in the webapp, trying to keep unform URI so that when the Java jar runs on a Linux VM it will work straight away – may need to revisit current code then … this might be the answer …  
  <https://weblogs.java.net/blog/kohsuke/archive/2007/04/how_to_convert.html>

27/01/2015

* Was able to get the reference app working with RabbitMQ reliably by running it as an application rather than a web service.
  + Stopped RabbitMQ in Services
  + Started RabbitMQ as an application by
    - Selecting Start->All Programs>RabbitMQ Server->”RabbitMQ Command Prompt (sbin dir)” – right-clicking this and choosing “Run as Administrator”, also start a 2nd one of these for monitoring commands, see below.
    - In one of these command prompts execute “rabbitmq-server.bat”
    - In the 2nd command prompt, you can use such commands as  
      rabbitmqctl list\_queues foo (list the no.of messages in queue)  
      rabbitmqctl list\_users  
      rabbitmqctl list\_user\_permissions <username>  
      rabbitmqctl list\_queues  
      rabbitmqctl list\_exchanges  
      rabbitmqctl list\_bindings  
      rabbitmqctl list\_connections  
      rabbitmqctl list\_consumers  
      rabbitmqctl status  
      rabbitmqctl (for full help)
* Need to try it back in service mode too.

30/01/2015

* STS/Eclipse/EGit – if git view is not showing all projects (i.e. not showing a recently added project to workspace) then right-click on that project and select Team->Synchronise (with GIT)

05/02/2015

* Generate Javadoc in Eclipse/STS
  + Choose Project->Generate Javadoc …
  + Choose the Javadoc.exe residing in the JDK under %JAVA\_HOME%
  + Click the defaults after that

08/02/2015

* Added jClouds inter-cloud framework to the WebApp and Processor components using the information here:
  + Added the following lines inside the dependencies tag:  
    <dependency>  
     <groupId>org.apache.jclouds</groupId>  
     <artifactId>jclouds-all</artifactId> <version>${jclouds.version}</version>  
    </dependency>
  + And changed ${jclouds.version} to the latest version of jClouds which is 1.8.1
  + Ran the following maven command from the command line:  
    mvn dependency:copy-dependencies
* JMX client jconsole.exe is available in the JDK at %JAVA\_HOME%\bin\jconsole.exe (C:\Program Files\Java\jdk1.8.0\_25\bin\jconsole)
* Trying out Live graph (<http://www.live-graph.org/>) and LogMX (<http://www.logmx.com/>) for telemetry options
* Used information here to get selenium working from Java exported from Selenium IDE  
  <http://www.seleniumhq.org/download/maven.jsp>

15/02/2015

* Test on AWS
  + Install & configure PuTTY client on Windows:  
    <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>
  + Install AWS CLI on Windows:  
    <http://docs.aws.amazon.com/cli/latest/userguide/installing.html>
  + Install AWS CLI on Ubuntu  
    sudo apt-get install awscli
  + Configure AWS CLI on Windows and Ubuntu  
    aws configure  
    (You will be asked for AWSAccessKeyId and AWSSecretKey, use us-east-1e for region and json for format)
  + Frequently Used Commands
    - aws s3 ls s3://  
      (list all my buckets)
    - aws s3 ls s3://johnwarde.net/  
      (list files in specified bucket)
    - aws s3 cp s3://johnwarde.net/WebAppConfig.java .  
      (copy file in s3 bucket to current local folder on EC2 instance)
  + Install RabbitMQ server on EC2 / Ubuntu:  
    sudo apt-get install rabbitmq-server
  + Install JRE on Ubuntu (Java Runtime):  
    sudo apt-get install default-jre (didn’t work, got fatal error, could start JVM)
  + To remove installed java:  
    sudo apt-get --purge remove default-jre  
    sudo apt-get autoremove
  + Used steps near start of this doc to get Java installed (where it adds a repository)
  + List all processes:  
    ps aux
  + Stop a process:  
    kill -SIGTERM <PID>   
     OR  
    kill –SIGKILL <PID>

**20/02/2015**

* To override the application.properties in a jar:
  + Add the following line to the src/main/resource/META-INF/application-context.xml  
    <context:property-placeholder location="file:config.properties, classpath:config.properties" ignore-resource-not-found="true"/>
  + Create a separate application.properties with the desired settings …
  + When it is in the same folder as the jar file when ‘java -jar myjar.jar’ then this file will be used to populate the configuration instead of the one embedded in the jar file
  + From: <http://stackoverflow.com/questions/1208848/is-it-bad-practice-to-include-properties-configuaration-files-within-jars#answer-12584644>
  + Removed AWS EC2 version from source control so keeping a copy here ..  
    **webapp.hostname = ec2-54-204-163-94.compute-1.amazonaws.com  
    webapp.imageFilesRoot = file:/home/ubuntu/webappresources  
    logging.file = /home/ubuntu/webapp.log**
* Got Logback logging working …
  + Added the following to pom.xml  
    <dependency>  
     <groupId>ch.qos.logback</groupId>  
     <artifactId>logback-classic</artifactId>  
     <version>1.0.13</version>  
    </dependency>  
    … and to main java file …  
    **import org.slf4j.Logger;  
    import org.slf4j.LoggerFactory;**  
    … and replace log object creation with …  
    **static final Logger log = LoggerFactory.getLogger(WebApp.class);**
  + Integrated with the application.properties file so it can be configured external to the jar
  + Creates log entries to STDOUT/Console and a log file
  + Each column/field is separated by tab for easier processing by sed or LiveGraph

21/02/2015

* Using Linux’s sed (stream editor) to prepare data file
  + This …  
    # Using Logback formatted logs with tabs between fields  
    # Delete those lines that do not contain 'effectfetch'  
    /effectfetch/!d  
    # Delete lines containing '[main]'  
    /\[main\]/d
  + <http://stackoverflow.com/questions/9541867/sed-delete-lines-not-containing-specific-string>
  + sed manual:  
    <http://www.grymoire.com/Unix/Sed.html>
  + Guide  
    <https://www.digitalocean.com/community/tutorials/the-basics-of-using-the-sed-stream-editor-to-manipulate-text-in-linux>
* Created LiveGraphHeader.txt file containing the header format for Live Graph …  
  ## ##  
  Timestamp ProcessName Level Class Action Result Host User CorrelationId Duration  
  Those are tabs between double cardinal symbols and between each field name
* One time do this …  
  cp LiveGraphHeader.txt data.txt
* Then repeatedly do this …

22/02/2015

* RabbitMQ management plugin:
  + Installed by running the following the RabbitMQ administrator command window  
    rabbitmq-plugins enable rabbitmq\_management
  + WebUI is then here:  
    <http://localhost:15672/>
  + To enable WebUI on an EC2 instance need to do one extra step – restart the server  
    sudo rabbitmqctl stop  
    sudo invoke-rc.d rabbitmq-server start
  + More info here:  
    <https://www.rabbitmq.com/management.html>
  + To clear out a queue (not tried yet):  
    rabbitmqadmin purge queue name=name\_of\_the\_queue\_to\_be\_purged  
    rabbitmqadmin purge queue name=processor\_rpc\_queue

24/02/2015

* AWS Metadata from launched instance
  + <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>
  + <http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-DescribeInstances.html> (examples here)
  + Still need to find a way to easily retrieve the tags I assign to an instance or maybe the “user data”
  + Parsing JSON output from AWS CLI commands:  
    <http://docs.aws.amazon.com/cli/latest/userguide/controlling-output.html>
* Getting Started with Auto Scaling - Auto Scaling   
  <http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/GettingStartedTutorial.html>
* Configure Custom Domain Name for Your Load Balancer - Elastic Load Balancing   
  <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/using-domain-names-with-elb.html>
* Pre-experiment test run where we were putting some load on the application as a whole and then too much – for testing scalability  
  2 x Processor + 1 x WebApp  
  maxopenreq 24 → 48  
  batchsize 8 → 16  
  timeout 2500 → 2500
* Confirmed that I can send logging from multiple WebApps to a single file and that I can use either of the following to filter completed processing request to a file readable by LiveGraph (just need to add the formatted header lines, see above in this document)  
  cat webapp.log | sed –f sed

25/02/2015

* Needed to create a separate workspace for working on the different multi-cloud frameworks while still editing the Dissertation document in the main branch – do not want to be editing the binary dissertation document in any other branch because a binary merge would not be good (but this probably does not happen) plus don’t want to be switch back and forth between branches opening and closing Word to add to document.
* cd C:\jbdissertation\scm
* git clone https://github.com/johnwarde/mscdissertation.git branching  
  (Creates folder C:\jbdissertation\scm\branching with repository from URL)
* cd C:\jbdissertation\scm\branching
* git branch jclouds  
  (Create branch)
* git checkout jclouds  
  (Switch to branch)
* Started STS/Eclipse and set (new) workspace location to C:\jbdissertation\scm\branching\workspace however the workspace was blank except for the standard ‘Servers’ folder.
* … the .project and .classpath folders seems to be missing in new local repository
* cd C:\jbdissertation\scm\mscdissertation
* Temporarily edited .gitignore and removed .project and .classpath entries
* git add workspace\processor\.project
* git add workspace\webapp\.project
* git add workspace\processor\.classpath
* git add workspace\webapp\.classpath
* git commit -m 'Adding .project & .classpath files so that the Eclipse projects will open automatically in other working folders'
* git push origin master  
  (Push the change to the remote)
* git status
* cd C:\jbdissertation\scm\branching
* git checkout master
* git pull origin master   
  (Pulls the latest changes from the remote that we made in other working folder)
* git checkout jclouds
* git merge master  
  (merge the local master branch with current (jclouds) branch, should just ‘fast forward’)
* git push origin jclouds  
  (push the latest changes in the local jclouds branch to the remote)
* After this the project did not appear in C:\jbdissertation\scm\branching\workspace workspace and needed to ‘Import Existing Projects’
  + File ->Import …
  + Look for ‘Import Existing Projects’ under Java folder, follow instructions.
* Project imported but processor project did not build because src/java/test folder did not exist, exists in other working folder/workspace because Eclipse created there it but GIT cannot commit empty folders hence the error.
* Went back to other workspace to add a basic JUnit java file and committed and repeated the process. Also, had to remove a src/java/test/resource folder from project properties of processor in C:\jbdissertation\scm\mscdissertation and repeat git commands again.