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

Application Downloads 3

Version Checklist 3

Java 4

Download and Install JDK version 1.8.0.xx 4

Optional - JDK / JRE version 1.7.0.xx 4

Apache Maven 5

Eclipse IDE 6

Git 7

Selenium & Webdrivers 8

GnuWin wGet 8

Jenkins, Optional 9

Install Instructions 9

Download and Install Java JDK 9

Download and Install Maven 10

Download & Install Perforce 11

Install Eclipse 11

Eclipse IDE for Java Developers 11

Configure Eclipse to use JDK 12

Install the Perforce Eclipse Plug-In 13

Install the TestNG Eclipse plug-in 14

Install the Maven Eclipse m2eclipse Plugin 15

Configure Eclipse 15

Sync with ATF Projects in Perforce 24

Add projects to Perforce 24

Build Projects for First Time 28

Additional Eclipse Information: 30

Hints and tips: 30

Possible Errors 32

Error: “Eclipse is running in a JRE, but a JDK is required.” 32

Error message: “Wrong Compiler version” or “Can’t find Compiler Version xxx.” 33

Error message: “TestNG class not found.” 34

Environmental - Variables & Miscellaneous Tips 34

Web Browsers 34

To Enable Developer tools in Chrome: 35

Add Browsers Directories to System Environment Path 35

Chrome Browser Launch Security Issue 35

DNS Issues 35

Edit Hosts Files 35

# Application Downloads

## Version Checklist

Use this table to identify what versions of the following applications we are currently using.

|  |  |
| --- | --- |
| **Application** | **Version** |
| Java |  |
| Apache Maven |  |
| Eclipse IDE |  |
| Git |  |
| GnuWin wGet |  |
| Jenkins |  |
| Other |  |

## Java

### Download and Install JDK version 1.8.0.xx

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

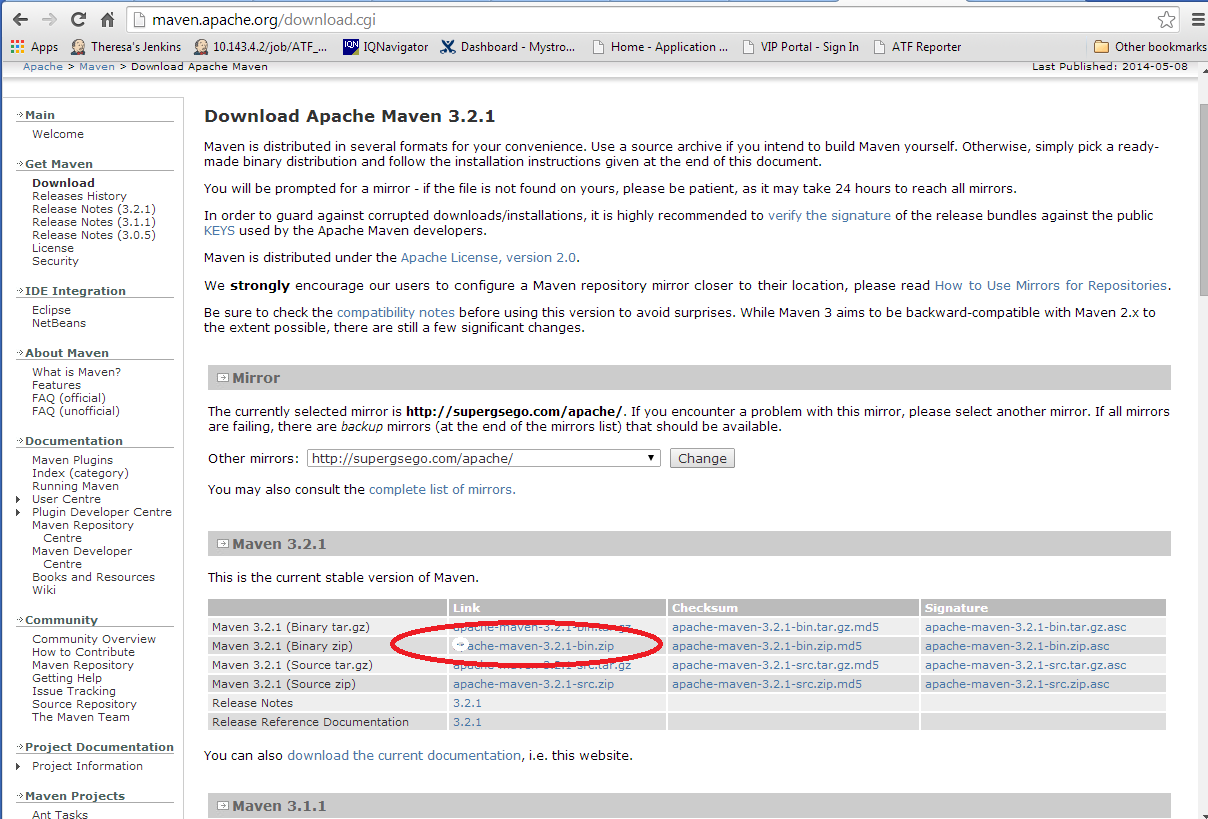


### Optional - JDK / JRE version 1.7.0.xx

You may also need JDK / JRE version 1.7.0.xx for compatibility issues, but **check with your automation lead** before installing.

## Apache Maven

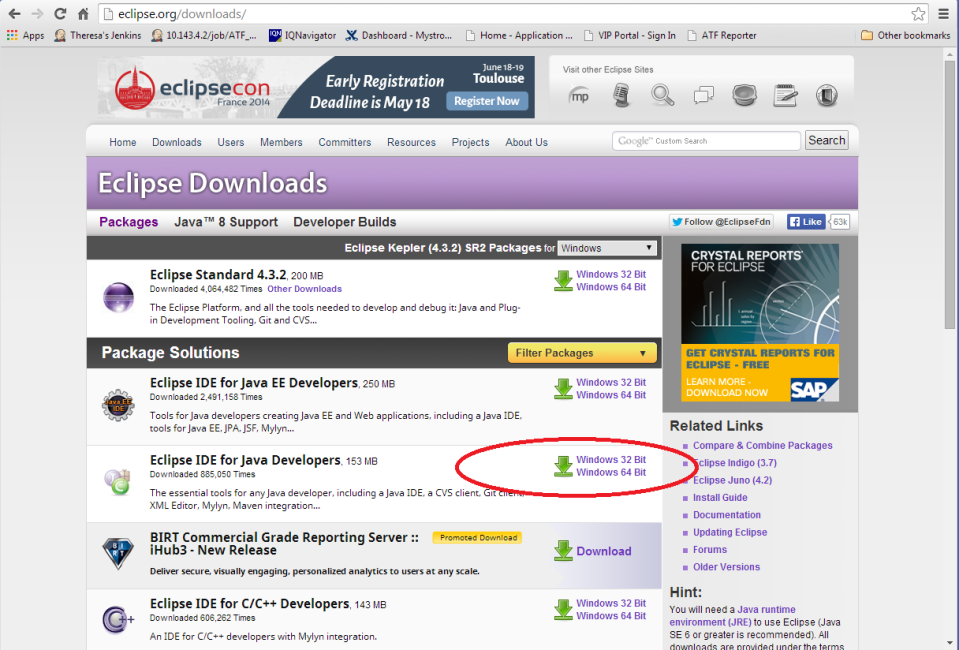
Latest Apache Maven (Binary zip) <http://maven.apache.org/download.cgi>

****

We are currently using Maven 3.2.1, but **check with your automation lead** before installing.

## Eclipse IDE

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



## 

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

[Git](http://git-scm.com/) is a source code management system, or version control system.  Git replaces tools like [CVS](http://en.wikipedia.org/wiki/Concurrent_Versions_System) or [Subversion](https://subversion.apache.org/).

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.

For more information on using Git and best practices please see [Visualized Git practices for team: branch, merge, rebase](http://kentnguyen.com/development/visualized-git-practices-for-team/)

## Using 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.**

**Git Bash command-line** client software (free download): <http://git-scm.com/> **Note:** this is also available in Software Center on Charter machines.

**Eclipse IDE** for Java EE developers (free download) comes with Git support built-in: <http://www.eclipse.org/downloads/>

If you're using **Oracle Design Studio** you may need to use an external Git client.  Git-Bash, a command line tool is available from Charter's Software Center.

[**TortoiseGit**](https://chalk.charter.com/display/ITARCH/How+to+install+TortoiseGit) is a Windows interface to Git.

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

## How should we use Branching in Git?

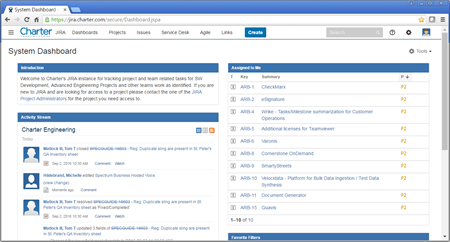
Charter's recommended branching procedure is here: [Git Branching](https://chalk.charter.com/display/ITARCH/Git+Version+Control)

Requesting Access to the Charter Git server

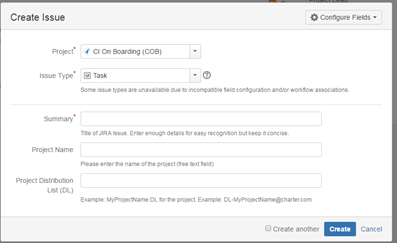
 1.Create a ticket in JIRA.   The link below will take you there.

<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

## Bitbucket

The Git server is a product called Bitbucket: <https://www.atlassian.com/software/bitbucket>

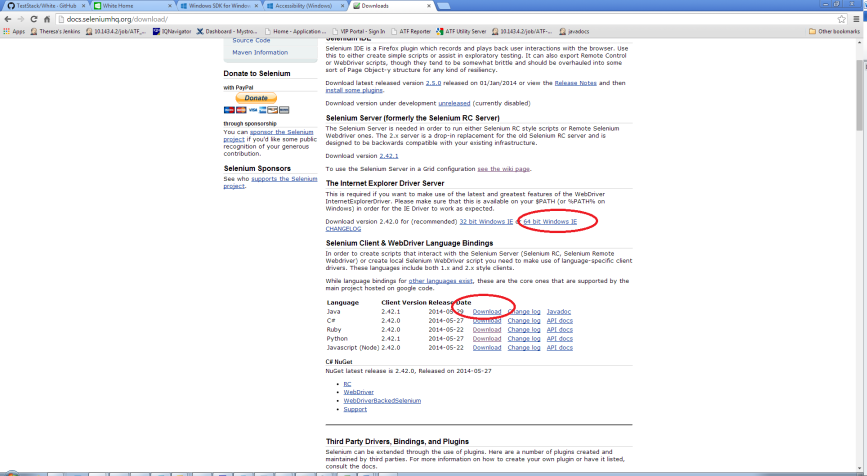
* Windows Login: Stash integrates with the company Windows logins
* Security: Stash allows users to define the user and group permissions using Windows groups or DLs
* Code Review: Stash provides a code review tool that is built-in to the Web UI

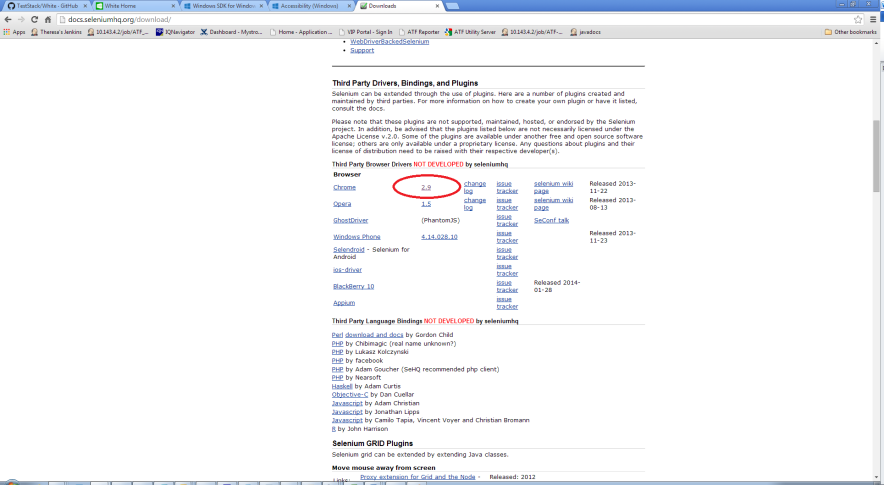
## Git Docs

Free Git book download: <http://git-scm.com/book/en/v2>

## Selenium & Webdrivers

Latest Selenium & Webdrivers <http://docs.seleniumhq.org/download/>





## GnuWin wGet

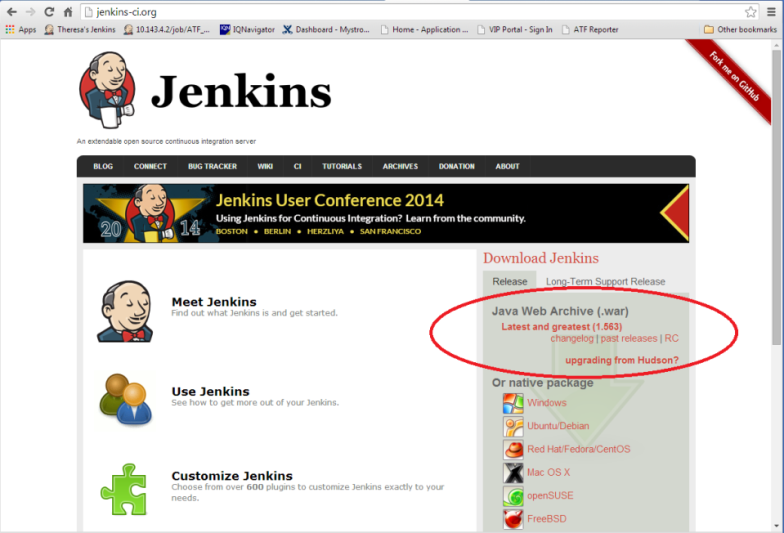
**Optional, check with your manager** before loading.

Latest GnuWin wGet <http://gnuwin32.sourceforge.net/packages/wget.htm>

## Jenkins, Optional

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

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

****

# Install Instructions

## Download and Install Java JDK

1. Run/Unzip the distribution archive, i.e. jdk-8u25-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\_25 *(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-8u25-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\_25.jdk/Contents/Home

export PATH=$JAVA\_HOME:$PATH

## Download and Install 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, 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:\maven\apache-maven-3.x.x

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

## Download & Install Perforce

Follow the P4V installation instructions from the Perforce site. Note that you must have your own login to Perforce and have it configured to continue with your environment setup. Ask your manager for instructions if you don’t have a Perforce account.

## Install Eclipse

### Eclipse IDE for Java Developers

1. When you install, extract it to C:\ *(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:\eclipse

1. For Windows PC, if you can’t locate eclipse.ini, change your folder options: go to C:\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:\Program Files\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

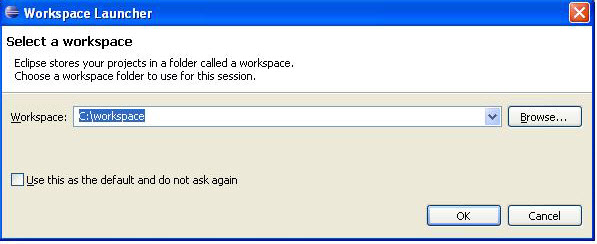
openFile

-vm

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

-vmargs -Xms300m -Xmx590m

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



### Install the Perforce Eclipse Plug-In

1. Be sure you already have Perforce P4V installed and have your own user ID and password and have used it to connect to the Perforce server to verify your login and set up your Perforce workspace.
2. Open Eclipse.
3. From Eclipse, *Help > Install New Software*,  
   In "*Work with:* " field enter the following based upon your version of eclipse:

 Eclipse 4.3: http://www.perforce.com/downloads/http/p4-eclipse/install/4.3

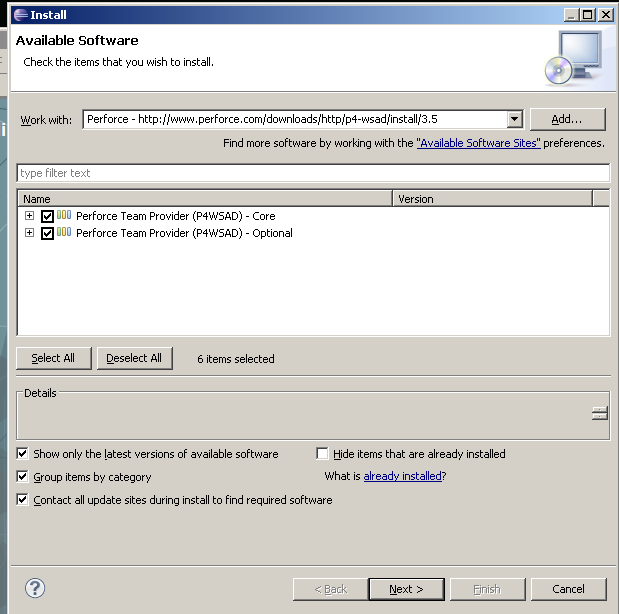
 Eclipse 4.2: http://www.perforce.com/downloads/http/p4-eclipse/install/4.2

 Eclipse 3.8: http://www.perforce.com/downloads/http/p4-eclipse/install/3.8

See more at: ["Perforce.com downloads step 3"](http://www.perforce.com/downloads/Perforce/20-User?qt-perforce_downloads_step_3=5%23qt-perforce_downloads_step_3%2321)

Add   
Name: Perforce

1. *Check* first check box (for Core) in the Name/Version fields. *(You don’t need Optional.)*



1. *Click “Next”* twice.
2. Accept the terms of the license agreement and *select “Finish”.*
3. Fill in fields (open Perforce P4V to figure out your login ID & the server IP). *If you don’t see this option, you can do it in a later step.*
4. Restart Eclipse if prompted.

### 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.

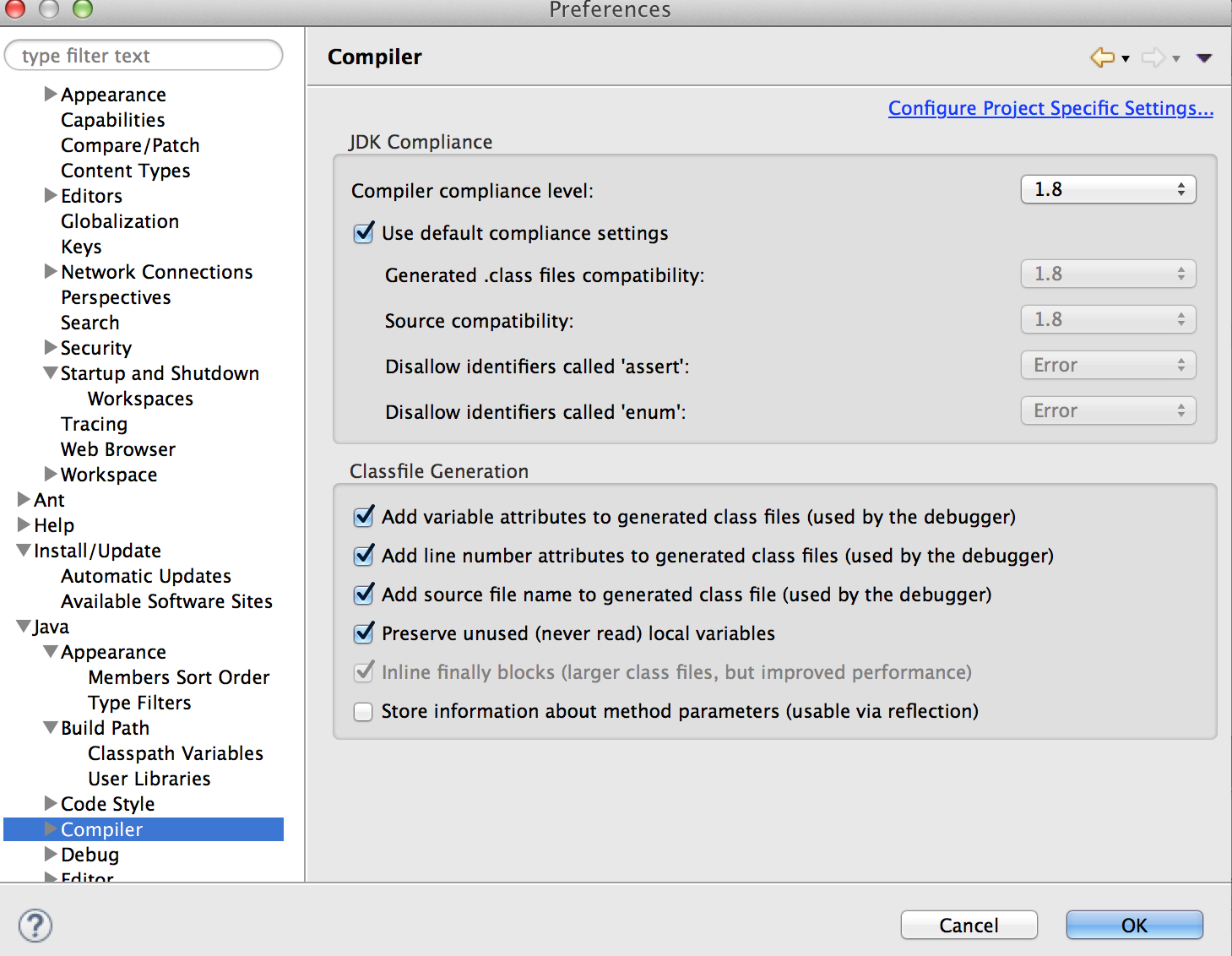
### Install the Maven Eclipse m2eclipse Plugin

To install this plugin in the Eclipse IDE:

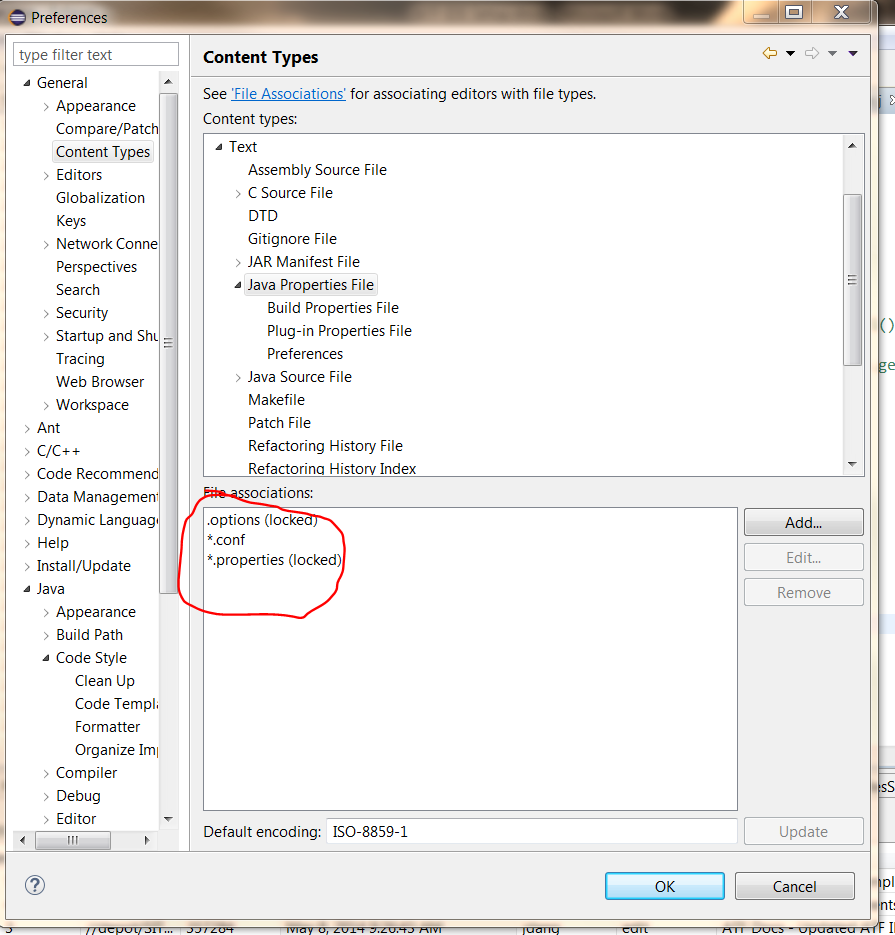
1. Open Eclipse.
2. Select *Help > Install New Software*. This should display the "Install" dialog.
3. Check to see if m2e is already installed with your Eclipse version. From the ‘Install’ Dialog, *click ‘What is already installed?’* link. *Expand* the ‘*Eclipse IDE for Java Developers’* and then *expand* 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.

## Configure Eclipse

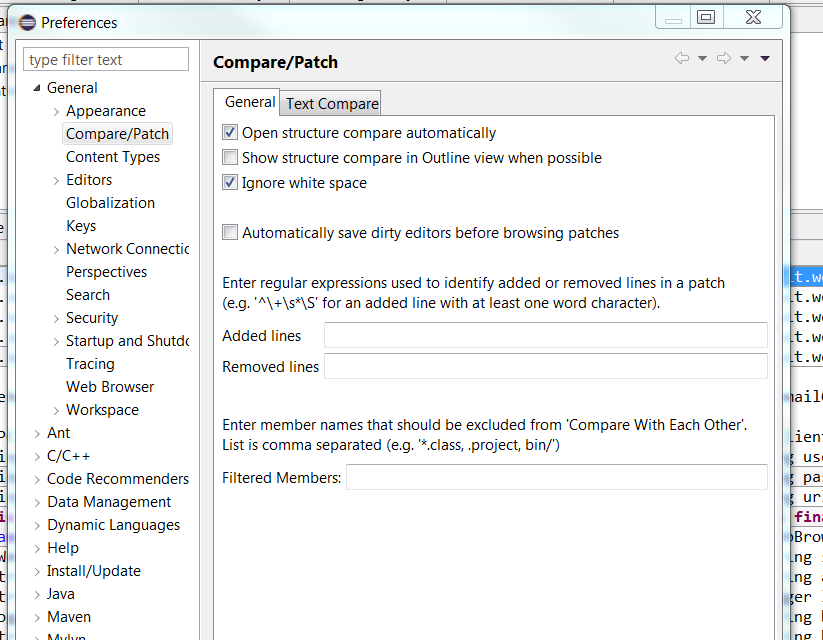
1. The root path in Perforce for the current ATF version is   
   **depot\SIT\atf-main**.   
   This will be referenced below as <atf-main>.
2. Open Perforce (P4V) to retrieve the following configuration files from the <atf-main>\documents directory into your local workspace:  
     
   ATF\_Eclipse\_Java\_CodeStyle\_CleanUp\_profile.xml  
   ATF\_Eclipse\_Java\_CodeStyle\_Formatter\_profile.xml  
   ATF\_Eclipse\_Java\_CodeStyle\_CodeTemplates\_profile.xml  
   SITServerDictionary **(check this one out)**.
3. In Eclipse, *Window > Preferences > Java > Code Style > Formatter > Import (*or for a Mac, *Eclipse > Preferences > Java > Code Style > Formatter > Import),* browse to your local workspace, such as   
   <atf-main>\documents\ATF\_Eclipse\_Java\_CodeStyle\_Formatter\_profile.xml, Open, OK.
4. In Eclipse, *Window > Preferences > Java > Code Style > Clean Up > Import*, browse to find  
   <atf-main>\documents\ATF\_Eclipse\_Java\_CodeStyle\_CleanUp\_profile.xml, Open, OK.
5. In Eclipse, *Window > Preferences > Java > Code Style > Code Templates > Import*, browse to find   
   <atf-main>\documents\ATF\_Eclipse\_Java\_CodeStyle\_CodeTemplates\_profile.xml, Open, OK.
6. In Eclipse, *Window > Preferences > Java > Compiler* verify that ‘Compiler compliance level’ is set to 1.8 (the same version as your Java JDK.)



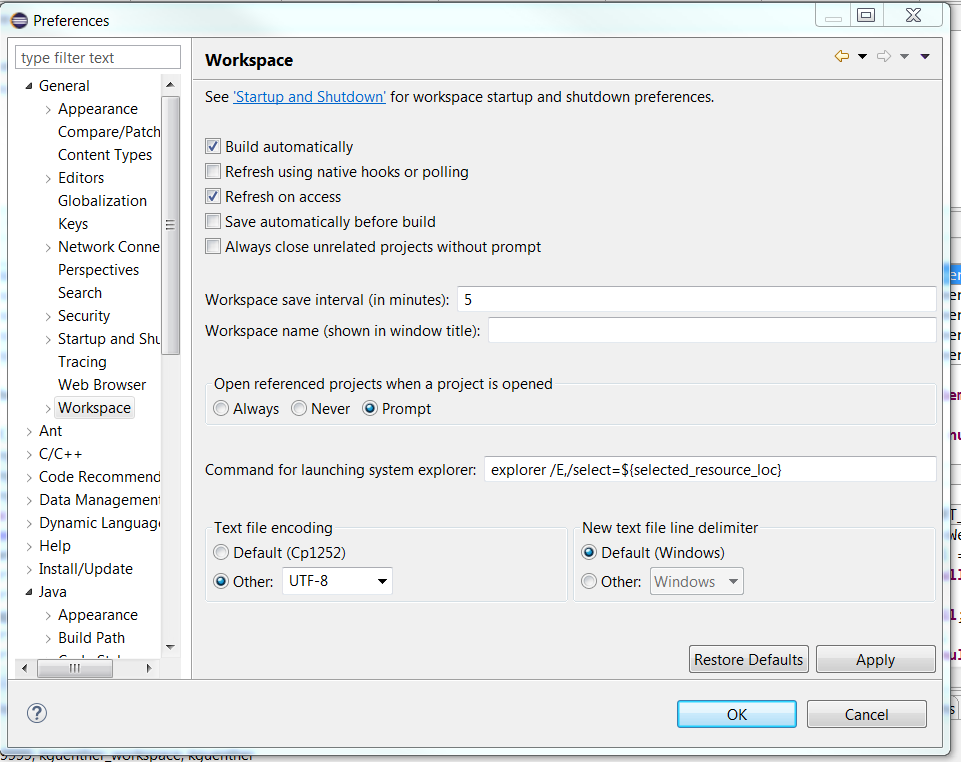
1. In Eclipse, *Window > Preferences > General > Content Types*, expand the *Text*. Browse to find Java Properties file and click *Add...*
   1. Enter in \*.conf



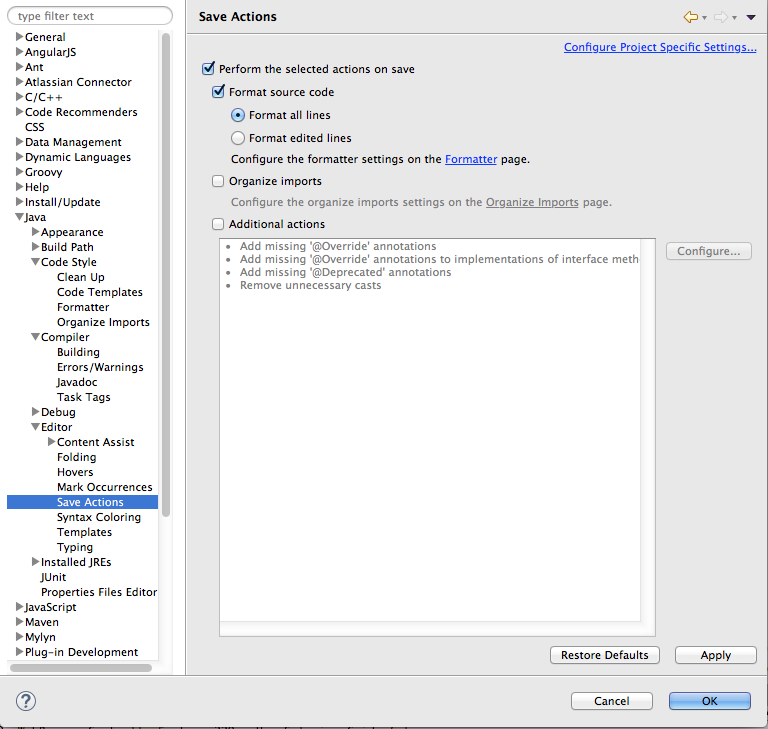
1. Also in the general section is Compare/Patch. Check “Ignore white space”. When comparing, this will ignore whitespaces that can make it harder to find real changes.



1. In Perforce, you should have checked out the <atf-main>\documents\SITServerDictionary so that it is writable. (Check out in Perforce by *right clicking* on the SITServerDictionary, *select ‘Check-out’*.) In Eclipse, *Window > Preferences*, expand *General >Editors > Text Editors > Spelling, User defined dictionary,* Browse to the <atf-main>\documents\SITServerDictionary location & select it, *Open*. When you are editing code, you can right-click on words & add them to the dictionary and then you can periodically reconcile & check-in your changes, being sure to keep the dictionary checked out after a check-in. You will probably have to merge changes to the dictionary so that you don't overwrite other coders’ additions to the dictionary.   
   On that same Spelling page, select *Encoding, Other, UTF-8, Apply, OK*.   
   *Tip: creating a changelist in Perforce labeled “Don’t Check In” helps to prevent accidentally submitting changes you want to keep local; if you move the SITServerDictionary to this changelist, you can still merge it periodically.*
2. In Eclipse, Window > Preferences > General >Workspace, Text file encoding, Other, UTF-8, Apply.



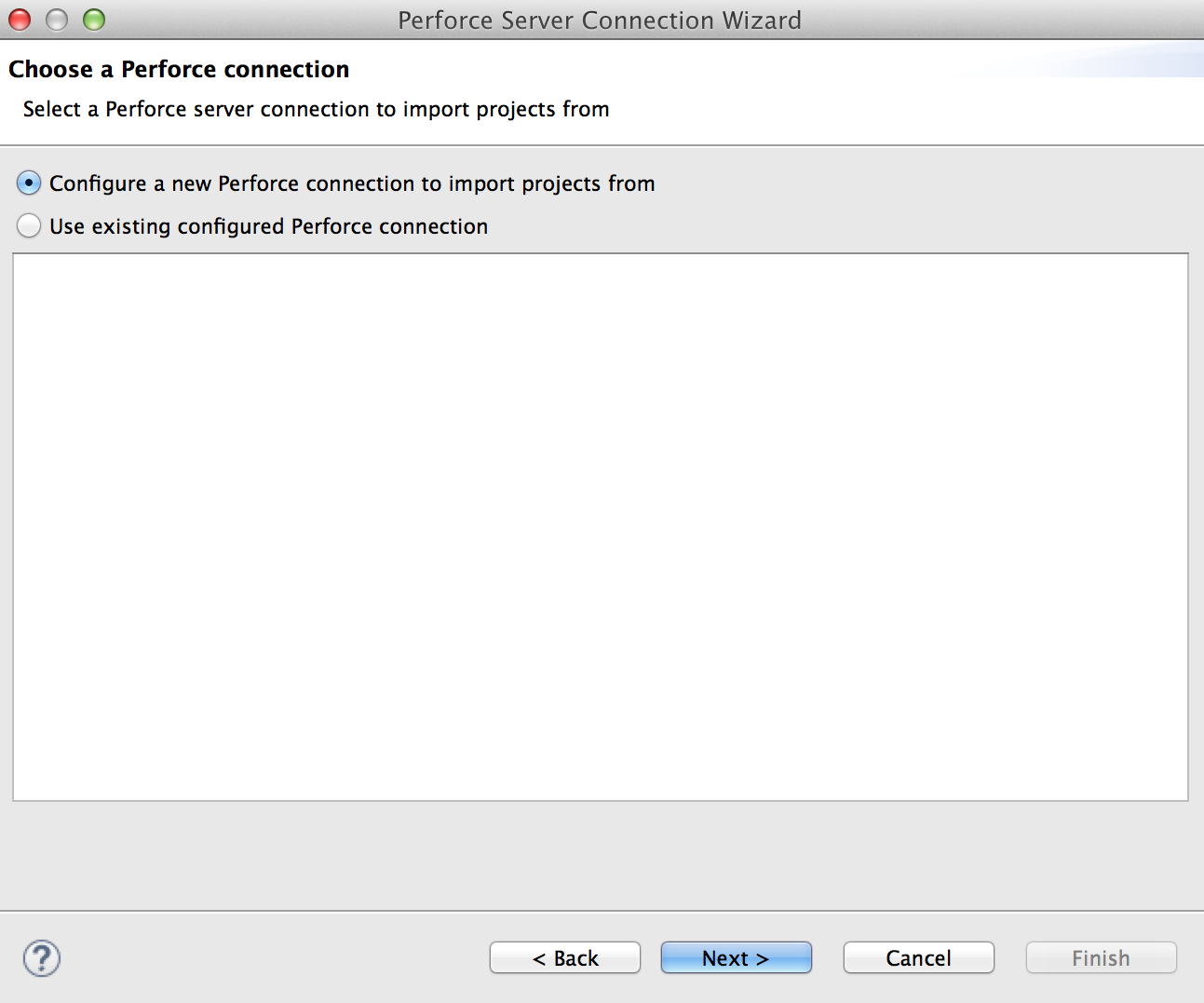
1. In Eclipse, Window > Preferences > Java > Editor > Save Actions. Check “Perform the selected actions on save”, Check “Format source code”, Select “Format all lines”, (Optional) Check “Organize Imports” and click Ok:



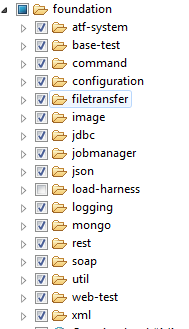
# Sync with ATF Projects in Perforce

## Add projects to Perforce

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 > Perforce > Projects from Perforce > Next*.

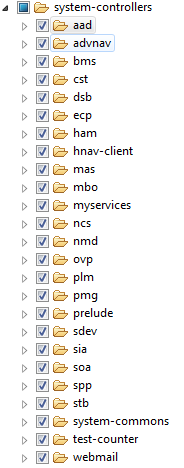
**

1. If you haven’t yet connected to Perforce from within Eclipse, you will need to select, *Configure a new Perforce connection > Next*
2. Fill in fields *(open Perforce P4V application to figure out your login ID & the server IP: open P4V > Connections > Open Connection for server IP:port, & user)*, add your password, then *Next*; select *Existing workspace*, then *Next*.
3. **If you have been told by your manager** to load the ATF Foundation, import the following projects from <atf-main>/foundation, and the projects under foundation, such as the following:



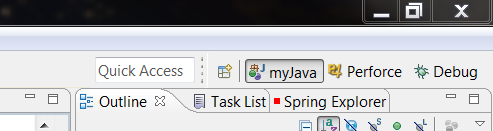
Etc. …

1. Import the projects that you have been told to load by your manager, from <atf-main>



Etc. …

1. Within Eclipse, Perforce has its own perspective. To add a link to it in the upper right hand corner like this…



Do: *Window > Open perspective > Other > Perforce* and click *ok*. This makes it a bit easier to perform most perforce actions from within Eclipse.

1. Add the location of the p4merge.exe app so that you can resolve change conflicts:
   1. *Window > preferences > Team > Perforce > External Tools*. The executable path for p4merge will be where Perforce is installed. *(It is not in your Eclipse workspace.)*
   2. If you can’t find p4merge.exe, you may need to open p4 and add it as a plug-in.

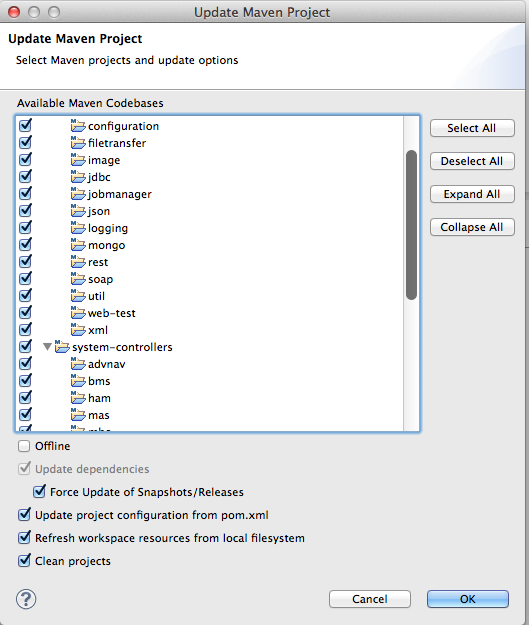
## Build Projects for First Time

In order to use maven goals, with the ATF, the source ATF projects must be built and deployed to your local maven repository so that your other projects can use them *(your system level tests)*. If you don’t have access to the Foundation projects, then you will just skip the foundation & follow these steps for the system-controller sets of projects that your manager has told you to load.

1. In Eclipse, select each project, starting with the ‘foundation’ set of projects.
2. *Right-click* on it and for Eclipse Juno release and newer, select *Configure > Convert to Maven project*. For older versions of Eclipse, select *Maven > Enable Dependency*.
3. Right-click on it again and select *Maven > Update Project.*
4. Check checkbox for *Force Update of Snapshots, Releases*.
5. Click *Run*
6. Note that if projects aren’t building this first build, you may need to run maven from the command line; go to a command prompt in the atf project directory (where the pom.xml is located) and run

mvn clean compile install -DskipTests

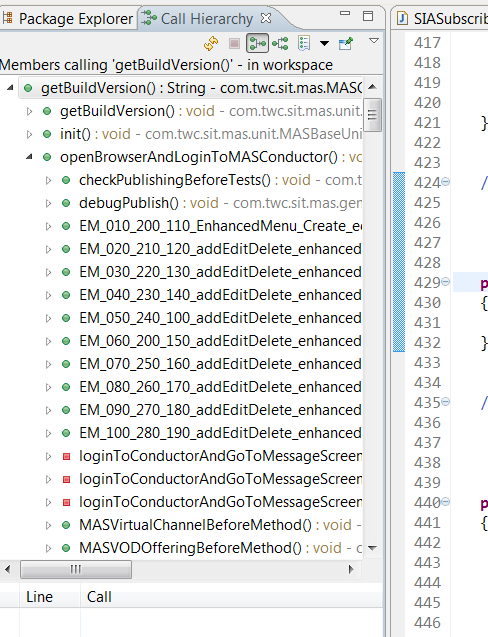
1. Repeat this build process for the system-controllers sets of projects.



# 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 Perforce actions in Eclipse: Go to *Window > Open perspective > other > Perforce*.
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. Perforce/Changelist tip: Create a changelist labeled “Don’t CheckIn” and move the docs you don’t want to submit (such as the SitServerDictionary) to it. This helps prevent accidentally submitting changes you want to keep local.
3. Look in *Eclipse > Help* for various tips, such as Keyboard shortcuts.
4. 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:\eclipse\eclipse.exe -vm "C:\Program Files\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:\Program Files\Java\jdk1.8.0\_25\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:\eclipse\eclipse.exe -vm "C:\Program Files\Java\jdk1.8.0\_xx\bin\javaw.exe"

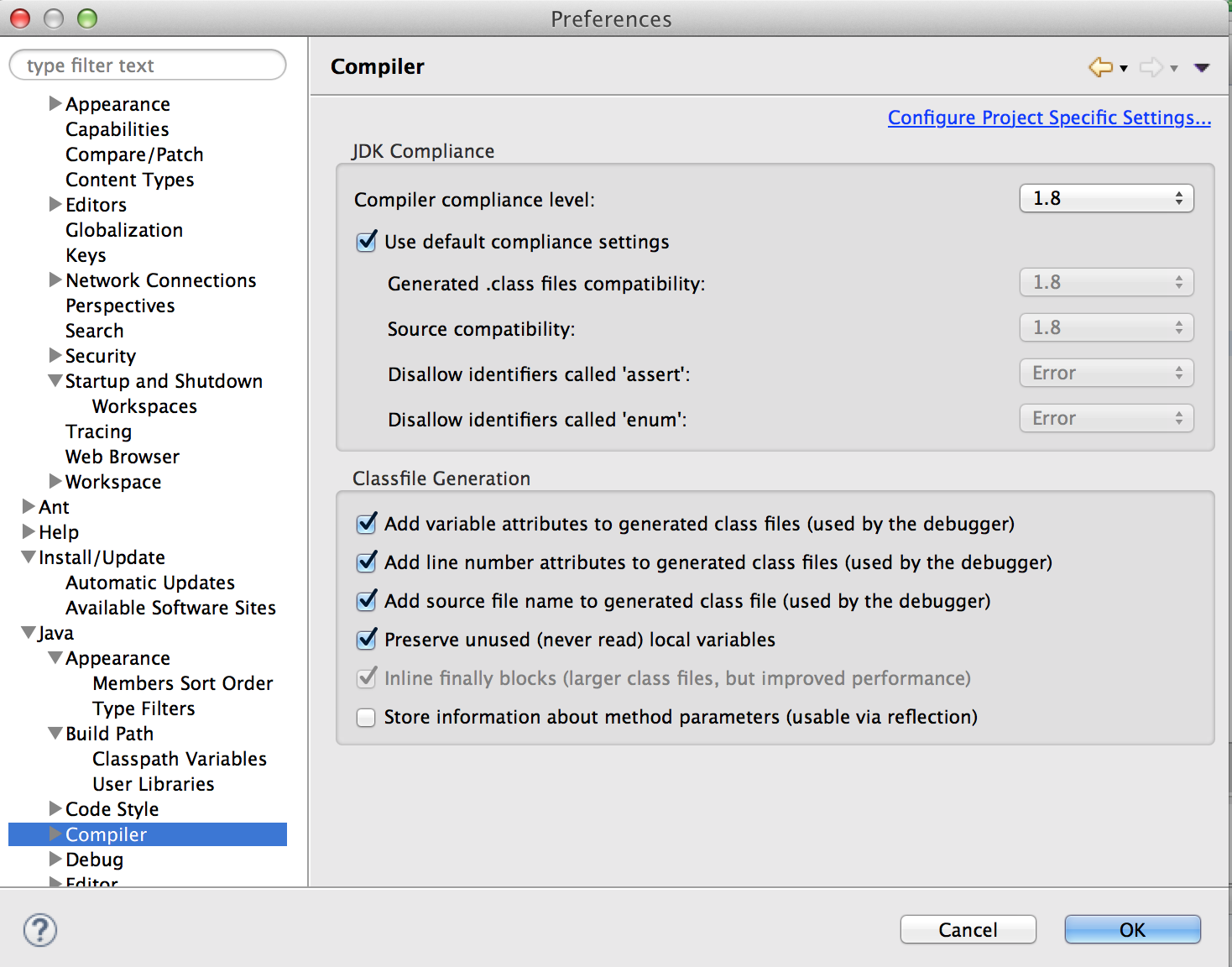
#### Solution 3:

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

# 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

### 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