# How to Setup Automation Development Environment

*This will show you how to set up an Eclipse development environment in order to begin using the Automation Test Framework.*

**Table of Contents**

[How to Setup Automation Development Environment 1](#_Toc474242352)

[Application Downloads 4](#_Toc474242353)

[Version Checklist 4](#_Toc474242354)

[Important Pre-Installation Notes 5](#_Toc474242355)

[Get Local Administrator Rights 5](#_Toc474242356)

[Common Installation Directory 5](#_Toc474242357)

[Request Shared Drive Access 5](#_Toc474242358)

[Required Software Downloads 6](#_Toc474242359)

[Download JDK version 1.8.0.xx 6](#_Toc474242360)

[Install Java JDK 6](#_Toc474242361)

[Download Apache Maven 7](#_Toc474242362)

[Install Apache Maven 8](#_Toc474242363)

[Git 9](#_Toc474242364)

[Accessing Git 9](#_Toc474242365)

[Requesting Access to the Charter Automation Git repository from within Charter 9](#_Toc474242366)

[Download and Install Git 9](#_Toc474242367)

[Configuring ssh keys for Git on Your Local Machine 10](#_Toc474242368)

[Sync with Automation Projects in Git 10](#_Toc474242369)

[Download Latest Selenium & Webdrivers 17](#_Toc474242370)

[Install Selenium & Webdrivers 18](#_Toc474242371)

[Download Eclipse IDE 19](#_Toc474242372)

[Install Eclipse 20](#_Toc474242373)

[Jenkins, Optional 23](#_Toc474242374)

[Additional Eclipse Information: 25](#_Toc474242375)

[Hints and tips: 25](#_Toc474242376)

[Possible Errors 27](#_Toc474242377)

[Error: “Eclipse is running in a JRE, but a JDK is required.” 27](#_Toc474242378)

[Error message: “Wrong Compiler version” or “Can’t find Compiler Version xxx.” 28](#_Toc474242379)

[Error message: “TestNG class not found.” 29](#_Toc474242380)

[FAQ 30](#_Toc474242381)

[1. Help, I’m getting I’m getting a “Class not registered : .\com4j.cpp:153” error when I attempt to run a test case. 30](#_Toc474242382)

[Environmental - Variables & Miscellaneous Tips 32](#_Toc474242383)

[Web Browsers 32](#_Toc474242384)

[To Enable Developer tools in Chrome: 32](#_Toc474242385)

[Add Browsers Directories to System Environment Path 32](#_Toc474242386)

[Chrome Browser Launch Security Issue 32](#_Toc474242387)

[DNS Issues 32](#_Toc474242388)

[(For Legacy Time Warner Only) 32](#_Toc474242389)

[Edit Hosts Files 32](#_Toc474242390)

[Requesting Access to the Charter Automation Git repository from outside of Charter 33](#_Toc474242391)

[How to Create a DL 34](#_Toc474242392)

[ADDITIONAL SETUP GUIDES 35](#_Toc474242393)

[SSH Connectivity to Charter Bit bucket 35](#_Toc474242394)

[Step 1 – Software’s to download/install 35](#_Toc474242395)

[Step 2 – Create a RSA Key Pair 35](#_Toc474242396)

[Step 3 – Add SSH key to your bit bucket profile. 36](#_Toc474242397)

[Step 4 - Access the bit bucket using the SSH URL 37](#_Toc474242398)

[Configure Eclipse IDE to connect to BitBucket Repository 40](#_Toc474242399)

[Step-1 Download/Install egit plugin into your eclipse 40](#_Toc474242400)

[Step 2 – Cloning bit bucket repository into git perspective. 41](#_Toc474242401)

[Step 3 – Import the cloned repository into project/package explorer 46](#_Toc474242402)

# Application Downloads

## Version Checklist

Use this table to identify what versions of the following applications we are currently using (As of 02/05/17). You should always get the latest version unless told otherwise.

|  |  |
| --- | --- |
| **Application** | **Version** |
| Java | JDK 1.8.0\_121 |
| Apache Maven | 3.3.9 |
| Eclipse IDE | NEON |
| Git | 2.11.0 |
| Jenkins |  |
| Other |  |

# Important Pre-Installation Notes

### Get Local Administrator Rights

This document ***assumes*** you have administrator rights on the box you are working to do these installations on. Please work to get admin access rights to your box under your id, or have the user svc\_automation set up as a local administrator on the box. Certain installations and configurations of environment variables cannot happen without this step happening first.

### Common Installation Directory

In an effort to be consistent across our dev environments, the recommended install path for all applications in this document is C:\apps. This will help with Jenkins slave setups and also in debugging environments quicker if the need ever arises.

### Request Shared Drive Access

All of our framework documentation is being stored on the network shared X drive at ([\\stl02mfps02\ossautomation$](file:///\\stl02mfps02\ossautomation$)). You must submit a Service Now ticket to gain access to this drive.

# Required Software Downloads

## Download JDK version 1.8.0.xx

Latest JDK <http://www.oracle.com/technetwork/java/javase/downloads/index.html>



### Install Java JDK

1. Run/Unzip the distribution archive, i.e. jdk-8u112-windows-x64.exe to the directory you wish to install JDK 1.8.0\_XX. These instructions assume you chose c:\apps\Java\jdk1.8.0\_112 *(if you choose another directory, replace your directory name where applicable)*.
2. Add the **JAVA\_HOME** system environment variable by opening up system properties (WinKey + Pause). Select the “Advanced system settings” link, then selecting the “Advanced” tab, and then pressing the “Environment Variables…” button. In the System Variables section, add a new variable, JAVA\_HOME, or edit the existing variable with the value c:\apps\Java\jdk1.8.0\_XX *(where XX represents your actual sub-version numbers.)* Be sure to omit any quotation marks around the path even if it contains spaces.

JAVA\_HOME

c:\apps\Java\jdk1.8.0\_XX

1. In the same dialog, create/update the **PATH** system variable. Add the value %JAVA\_HOME%\bin

Path  
%JAVA\_HOME%\bin;

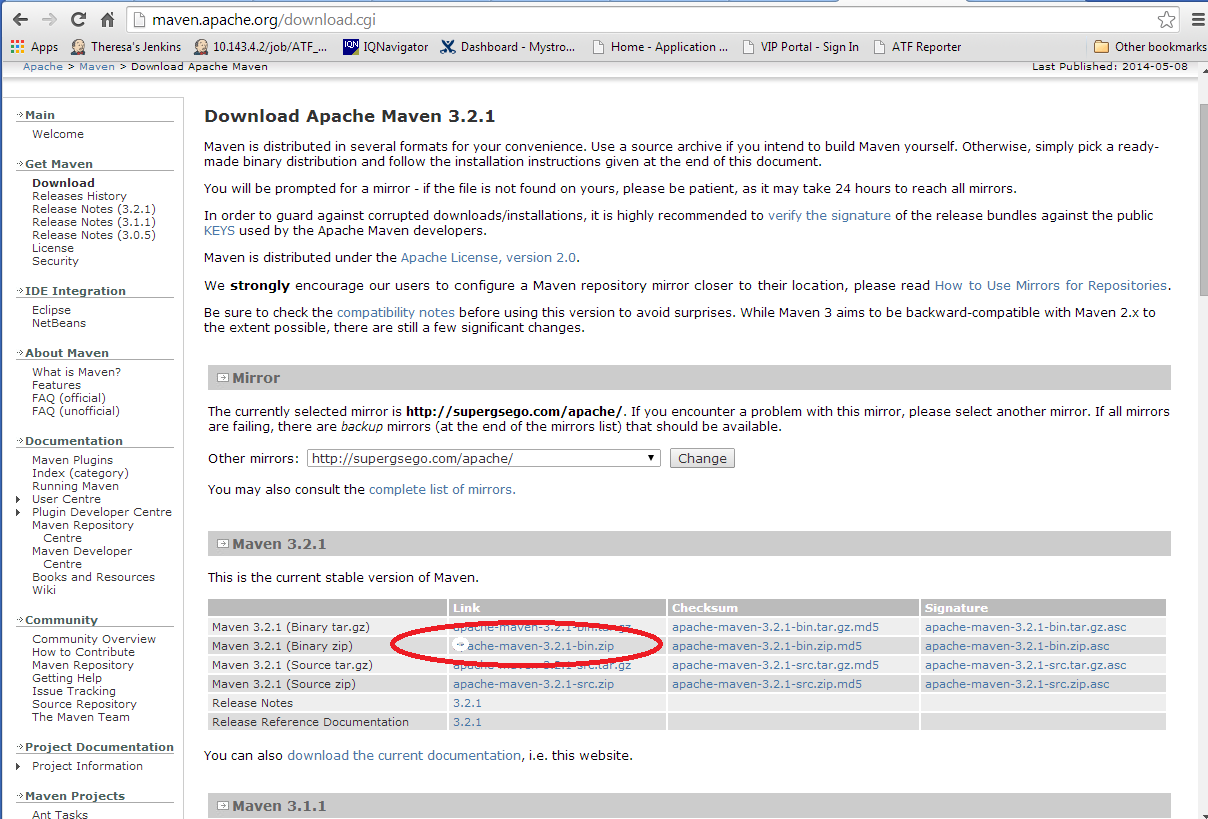
1. For a **Mac computer**, the Java JDK jdk-8u112-macosx-64.dmg is usually installed in /Library/Java/JavaVirtualMachines. Follow the instructions from Oracle.com for installing the JDK Mac version. Add JAVA\_HOME system variable and PATH to your **home .bashrc** and to **/etc/launchd.conf**. They should look something like this:

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0\_112.jdk/Contents/Home

export PATH=$JAVA\_HOME:$PATH

## Download Apache Maven

Latest Apache Maven (Binary zip) <http://maven.apache.org/download.cgi>

****

We are currently using Maven 3.3.9, but **check with your automation lead** before installing.

### Install Apache Maven

1. Unzip the distribution archive, i.e. apache-maven-3.x.x-bin.zip to the directory you wish to install Maven 3.x.x. These instructions assume you chose C:\maven *(if you choose another directory, replace your directory name where applicable)*. The subdirectory apache-maven-3.x.x (**where x.x represents your actual sub-version numbers**) will be created from the archive.
2. Add the **M2\_HOME** environment variable by opening up the system properties (WinKey + Pause), selecting the "Advanced" tab and the "Environment Variables" button, and then adding the M2\_HOME variable in the **System variables** with the value C:\maven\apache-maven-3.x.x. Be sure to omit any quotation marks around the path even if it contains spaces. **Note**: For Maven < 2.0.9, also be sure that the M2\_HOME doesn't have a '\' as last character.

M2\_HOME  
C:\apps\apache\maven

1. In the same dialog, add the **M2** environment variable in the **User variables** with the value %M2\_HOME%\bin.

M2  
%M2\_HOME%\bin

1. In the same dialog, update/create the **Path** **User variable** in the user variables and prepend the value %M2% to make Maven available in the command line.

Path  
%M2%;

1. **Optional**: In the same dialog, add the **MAVEN\_OPTS** environment variable in the **User variables** to specify JVM properties, e.g. the value -Xms256m -Xmx512m. This environment variable can be used to supply extra options to Maven.

MAVEN\_OPTS  
-Xms256m -Xmx512m

1. For a **Mac computer**, add these system variables to your home .bashrc and to /etc/launchd.conf. They should look something like this:

export M2\_HOME=/maven/apache-maven-3.x.x

export M2=$M2\_HOME/bin

export MAVEN\_OPTS="-Xms256m -Xmx512m"

export PATH=$JAVA\_HOME/bin:$M2:$PATH

1. Reboot your machine to apply the system environment variables.
2. Open a newcommand window (*Winkey + R* then type cmd) and run mvn --version to verify that it is correctly installed. Run java -version to verify that it is correctly installed

## [Git](https://chalk.charter.com/display/ITARCH/Git)

[Git](http://git-scm.com/) is a source code management system, or version control system. Charter is using Git as part of the [Charter Continuous Integration Environment](https://chalk.charter.com/display/ITARCH/Charter+Continuous+Integration+Environment), we are using the Bitbucket product for the Git server.

### Accessing Git

**Charter's Git server:** [https://git.corp.chartercom.com:8443](https://git.corp.chartercom.com:8443/)  - Use Chrome or Firefox, **Internet Explorer is not recommended.**

### Requesting Access to the Charter Automation Git repository from within Charter

1. Send a message to Mark Elking to be added to the email list [DLApolloAutomationTeam@charter.com](mailto:DLApolloAutomationTeam@charter.com) .
2. All people who are added to this email list will automatically have access to the bit bucket code.
3. For any GitHub questions, please contact the DL-IT-ContinuousIntegration [DL-IT-ContinuousIntegration@charter.com](mailto:DL-IT-ContinuousIntegration@charter.com)

### Download and Install Git

* 1. Download the latest version of git from <https://git-scm.com/downloads>
  2. Install git into C:\apps\git

### [Configuring](https://chalk.charter.com/display/ITARCH/How+To+Create+a+DL) ssh keys for Git on Your Local Machine

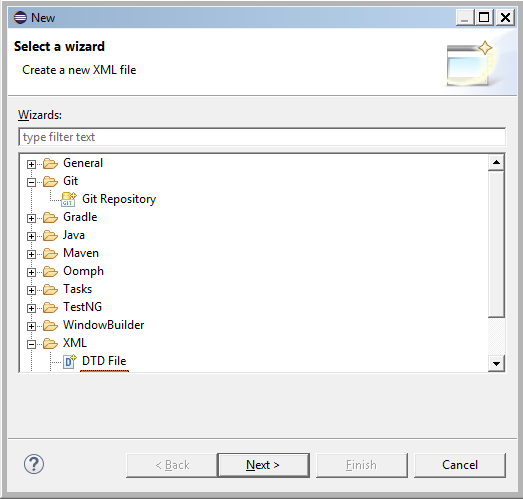
For a detailed explanation, please refer to this guide document: [SSH\_Connectivity](#SSH_Connectivity)

1. After installation completes, go to the “Start Menu” and find the Git folder and launch Git Bash.
2. At the Git Bash prompt, type ‘ssh-keygen’ (with no quote marks)
3. For the next three prompts, just accept the default values by pressing ‘enter’
4. Once generated, you will see 2 files created (id\_rsa & id\_rsa.pub) under the .ssh folder of user home directory.
5. Open the id\_rsa.pub file using notepad and copy the content of the file.
6. Login to bit bucket server using your PID / Charter domain password.
7. Go to **Manage Account** option under your bit bucket user icon on top right corner.
8. Click on **SSH keys** option and then click on **Add Key** button.
9. Paste the content of id\_rsa.pub file from notepad into key text field and click on **Add Key** button

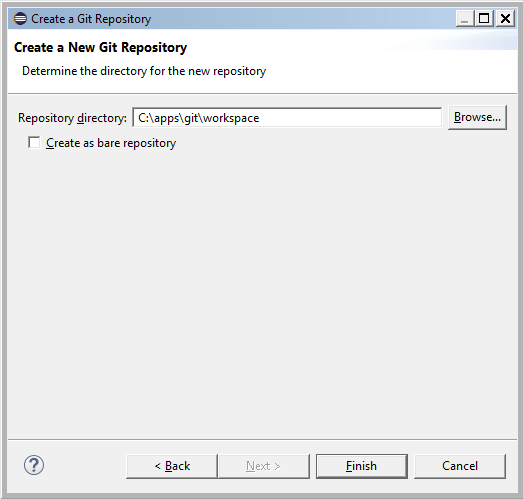
### Sync with Automation Projects in Git

#### Add projects to Git

1. Open Eclipse. This time when you open it, you may choose a different workspace or rename your workspace.
2. From the Eclipse menu, select *File > New > Other > Git > Git Repository > Next*.

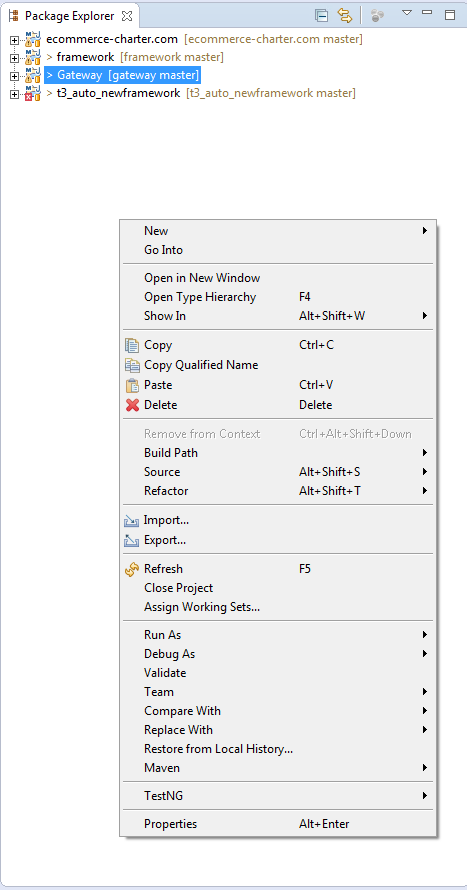


1. If you haven’t yet, set up a git workspace where the code will be pulled out of and placed on your machine.

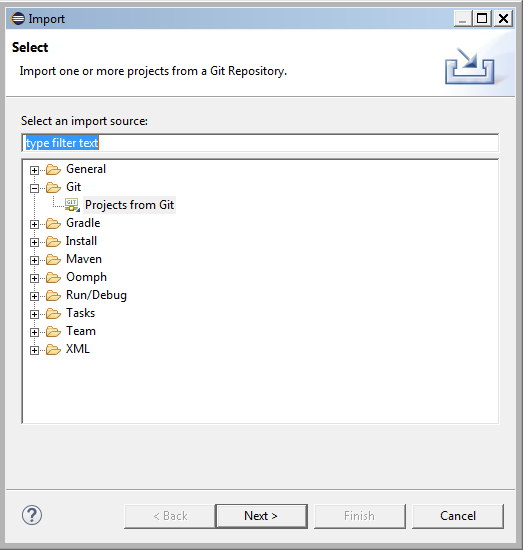


#### Import from a Git Repository

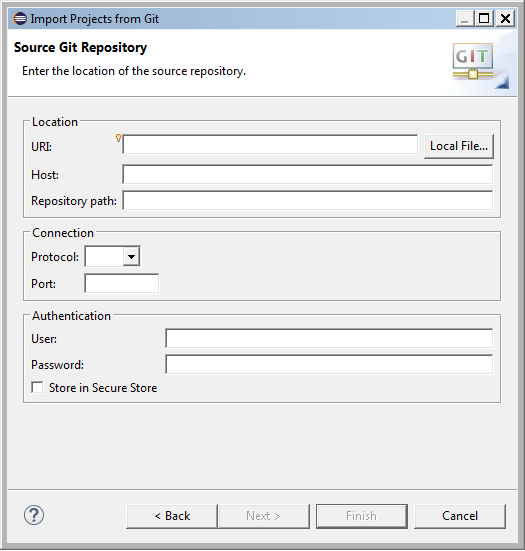
* 1. Right click in the white space in Package Explorer and select *Import*.



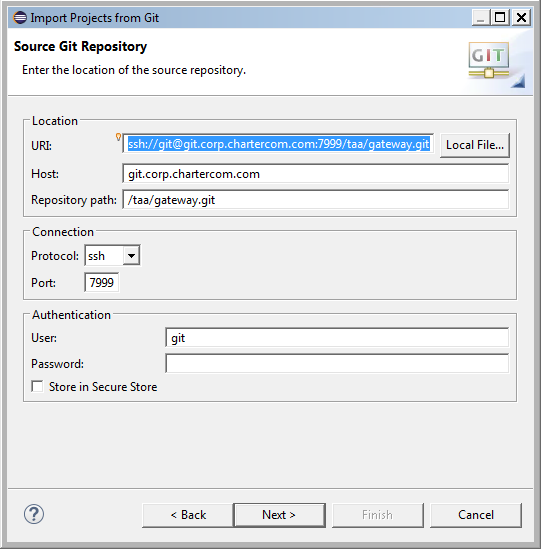
* 1. Select *Git > Projects from Git > Next > Clone URI > Next*



* 1. In the URI, if you copy and paste the URL for the project, it should auto populate the rest of the fields. (i.e.,ssh://git@git.corp.chartercom.com:7999/taa/gateway.git)

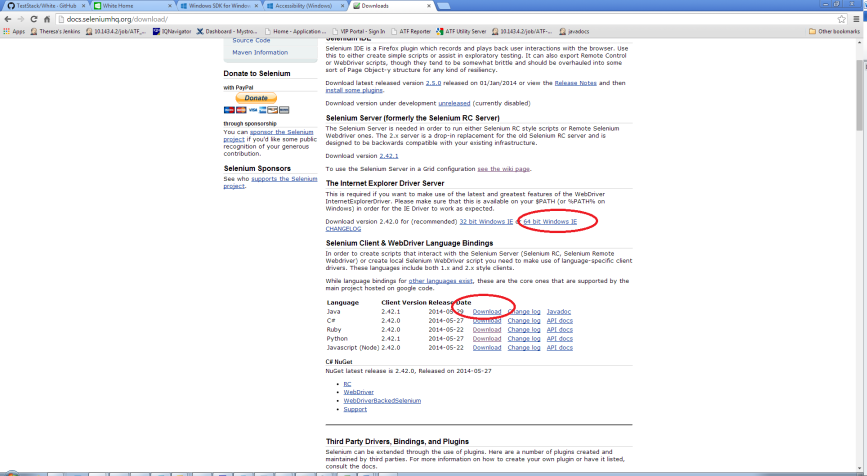


Just change your Authentication credentials to be your network PID/password.

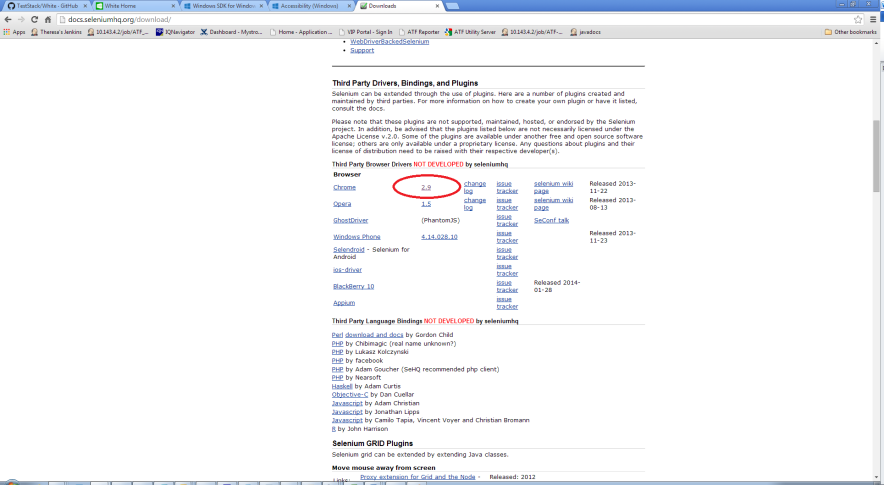


## Download Latest Selenium & Webdrivers

* 1. Go to <http://docs.seleniumhq.org/download/> and download the latest “The Internet Explorer Driver Server” (32-bit Windows IE version). Also download the latest “Selenium Client & WebDriver Language Bindings” for the **Java** client.



* 1. Under the “Third Party Drivers, Bindings, and Plugins” section, download the latest firefox (Mozilla GeckoDriver) and Google Chrome Driver available. (Optional) If you have a Windows 10 OS, you can also download the latest Microsoft Edge Driver.



### Install Selenium & Webdrivers

1. Create this directory C:\apps\Selenium.
2. Unzip all of the web drivers that you have downloaded so that the executables will be under C:\apps\Selenium and not a different folder. For instance, you should see C:\apps\Selenium\IEDriverServer.exe if you have unzipped properly.
3. Unzip the Java Selenium Client into the same directory such that the contents of the zip file are under C:\apps\Selenium and not its own folder (i.e., C:\apps\Selenium\selenium-java-3.0.0-beta4). If everything was done properly, you should see a C:\apps\Selenium\lib folder and a .jar file now.
4. Add/Edit the **PATH** environment variable by opening up the system properties (WinKey + Pause), selecting the "Advanced" tab and the "Environment Variables" button, and then adding/editing the PATH variable in the **System variables** with the value C:\apps\Selenium. Be sure to omit any quotation marks around the path even if it contains spaces. And save afterwards.

PATH  
C:\apps\Selenium

1. To test that your PATH variable was set properly, open a cmd prompt and type ‘chromedriver’. If you see:

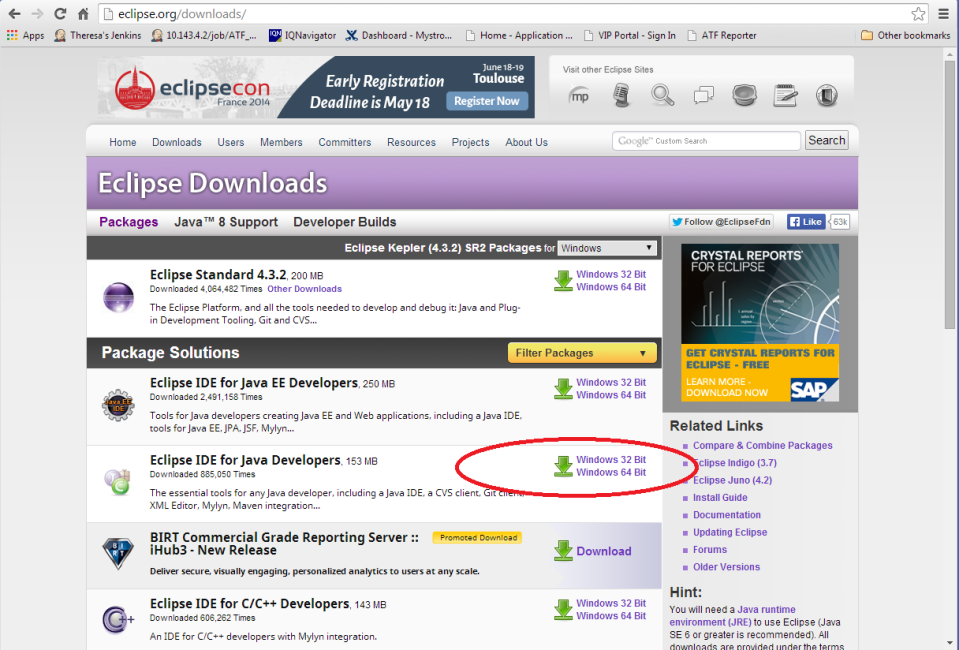
*Starting ChromeDriver 2.22.397933 (1cab651507b88dec79b2b2a22d1943c01833cc1b) on port 9515*

*Only local connections are allowed*.

Then everything worked properly.

## Download Eclipse IDE

Download the latest Eclipse IDE for Java Developers <http://www.eclipse.org/downloads/>



### Install Eclipse

#### Eclipse IDE for Java Developers

1. When you install, extract it to C:\apps *(don't put spaces in names)*; check use folder names for the extraction - it will put it under eclipse
2. Note: 32bit vs 64bit – this must match what you downloaded for java.

#### Configure Eclipse to use JDK

Edit eclipse.ini located in C:\apps\eclipse

1. For Windows PC, if you can’t locate eclipse.ini, change your folder options: go to C:\apps\eclipse > Tools > Folder Options > View > Under Advanced settings and check to see if “Hide extensions for known file types” is selected, if so deselect it.   
   For a Mac computer, if you can’t locate eclipse.ini, in Finder select Eclipse.app and click with the *Control key held down*, select “Show Package Contents” from the popup menu that appears. The eclipse.ini file is under the Contents/MacOS directory in Eclipse.app.
2. When you edit eclipse.ini don’t use notepad. Use either WordPad or a different editing tool.

Add the following line

-vm

C:\apps\Java\jdk1.8.0\_xx\bin or /Library/Java/JavaVirtualMachines/jdk1.8.0\_xx.jdk/Contents/Home/bin

Example eclipse.ini *(Note that these settings may need “tweaking”, but try these first)*:

-startup

plugins/org.eclipse.equinox.launcher\_1.1.0.v20100507.jar

--launcher.library

plugins/org.eclipse.equinox.launcher.win32.win32.x86\_1.1.0.v20100503

-product

org.eclipse.epp.package.java.product

--launcher.defaultAction

openFile

--launcher.XXMaxPermSize

256M

-showsplash

org.eclipse.platform

--launcher.XXMaxPermSize

256m

--launcher.defaultAction

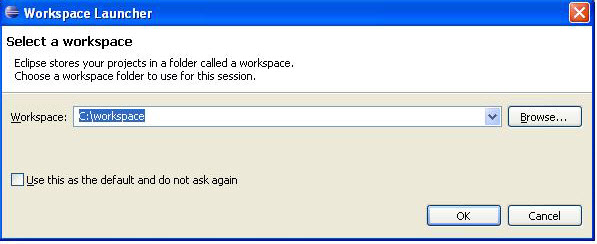
openFile

-vm

C:\Program files\Java\jdk1.8.0\_25\bin

-vmargs -Xms300m -Xmx590m

1. Find c:\apps\eclipse\eclipse.exe, make a shortcut to desktop; then *double-click* on it to launch.
2. Workspace Launcher, change default workspace to "C:\apps\eclipse\workspace" *(this is just temporary, but necessary)*



#### Install the TestNG Eclipse plug-in

1. Open Eclipse.
2. Select *Help > Install New Software…*
3. *Enter* “http://beust.com/eclipse” into the *“Work with:”* space and *press Enter*. Pressing Enter should cause Eclipse to update list of available plugins and components.
4. *Select “TestNG”* to expand it and *select* the most recent version and *click “Next”* twice.
5. *Accept* the terms of the license agreement and *select “Finish”**.*
6. Restart Eclipse if prompted.

#### Install the Maven Eclipse m2eclipse Plugin

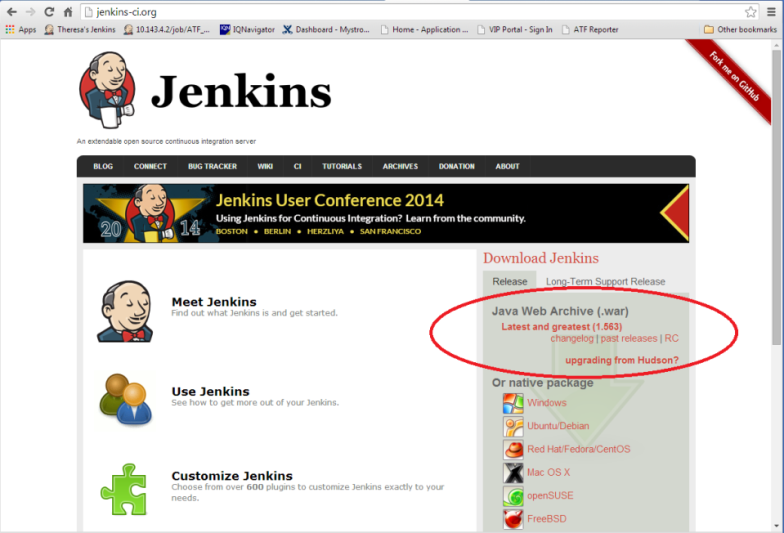
To install this plugin in the Eclipse IDE:

1. Open Eclipse.
2. Select *Help > Install New Software*. This should display the "Install" dialog.
3. Check to see if m2e is already installed with your Eclipse version. From the ‘Install’ Dialog, *click ‘What is already installed?’* link. *Expand* the ‘*Eclipse IDE for Java Developers’* and then *expand* the *‘EPP Java Package’*. If you see do NOT see ‘m2e’ among the packages, close that dialog and proceed with the following steps. If you DO see ‘m2e’, jump to step 10.
4. Paste the Update Site URL “http://download.eclipse.org/technology/m2e/releases” into the field named "*Work with:"* and press *Enter*. Pressing Enter should cause Eclipse to update list of available plugins and components.
5. *Expand* the component listed under m2eclipse: "*Maven Integration for Eclipse (Required)*" and *check* the latest version.
6. Click *Next*. Eclipse will then check to see if there are any issues which would prevent a successful installation.
7. Click *Next* and *agree* to the terms of the Eclipse Public License v1.0.
8. Click *Finish* to begin the installation process. Eclipse will then download and install the necessary components.
9. Once the installation process is finished, restart Eclipse *(you should be prompted to restart)*.
10. Set the Eclipse default **JRE** to the **JDK** (verify this is correct, if you find it existing):  
    Eclipse, *Window > Preferences > Java >* *Installed JREs*, Add button, ‘Standard VM’, Next, Directory C:\Program Files\Java\jdk1.8.0\_xx (find the path on your machine to the JDK), *Finish.* Make sure there is a check mark on this one if you have more than one here.   
    Expand the *Installed JREs* to view the *Execution Environment*. *Select* JavaSE-1.8 and *Checkbox* your **JDK** under *Compatible JREs*. If you have any other Java versions installed, (such as Java 1.7) follow these steps for it.  
    Press *OK* to close the Preferences Dialog.

## Jenkins, Optional

**Optional, check with your manager** before loading.

Latest Jenkins <http://mirrors.jenkins-ci.org/war/latest/jenkins.war> (<http://jenkins-ci.org/>)

****

This is only required if you want to setup your own personal Jenkins server for testing purposes. Otherwise, we will use the Jenkins server that has been installed at:

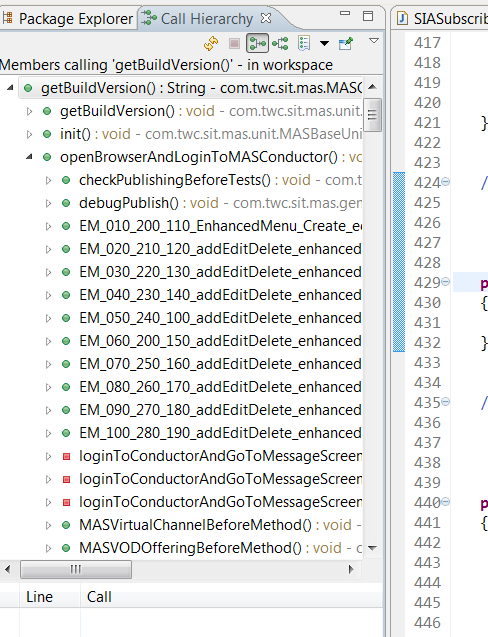
[http://kstlitqavm033.corp.chartercom.com:8080](http://kstlitqavm033.corp.chartercom.com:8080/)

Contact Mark Elking (mark.elking@charter.com) or Danny Byers ([danny.byers@charter.com](mailto:danny.byers@charter.com)) to have a user created for access to this Jenkins server.

# Additional Eclipse Information:

### Hints and tips:

1. Resource view: typing *ctrl-shift-R* pops up the resources window where you can easily search for java classes or other resource files without having to blindly search for something via the project tree view.
2. Perforce view: To be able to perform most of your Git actions in Eclipse: Go to *Window > Open perspective > other > Git*.
3. Call Hierarchy view: This view is helpful in finding all the places a method is called from. Really useful for figuring out what the values of a parameter should be or in figuring out if a method can be deleted cause it’s not used anywhere. To open it: *Window > Show View > Other... > Call Hierarchy*. To use this view you select a method you want to trace in the Outline view (usually on the right). Drag and drop it into the Call Hierarchy view. For example see Figure below:



1. Once you’ve customized a perspective, you can save it with a unique name. *Right-click* on the perspective in the upper right that you’ve modified and *select “save as”.*
2. Look in *Eclipse > Help* for various tips, such as Keyboard shortcuts.
3. When typing a method or variable name, *ctrl-space* helps with auto-completing the value. This type of Content Assist can be customized by going to: *Windows > Preferences > Java > Editor > Content Assist*.
   1. One customization: make it so that testng.Assert statements are suggested and the appropriate import is automatically added:
      1. Go to ***Window > Preferences > Java > Editor > Content Assist > Favorites*.**
      2. Click ***New Type***.
      3. Enter org.testng.Assert. Click *OK.* *(You can also use the browse.)*
      4. Click *OK*. Now when you are adding an assert statement, ctrl-space to see the different statements suggestions and the appropriate import statement will automatically be added.

## Possible Errors

### Error: “Eclipse is running in a JRE, but a JDK is required.”

If you see this message in your console output or as a popup when you are opening Eclipse, you may need to edit the eclipse.ini file or the shortcut target that launches Eclipse.

#### Solution 1:

Sample eclipse.ini (change to your paths, of course & be sure to backup before you edit):

C:\apps\eclipse\eclipse.exe -vm "C:\apps\Java\jdk1.8.0\_xx\bin\javaw.exe"

-startup

plugins\org.eclipse.equinox.launcher\_1.1.0.v20100507.jar

--launcher.library

plugins\org.eclipse.equinox.launcher.win32.win32.x86\_1.1.0.v20100503

-product

org.eclipse.epp.package.java.product

--launcher.defaultAction

openFile

--launcher.XXMaxPermSize

256M

-showsplash

org.eclipse.platform

--launcher.XXMaxPermSize

512m

--launcher.defaultAction

openFile

-vm

C:\apps\Java\jdk1.8.0\_112\jre\bin\client\jvm.dll

-vmargs

-Xms128m

-Xmx640m

-Dosgi.requiredJavaVersion=1.5

-XX:PermSize=256M

-XX:MaxPermSize=512M

Please note the following:

-vm MUST be before –vmargs

The c:\javaJDK must be on a different line from –vm

Eclipse is looking for jvm.dll and it may not exist in the location specified in this example.

#### Solution 2:

Sample Shortcut Target *(change to your paths, of course)*:

C:\apps\eclipse\eclipse.exe -vm "C:\apps\Java\jdk1.8.0\_xx\bin\javaw.exe"

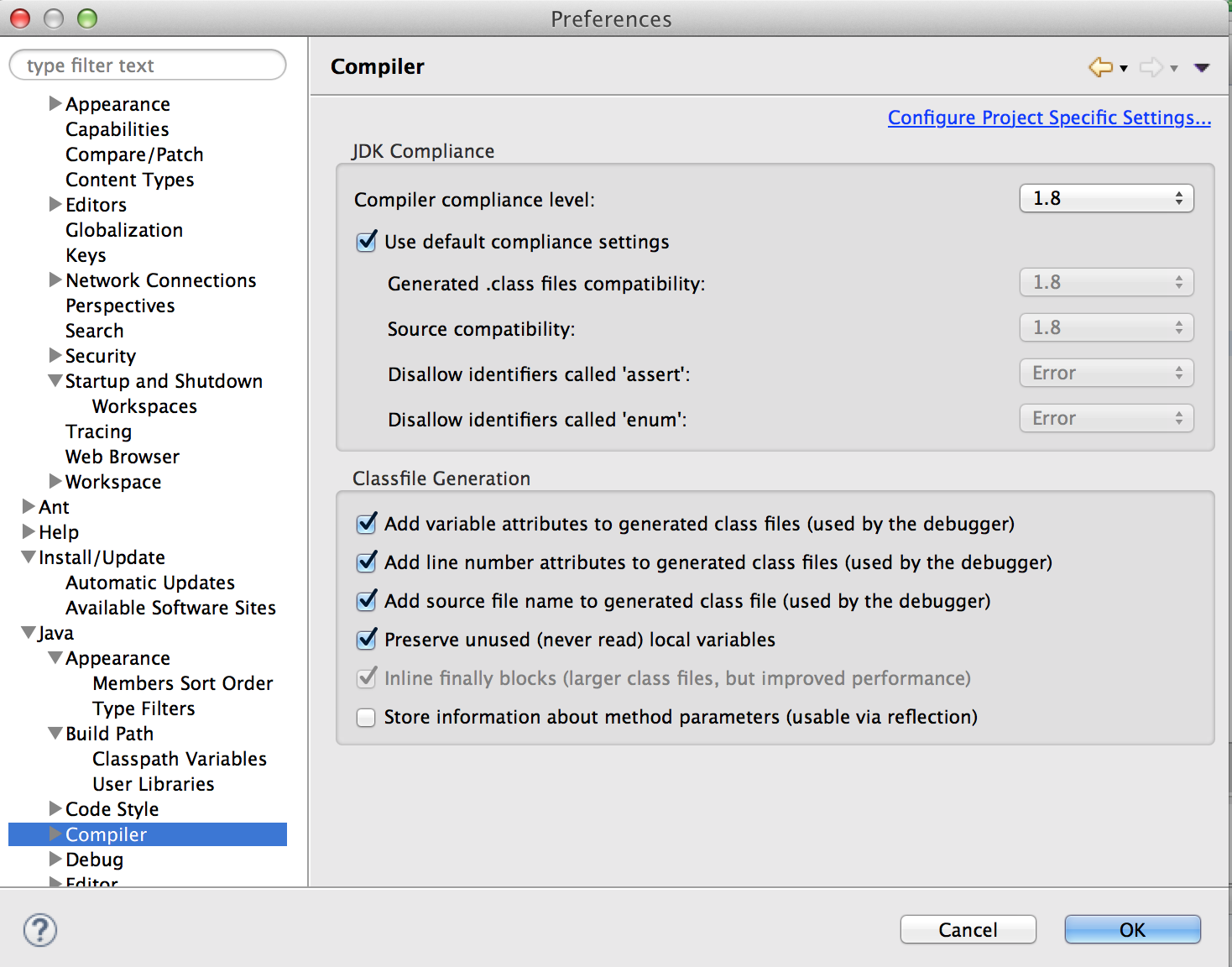
#### Solution 3:

There are notes on this thread that state that it is better to install your jdk in a location other that the default c:\program files\java since supposedly Eclipse doesn’t like the space in the “program files” path. However, some people make it work just fine. See notes at<http://www.coderanch.com/t/488574/vc/Eclipse-running-JRE-JDK-required#2764930>

### Error message: “Wrong Compiler version” or “Can’t find Compiler Version xxx.”

#### Solution 1:

Verify that you have the correct JDK loaded *(to match what the project has specified in the pom.xml)* and that Eclipse has the Compiler compliance level correctly set in *Settings > Java > Complier > JDK Compliance*



### Error message: “TestNG class not found.”

#### Solution 1:

You need to build the projects manually with a Command Console. Open a Command Window, cd into each project directory *(wherever there is a pom.xml file)* and type

mvn clean compile install –DskipTests

# FAQ

### Help, I’m getting I’m getting a “Class not registered : .\com4j.cpp:153” error when I attempt to run a test case.

There could be a number of reasons that might be causing this issue, but there are a few steps you should try first. After applying any of these changes, be sure to Maven update the project. (Right click on project and select Maven > Update Project. Make sure the specific project is checked as is the option to Force Update of Snapshots/Releases, and then select OK.

1. On the machine you are attempting to run the automated test case from, Login as an Administrator and open up a browser and login to the Charter ALM environment ( <http://hp-alm-prod.corp.chartercom.com:8080/qcbin/> ) or (https://alm.corp.chartercom.com/qcbin/start\_a.jsp). This will download specific HP ALM drivers to the local machine, and hopefully resolve the problems for you. But if not the issue persists, continue to the next step.
2. Copy the com4j DLLs from (X:\Automation\Framework\Com4j\com4j-x86.dll) into (c:\windows\system32) and register it from a windows command prompt using the command ‘regsvr32 com4j-x86.dll’ (of course, remove the quotation marks before typing the command.). Also, for 64-bit systems, copy the com4j DLL from (X:\Automation\Framework\Com4j\com4j-amd64.dll) into (c:\windows\SysWOW64) and register it from a windows command prompt using the command ‘regsvr32 com4j-amd64.dll’ (of course, remove the quotation marks before typing the command.). Make sure you cd into the respective directories before you register the DLLs. Hopefully, this will resolve the issues, but if not, continue to the next step.
3. Navigate to the following directory (C:\ProgramData\HP\ALM-Client\12.53.0.0\_952) and copy the OTAClient.dll and WebClient.dll into the following directory (C:\Windows\SysWOW64).
4. Go to the working directory where you have installed git and have downloaded the framework code. Under the main framework folder, there will be a mvn\_install.bat file. Double click this file and let it run. Hopefully, this will resolve the issues, but if not, continue to the next step.
5. If we are still encountering issues, it may be a JAVA versioning problem. We can change the JRE through the Eclipse IDE to use the 32-bit JRE instead of the 64-bit version. In Eclipse, Right Click on Project and select Build/Configure Build Path option. Select the Libraries tab. Add a reference to the 32-bit JRE version at C:\Program Files (x86)\Java (Or wherever your machine has it installed) and remove the C:\Program Files\...jdk reference.
6. The last thing we can do to resolve this issue if it hasn’t been resolved yet is edit the .pom in your application and remove any dependency on com4j.

# Environmental - Variables & Miscellaneous Tips

## Web Browsers

### To Enable Developer tools in Chrome:

Delete the DeveloperToolsDisabled registry key in Software\Policies\Chromium\DeveloperToolsDisabled

### Add Browsers Directories to System Environment Path

Add Firefox and Chrome directories to the System Environment Path variable to help Selenium tests find the browsers.

Path  
;C:\Program Files (x86)\Google\Chrome\Application; C:\apps\Mozilla

### Chrome Browser Launch Security Issue

Regedit:

Computer\HKEY\_CURRENT\_USER\Software\Policies\Google\Chrome\ExtensionInstallBlacklist

Delete Name=1 Data=\* entry.

You may have to do this every time there is a reboot or patch.

## DNS Issues

## (For Legacy Time Warner Only)

### Edit Hosts Files

Edit C:\Windows\System32\drivers\etc\hosts

# Copyright (c) 1993-2009 Microsoft Corp.

#

# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.

#

# This file contains the mappings of IP addresses to host names. Each

# entry should be kept on an individual line. The IP address should

# be placed in the first column followed by the corresponding host name.

# The IP address and the host name should be separated by at least one

# space.

#

# Additionally, comments (such as these) may be inserted on individual

# lines or following the machine name denoted by a '#' symbol.

#

# For example:

# 102.54.94.97 rhino.acme.com # source server

# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.

# 127.0.0.1 localhost

# ::1 localhost

10.143.4.1 atf.mystro.mystrotv.com

10.143.4.2 atf2.mystro.mystrotv.com

10.70.130.33 atf.lab.mystrotv.com

#10.254.75.89 atf.lab.mystrotv.com

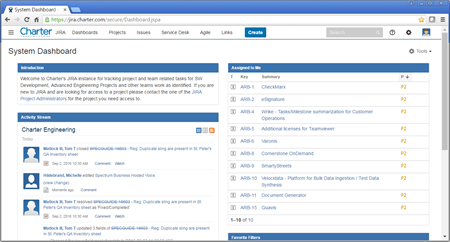
10.100.152.179 nexus.mystro.mystrotv.com

## Requesting Access to the Charter Automation Git repository from outside of Charter

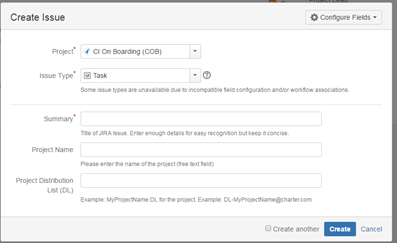
1. Create a ticket in JIRA.   The link below will take you there.

<https://jira.charter.com/secure/Dashboard.jspa>

2. Create a ticket by clicking on the Blue Create button.



3. Select the CI on Boarding (COB) project from the Project Drop Down.



4. Provide the requested information:

Summary:  [a descriptive on line summary]

Project Name: [a meaningful project name]

Project Distribution List (DL): [the DL that will be used for the project]

If you need a distribution list for you project, the link below will explain how.

## [How to Create a DL](https://chalk.charter.com/display/ITARCH/How+To+Create+a+DL)

The project will be created, and then the people in the DL can:

* Login to the Git server using their Windows login here:  [https://git.corp.chartercom.com:8443](https://git.corp.chartercom.com:8443/)
* [Create an SSH key if you need one](https://chalk.charter.com/display/ITARCH/How+to+Use+SSH+keys) for your Stash user account,
* To install your key, Click on the User (upper right), Manage Account, SSH Keys, Add Keys, Paste your public SSH key text into the Windows
* create new code repositories
* clone the new repos
* grant permissions to other users or DL's for this project Selenium & Web drivers

# ADDITIONAL SETUP GUIDES

## SSH Connectivity to Charter Bit bucket

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Remarks |
| 0.1 | Atul Gogte | 10thNov 2016 | initial draft version |
| 1.0 | Atul Gogte | 16th Nov 2016 | Baselined after validating procedure on few more machines. |

### Step 1 – Software’s to download/install

**Charter Shared Drive**

X:\Automation\Z-atul\Softwares

**Internet**

<https://sourceforge.net/projects/sshwindows/>

<https://git-scm.com/downloads>

Install Openssh

Install Git GUI

### Step 2 – Create a RSA Key Pair

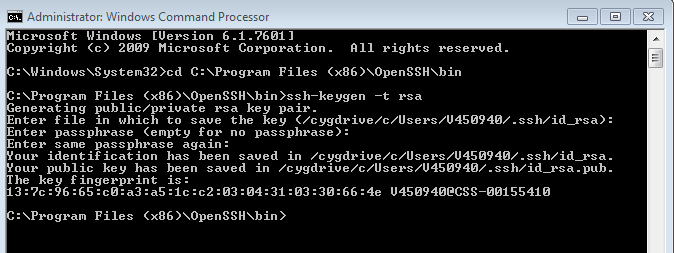
Open command window and travel to bin directory under openssh

Execute **ssh-keygen –t rsa** command

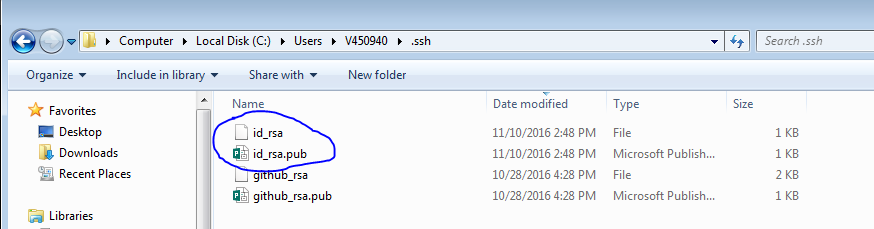
For file prompt, just press enter

For passphrase prompt, just press enter

For passphrase confirmation prompt, again press enter

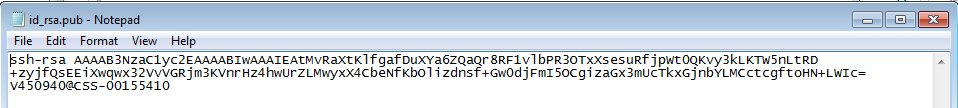


After successfully generating the key pair, you will see 2 files created (id\_rsa & id\_rsa.pub) under the .ssh folder of user home directory. These are your SSH private (id\_rsa) & SSH public (id\_rsa.pub) keys specific to your machine.



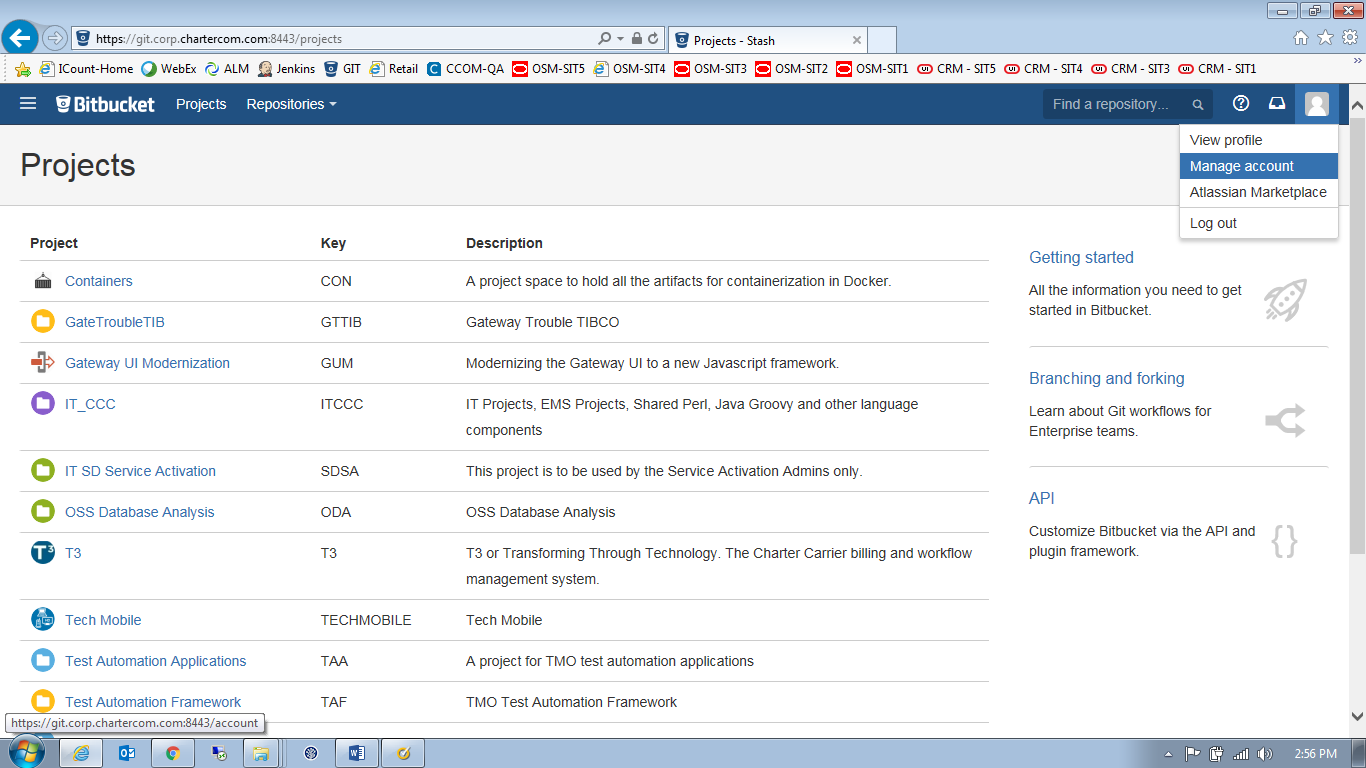
### Step 3 – Add SSH key to your bit bucket profile.

Open the id\_rsa.pub file using notepad and copy the content of the file.

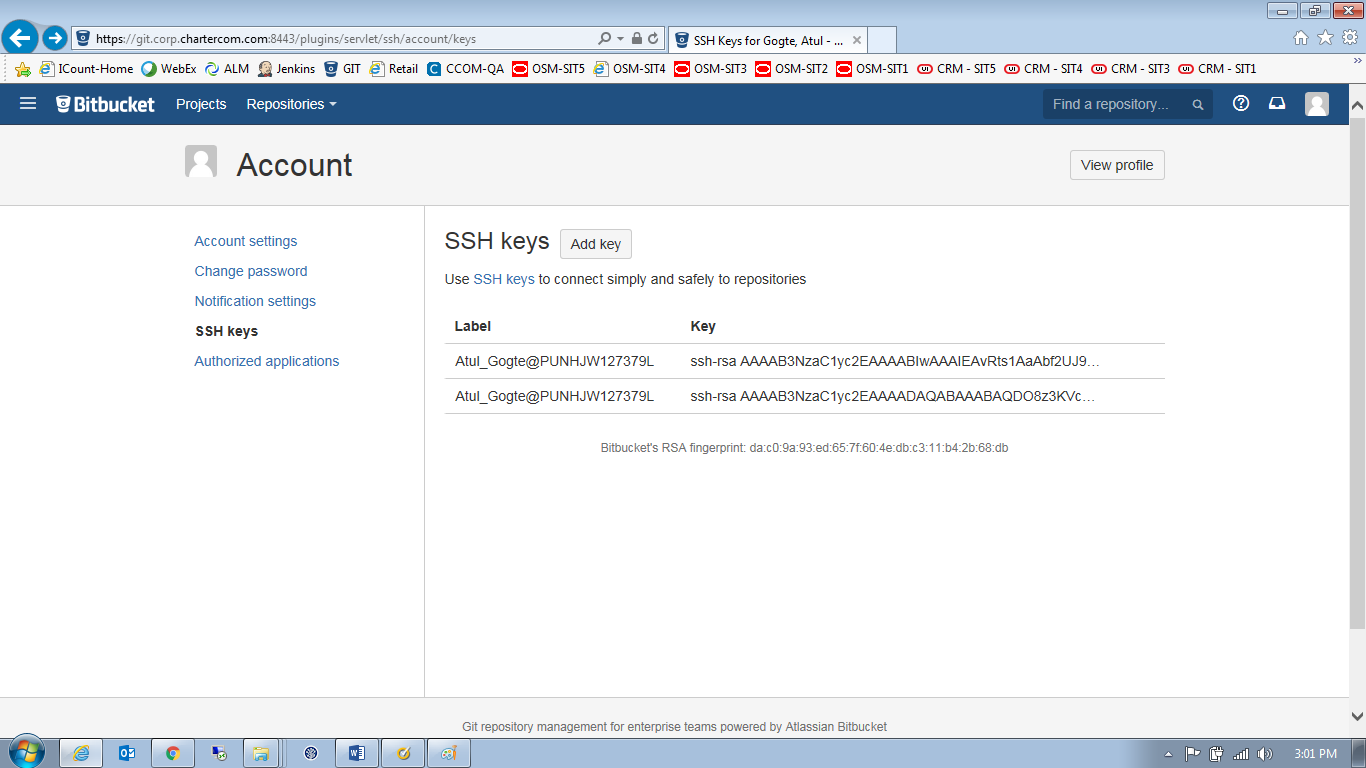


Login to bit bucket server using your PID / Charter domain password.

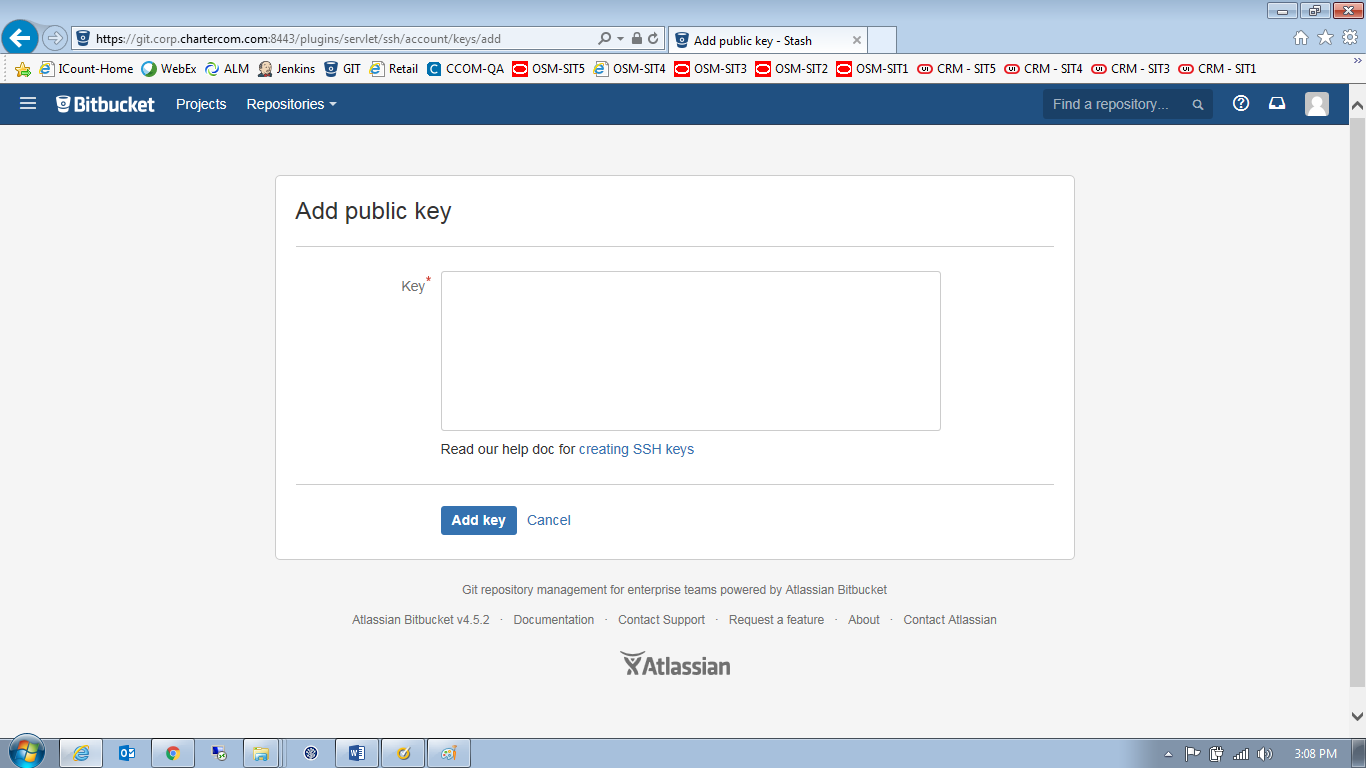
Go to **Manage Account** option under your bit bucket user icon on top right corner.



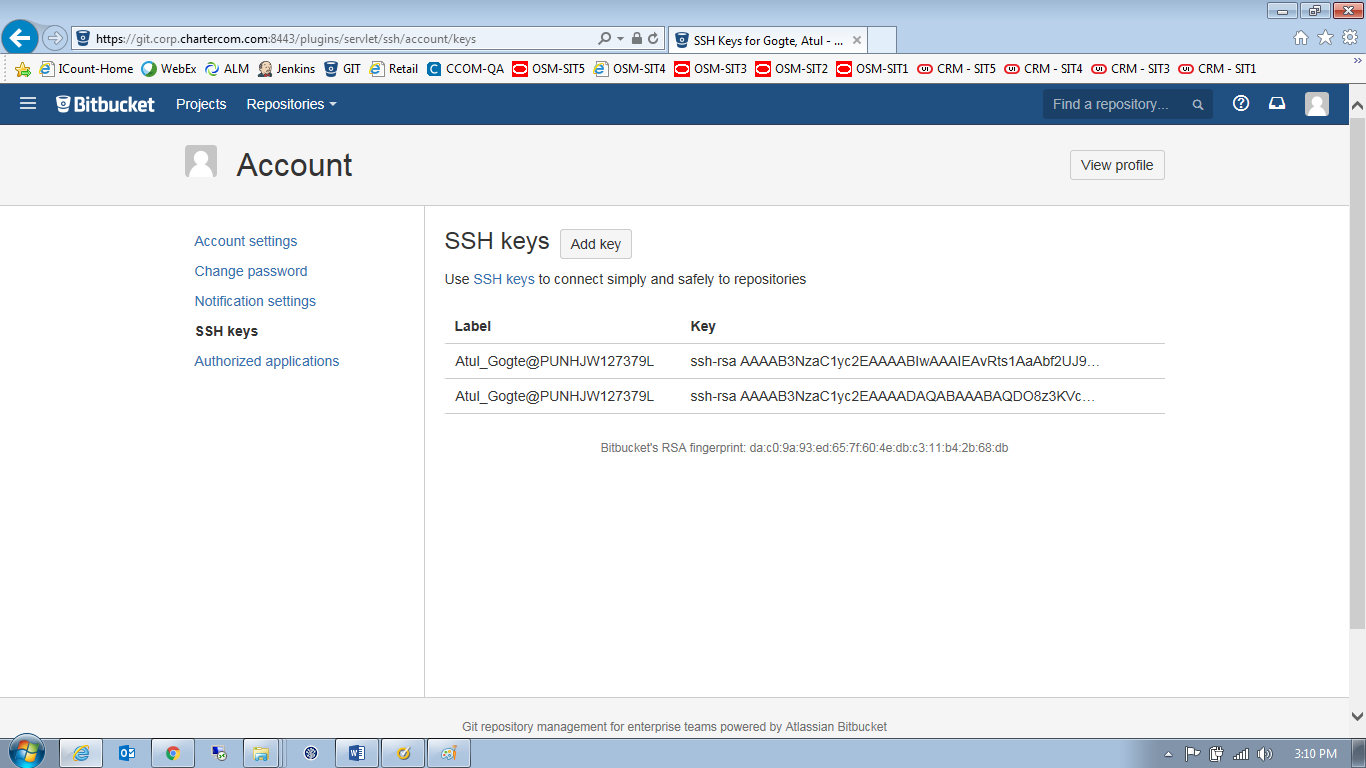
Click on **SSH keys** option and then click on **Add Key** button.



Paste the content of id\_rsa.pub file from notepad into key text field and click on **Add Key** button



Verify that your key being successfully added to your profile



### Step 4 - Access the bit bucket using the SSH URL

Now you can access/create/clone bit bucket repositories using the SSH URL

Format of the SSH URL is as follows

**ssh://<userid>@<bitbucket server name>:<port>/<project name>/<repository name>.git**

Userid = PID

Bit bucket server = git.corp.chartercom.com

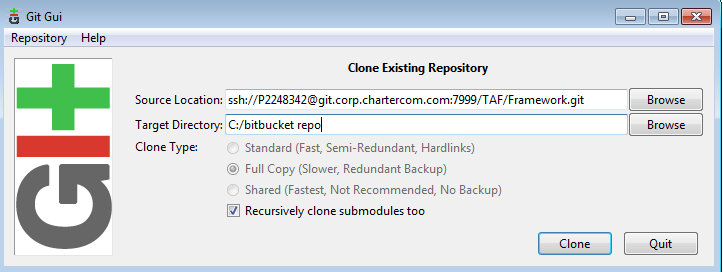
Port = 7999

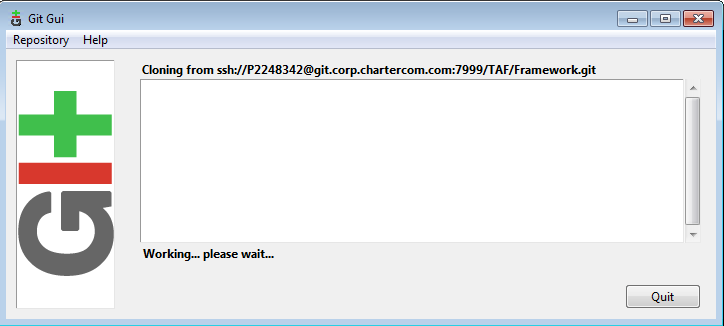
Project = TAF (framework project) or TAA (Application project)

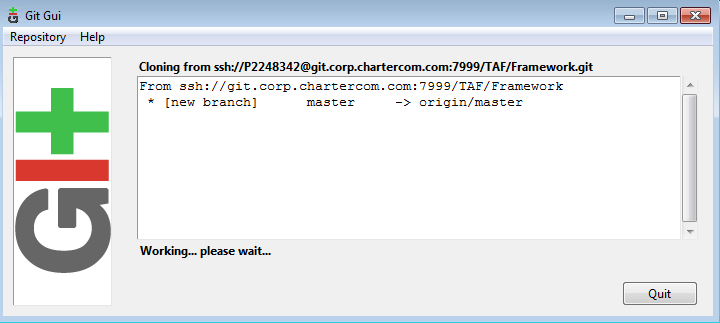
Repository name = Framework or Ecommerce or Gateway etc..

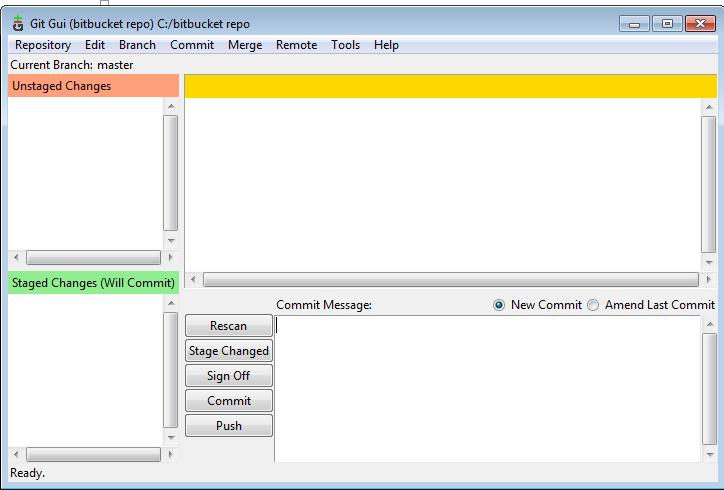
Working Example of SSH URL

**ssh://P2248342@git.corp.chartercom.com:7999/TAF/Framework.git**

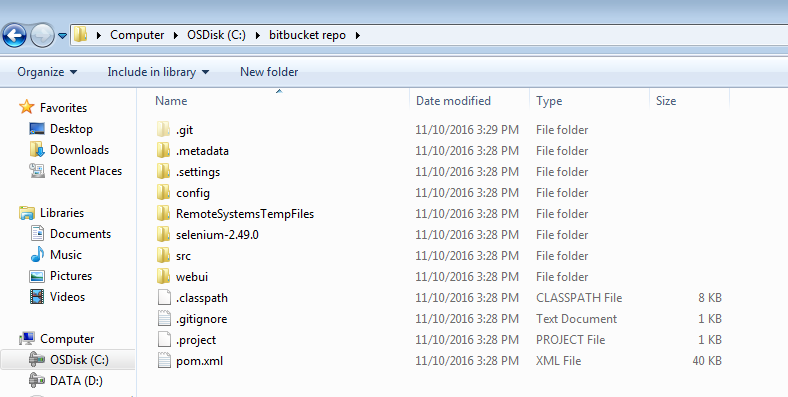








You can see your bit bucket repository is successfully cloned onto your local machine

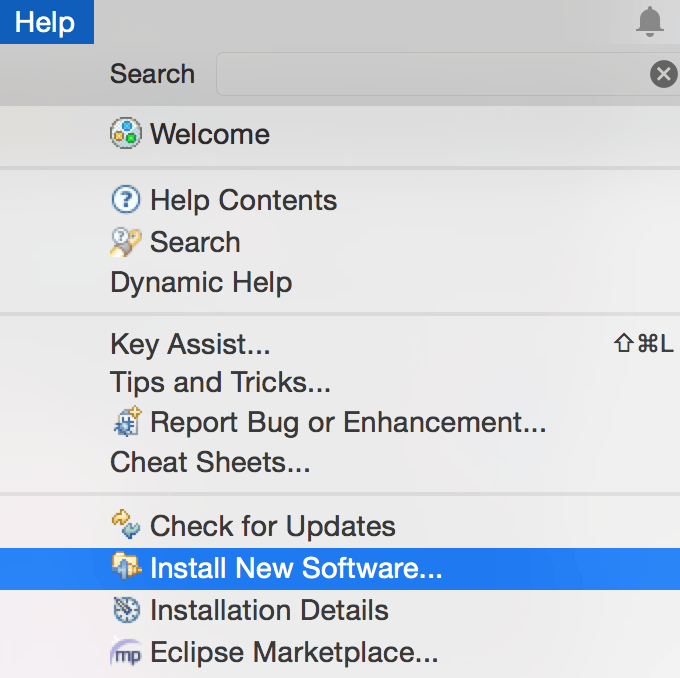


## Configure Eclipse IDE to connect to BitBucket Repository

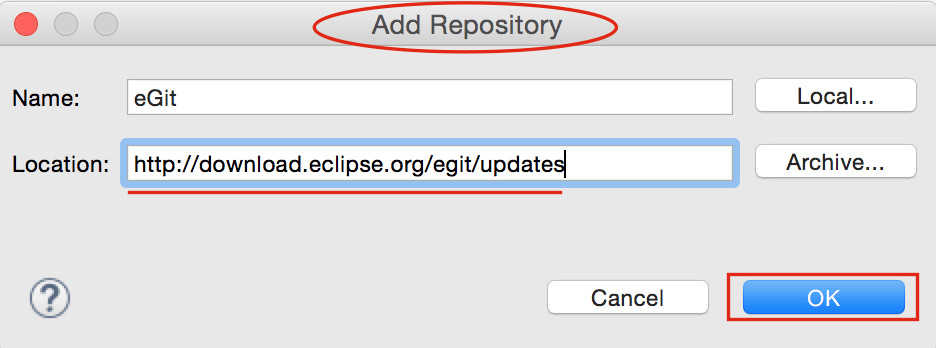
|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Remarks |
| 0.1 | Atul Gogte | 4th Nov 2016 | initial draft version |
| 1.0 | Atul Gogte | 11th Nov 2016 | Updated few sections |

### Step-1 Download/Install egit plugin into your eclipse

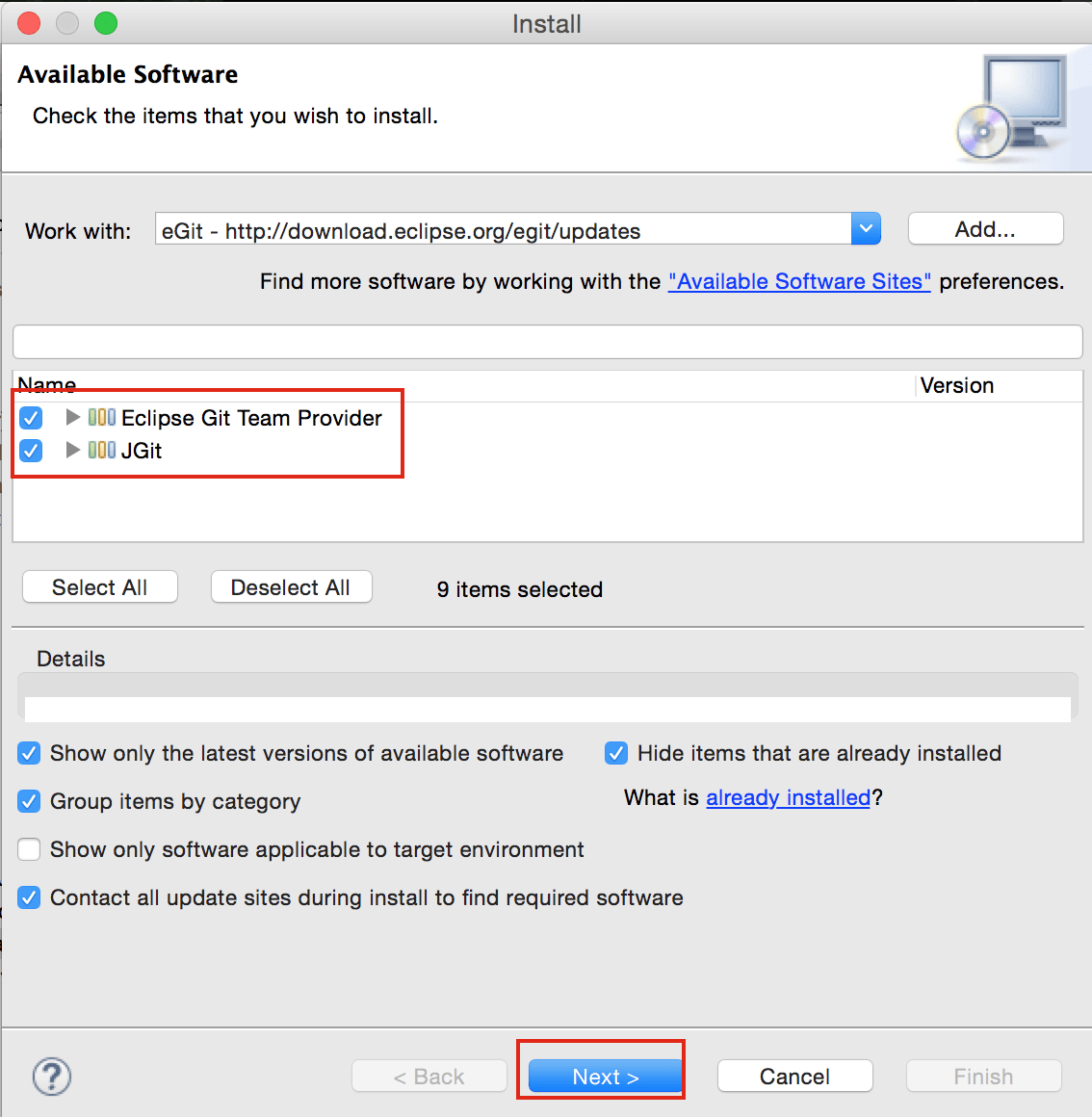
* Open Eclipse
* Click on Help menu
* Click Install New Software



Enter URL: <http://download.eclipse.org/egit/updates>

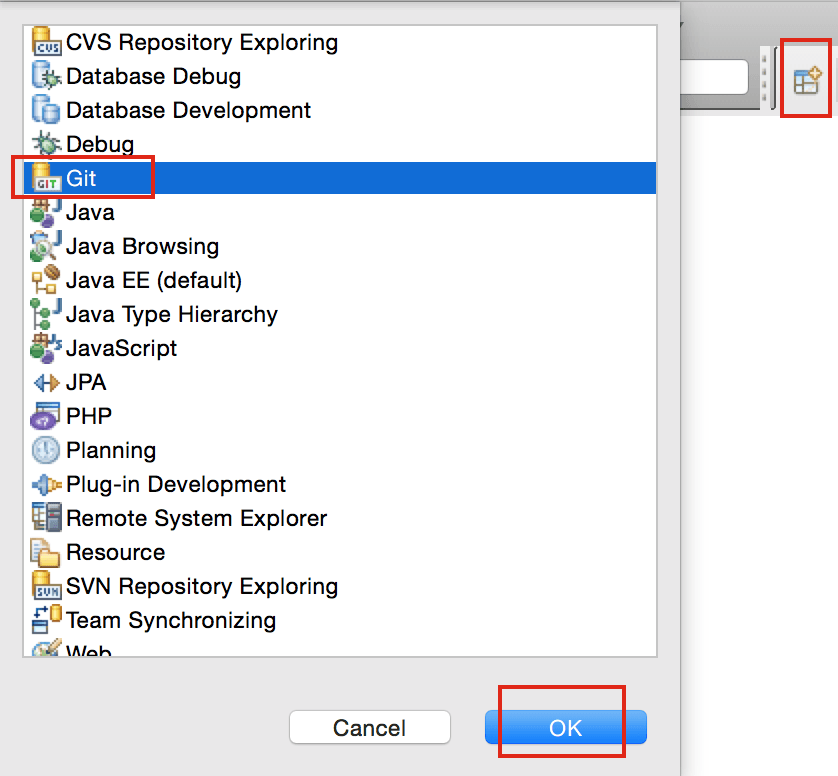


Select Eclipse Git Team Provider and JGit form option and click next and finish install.

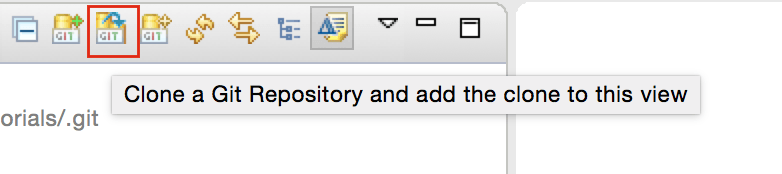


### Step 2 – Cloning bit bucket repository into git perspective.

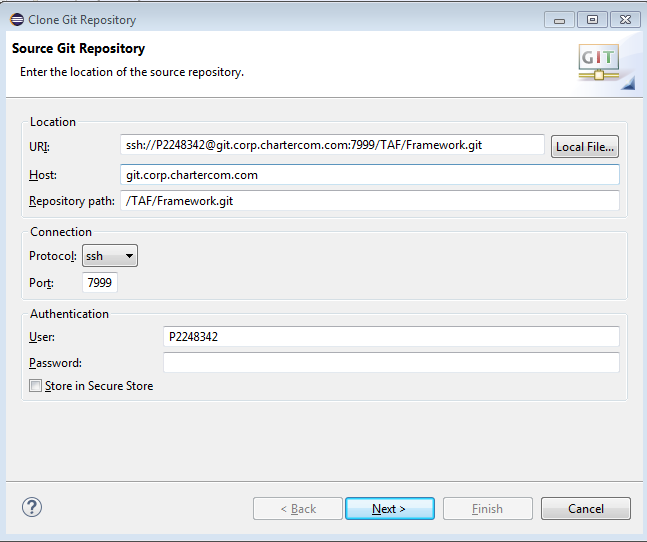
Now Open Perspective and choose Git from list.

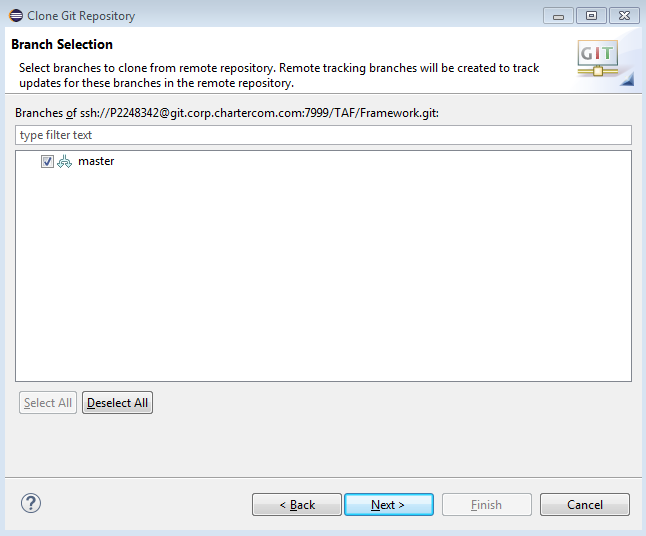


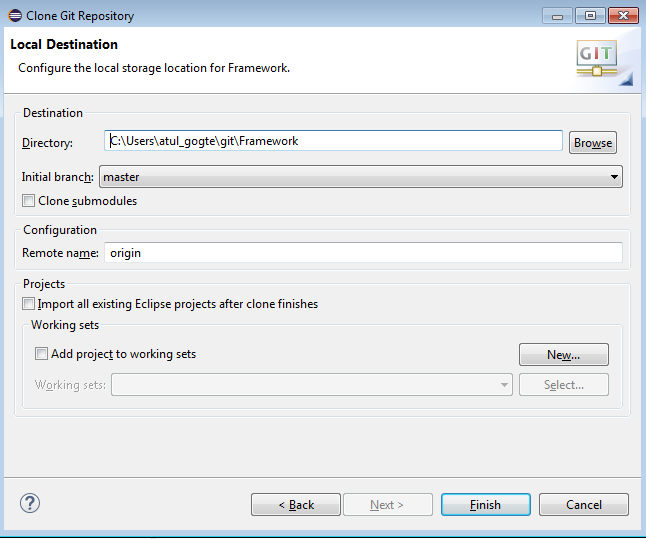
Click Clone Repository Button.

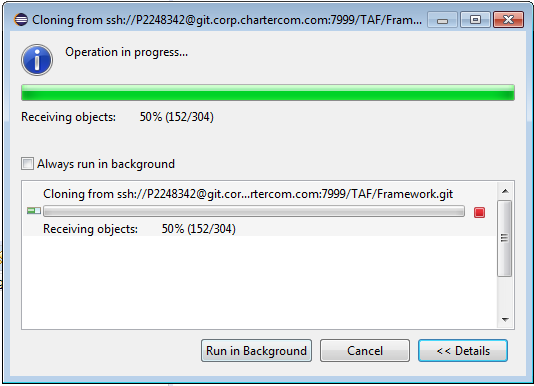


Enter your Bitbucket URL and User Information as mentioned in below diagram. Click Next and Finish. No need to change other configuration in next window.

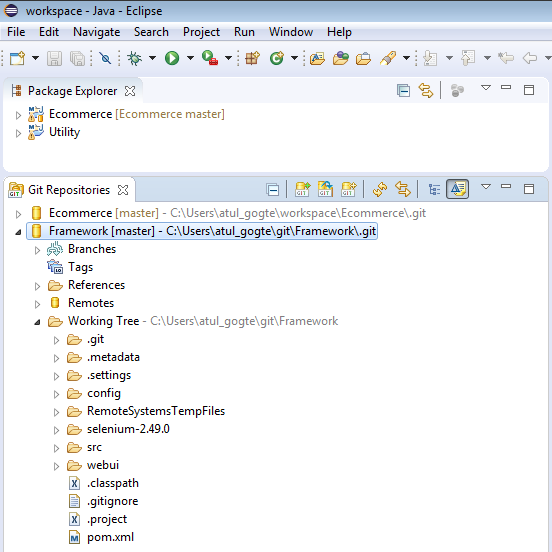








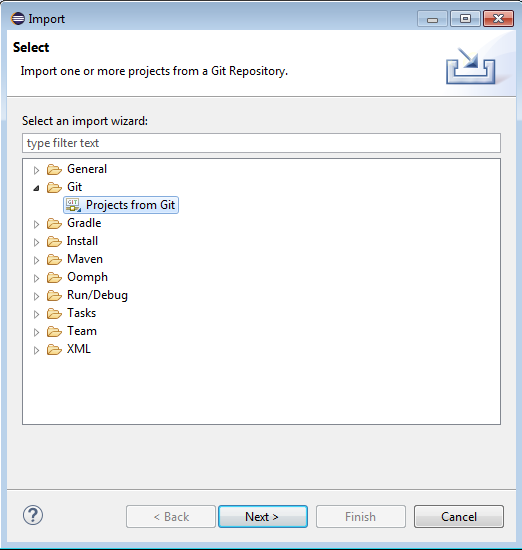
Now you will see the bit bucket repository inside your Eclipse

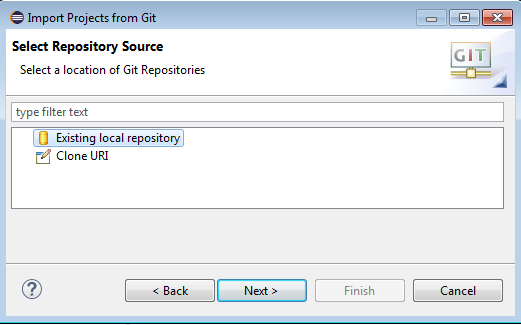


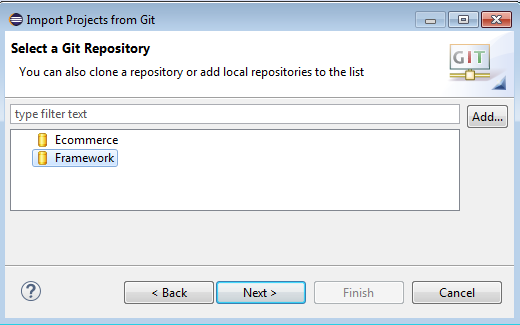
### Step 3 – Import the cloned repository into project/package explorer

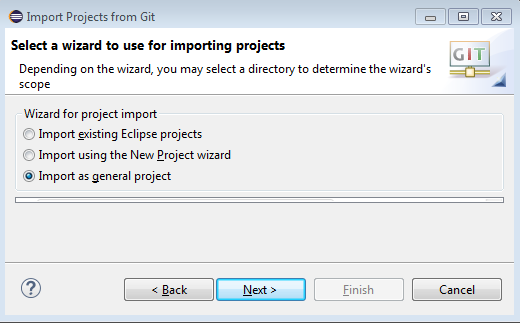
Now you may want to import the project so you can work on the source code.

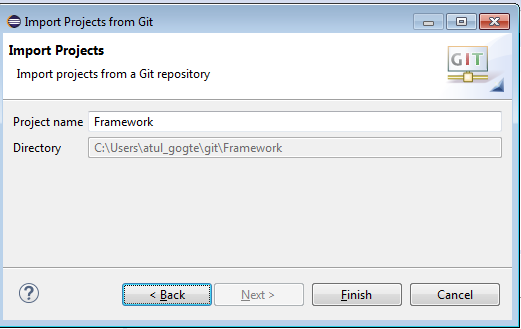
1. Click
   1. ‘Windows’ >
   2. ‘Open Perspective’ >
   3. ‘Resource’
2. Click
   1. ‘File’ >
   2. ‘Import’ >
   3. ‘Git’ >
   4. ‘Projects from Git’ >
   5. ‘Existing local repository’ >
   6. ‘Select a Git Repository’ >
   7. ‘Import as General Project’ >
   8. ‘Next’ >
   9. ‘Finish’



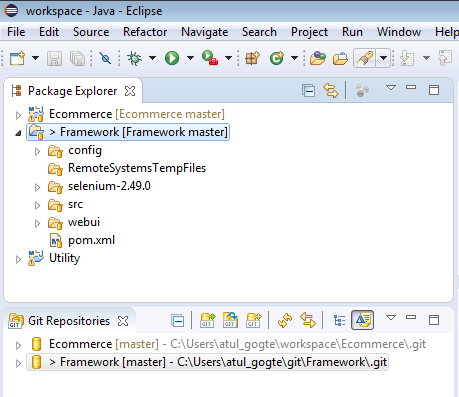








You will see that repository is now imported as project inside Eclipse to be able to make code changes, pull and push the code back and forth with bit bucket repository.



1. The code should then appear in your ‘Project Explorer’ window as a normal project
2. Now make changes to your file as you want
3. Look at ‘Git Staging’ view to see your changed files and Click ‘Commit and Push‘

