# 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

## 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”
* To share a folder between Linx 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)

SSH client -> use PuTTY

## 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/>

### 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>

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)

**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