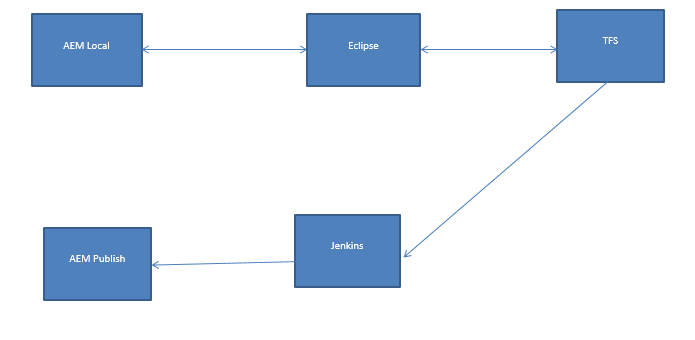
Intranet AEM Development Process

Below is a diagram of the AEM development flow.



Eclipse is used to download files from TFS and build/publish to a local AEM instance.

Eclipse is used to upload edited/new files into TFS.

Jenkins is used to execute a build using the project files in TFS and publish to the AEM publish instance.

The following instructions describe how to setup a local development environment to follow this process.

# The initial Setup

1. Download Eclipse (Luna) from shared Drive and put in local **C:/** drive and AEM jar file with license file in local path **C:/aem**:

[\\Us.bank-dns.com\ptlmetro\or14p5\WEBDEV\users\ECM\AEM\eclipse](file:///\\Us.bank-dns.com\ptlmetro\or14p5\WEBDEV\users\ECM\AEM\eclipse)

AEM 6.3 Jar: [\\Us.bank-dns.com\ptlmetro\or14p5\WEBDEV\users\ECM\AEM\AEM 6.3](file:///\\Us.bank-dns.com\ptlmetro\or14p5\WEBDEV\users\ECM\AEM\AEM%206.3)

1. Plugins needed : AEM and TFS Plugins

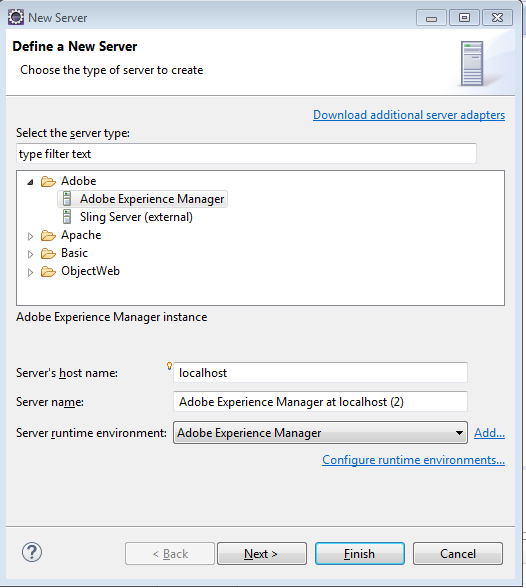
Above set up of eclipse has both the plugins and their respective dependencies already included.

AEM plugin manual [install](https://docs.adobe.com/docs/en/aem/6-3/develop/dev-tools/aem-eclipse.html?wcmmode=disabled)

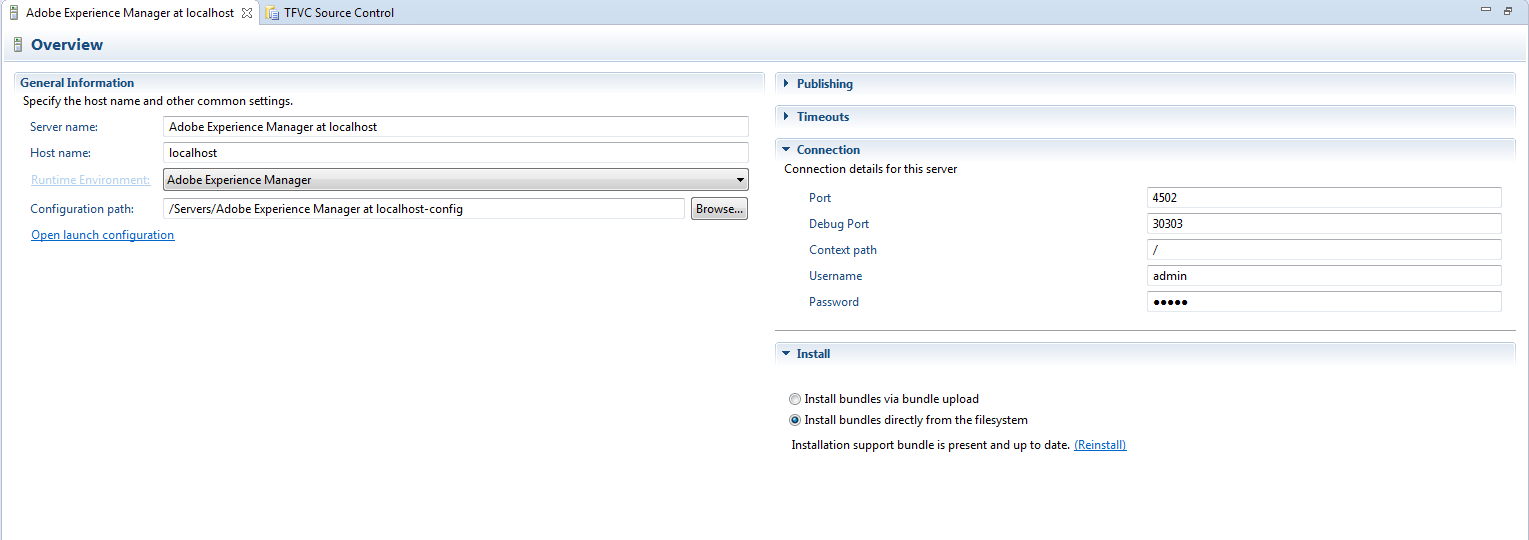
TFS plugin manual [install](https://msdn.microsoft.com/en-us/library/hh301122(v=vs.120).aspx)

1. Configure Servers to localhost and connect to Team Project in TFS Explorer.

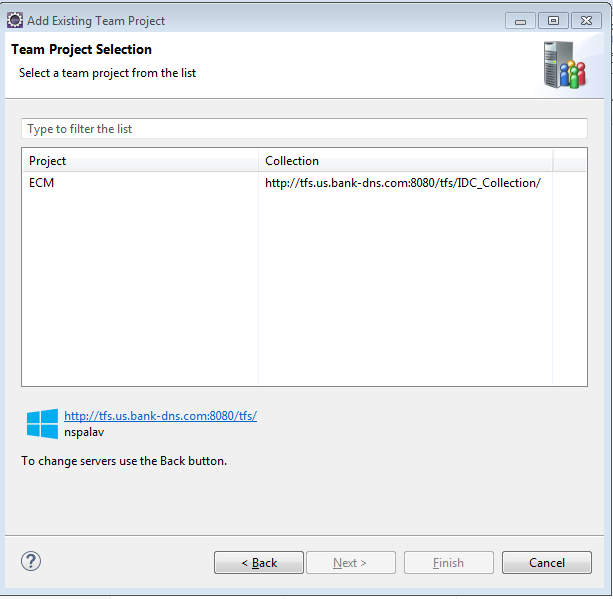
In the Servers View, right click and add a new server.



Click Finish. Double Click the Server and configure as shown below.



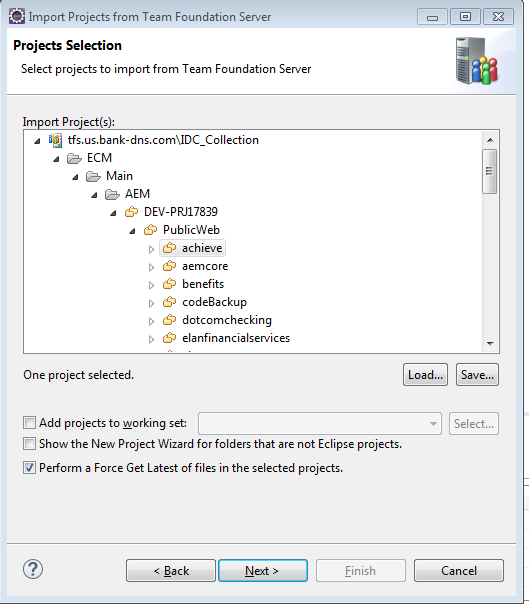
In the Team Explorer View, Click on connect to Team Projects and configure as shown below:



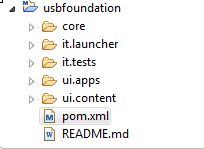
1. Import Project from TFS.

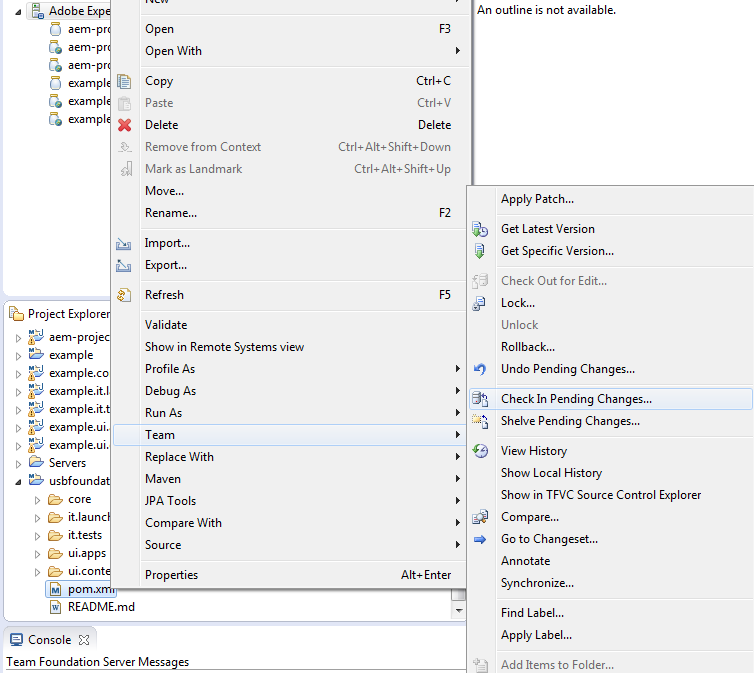
In AEM View , click on File 🡪 Import 🡪Team🡪 Team Foundation Server🡪 Next

Select the Team Project. The Import Project from Team Foundation Server dialog box appears.

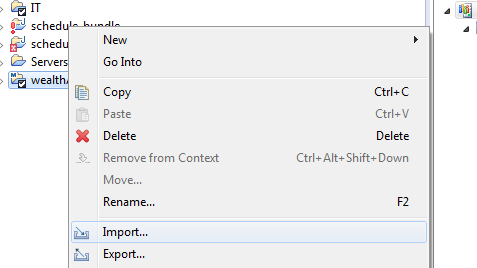


Click Finish. It should show up in Project Explorer as shown below :

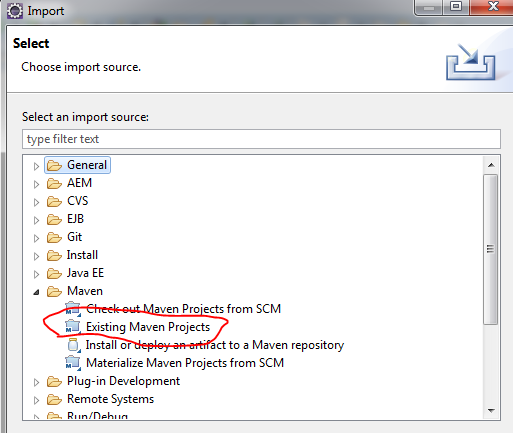




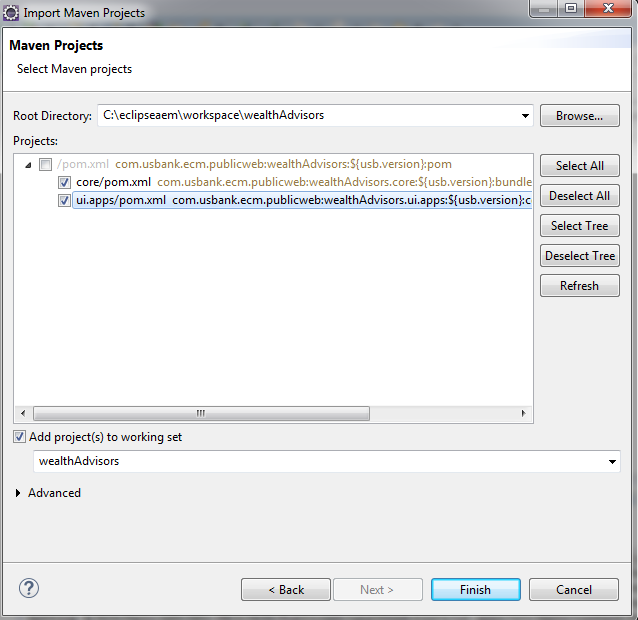
On newly created Eclipse project, right click and select Import:



Select ‘Existing Maven Projects’:

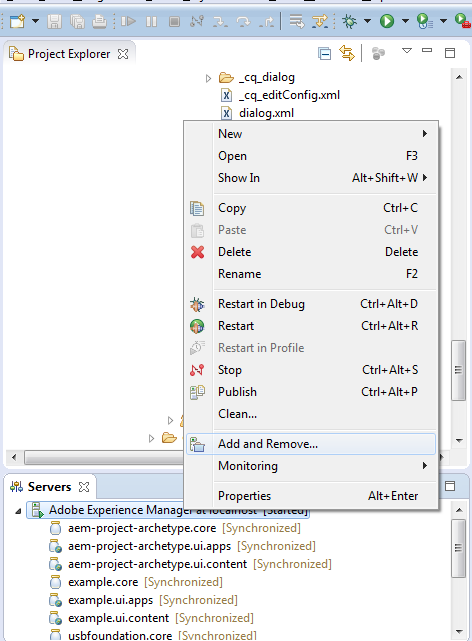


Select all ‘sub’ projects to import and click ‘Finish’

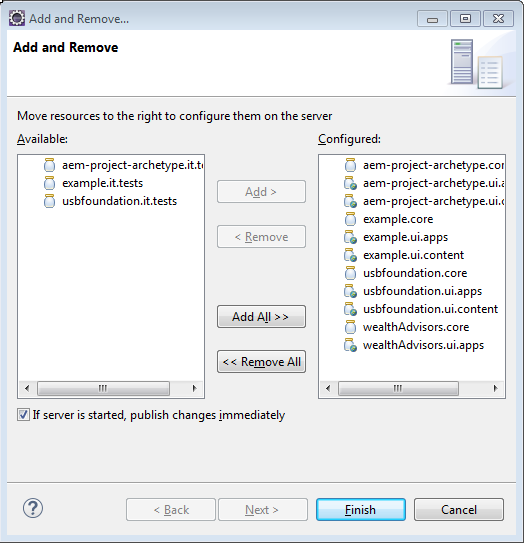


Right click the server that has been configured to add the project to the server :

Right click 🡪 Add and Remove



From the list of available projects, add to the configured list. Ensure the checkbox is checked for changes to get published immediately. Click Finish.



Changes will automatically be synced to localhost.

# Configure maven build command

1. Mvn command doesn’t run from behind the firewall. Updated settings.xml with proxy settings.

<proxies>

<proxy>

<active>true</active>

<protocol>http</protocol>

<host>web-proxymain.us.bank-dns.com</host>

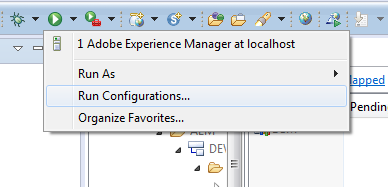
<port>3128</port>

</proxy>

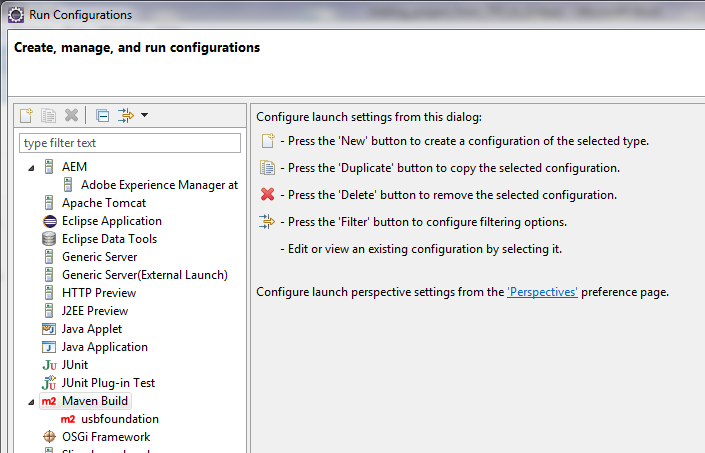
</proxies>

</settings>

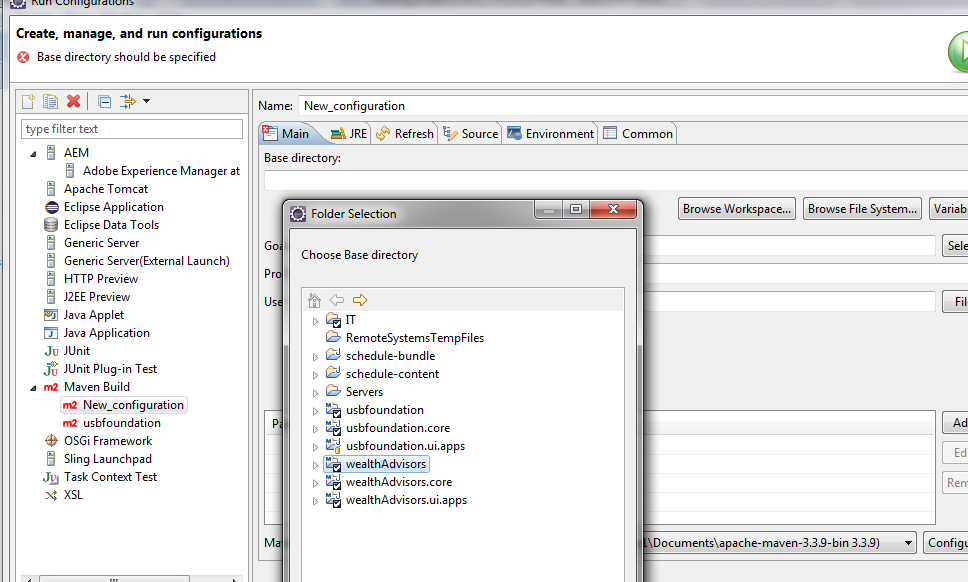
1. Go to Run Configurations:



1. Click on Maven Build:



1. Press new button to create a new build config



Base directory – click Browse Workspace and select main project

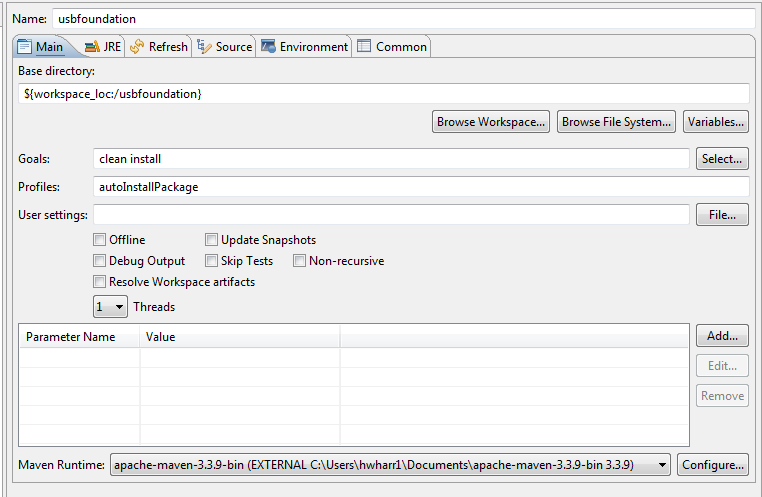
Under Goals add “clean install”

Project profiles – “autoInstallPackage”

Select which Maven runtime to use – click Configure to add an entry for your installed version



Build config should look like:

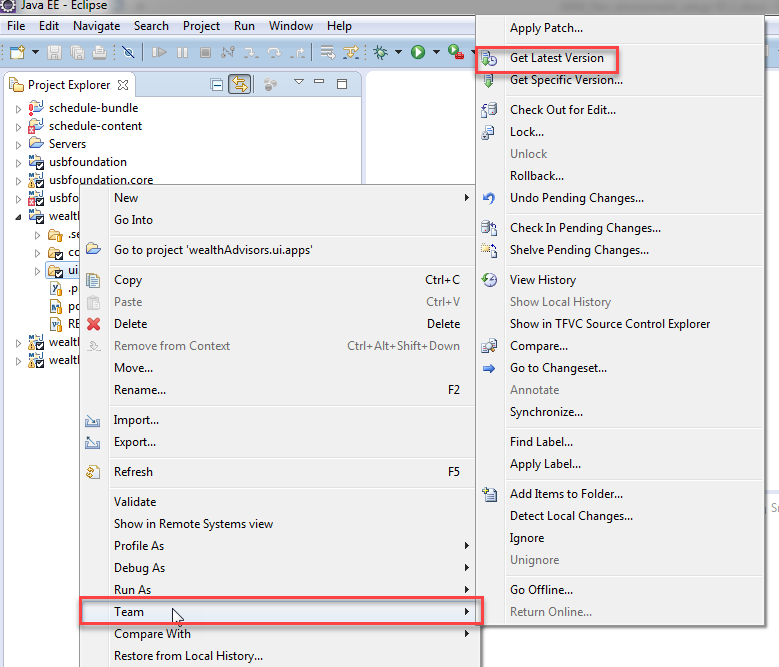


At this point you’ll want to make any updates/changes/additions in the project in Eclipse.

Make your code updates by checking out the latest file, performing the edit and checking it back in.

# Get Latest changes from TFS

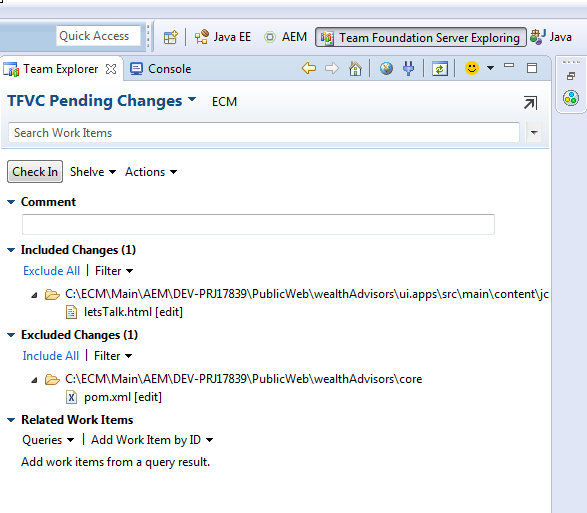
Right click a project, a folder or an individual file, select Team > Get Latest Version



# Move changes into TFS and execute the build/install process for publish instance

1. Check in your component into TFS , see below steps :

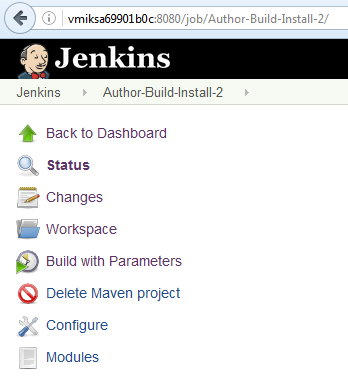
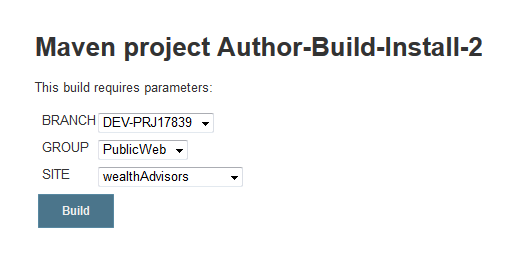
Ensure the **Included Changes** has the file you wish to check in. Add appropriate comments and Click “Check In”. It will display “Changeset #### successfully checked in.”



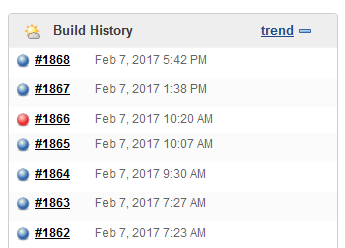
1. Build in DEV: Once done, go to Jenkins Dev and perform a build to move the changes to DEV AEM.

JENKINS DEV : <http://vmiksa69901b0c:8080/job/Author-Build-Install-2/>

Click on “Build with Parameters” and select “WealthAdvisors” as the site & build.

Track the build to ensure it was a success and check the console output by clicking on the build #.



The component should show up in crx/de and the package timestamp updated in package manager console. See snapshot below:

AEM DEV: <http://vmiksa69901b0c:4502/crx/de>

