Developer Setup Guide

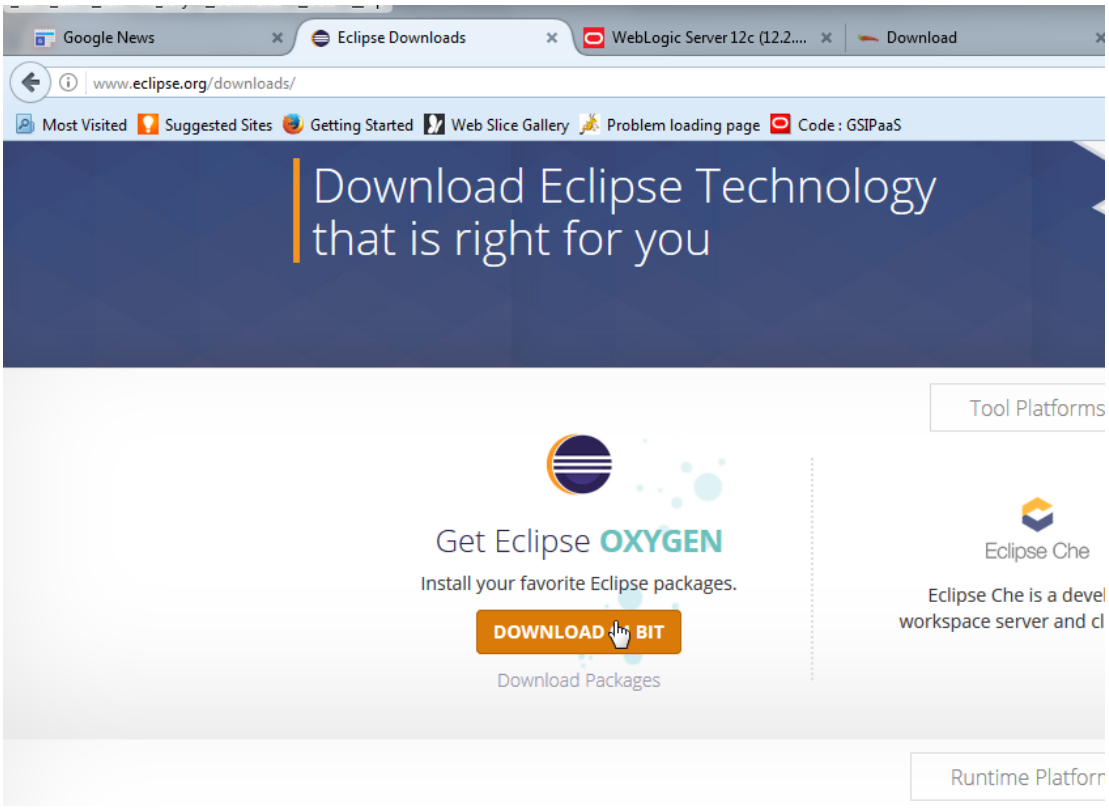
Installation

Eclipse

Download Eclipse IDE from the below location. At the time of writing this guide, NEON is the latest GA version.

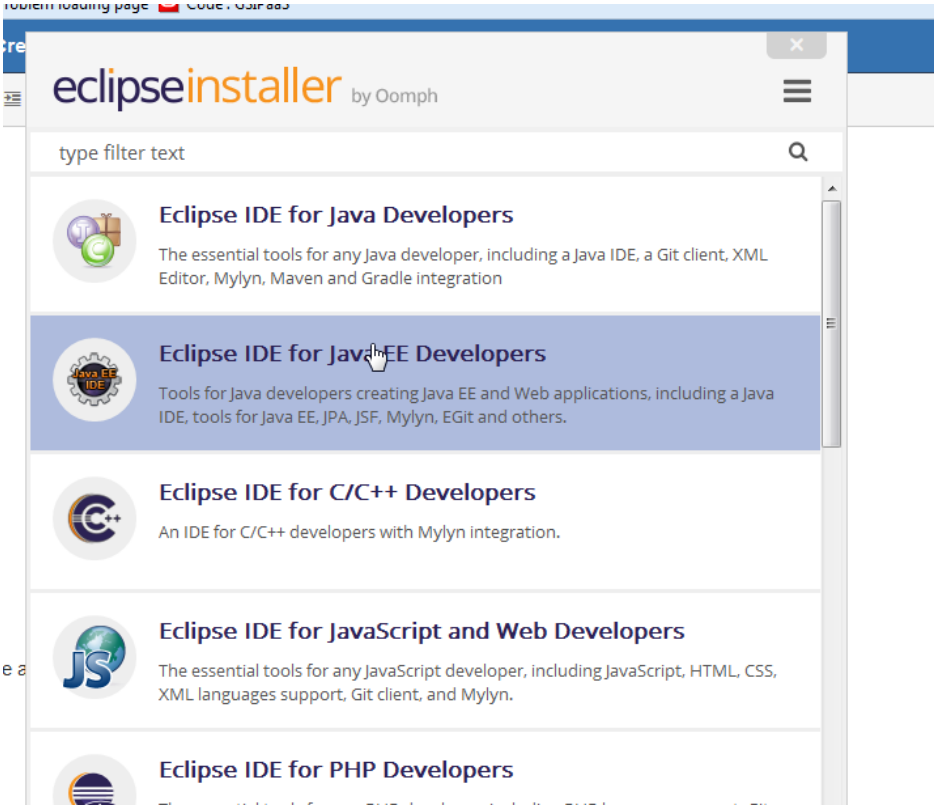
http://www.eclipse.org/downloads/

Extract the IDE (download as a ZIP/DMG depending on platform) to a location of choice.

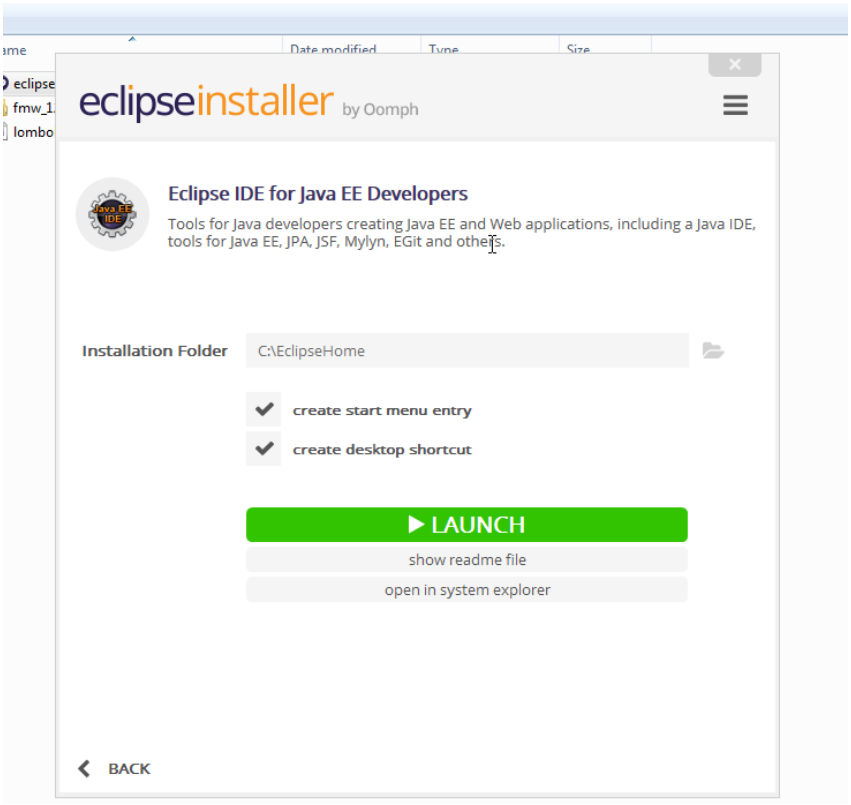


Select the downloaded executable file (in Windows). In Windows case and this version, it is eclipse-inst-win64.exe

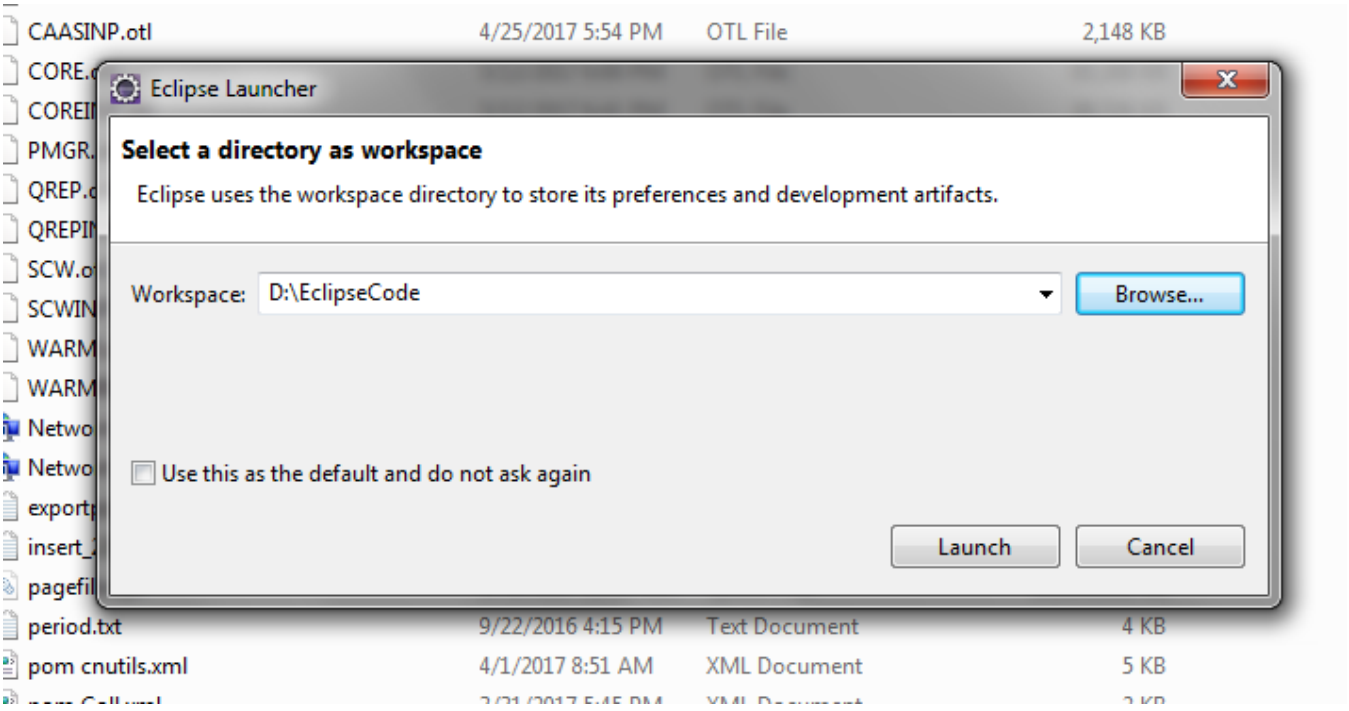
Select Eclipse IDE for Java EE Developers and put in the folder to install. Accept the License terms to start installaiton



Once completed, the following screen shows:



Launch and select your workspace in beginning



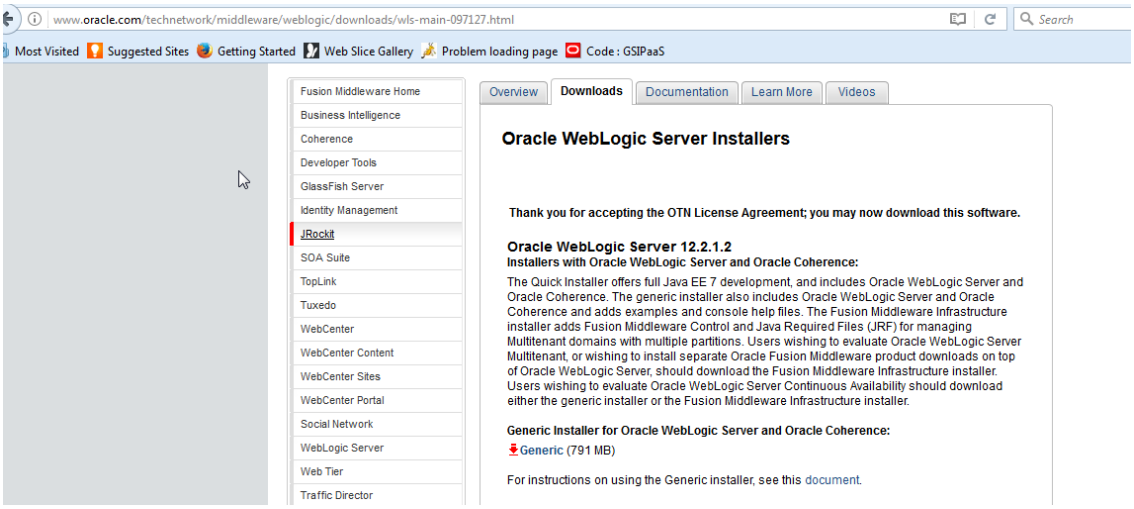
Weblogic Server

Download Weblogic Server from the below location. At the time of writing this guide, 12.2.1.2 is the latest GA version.

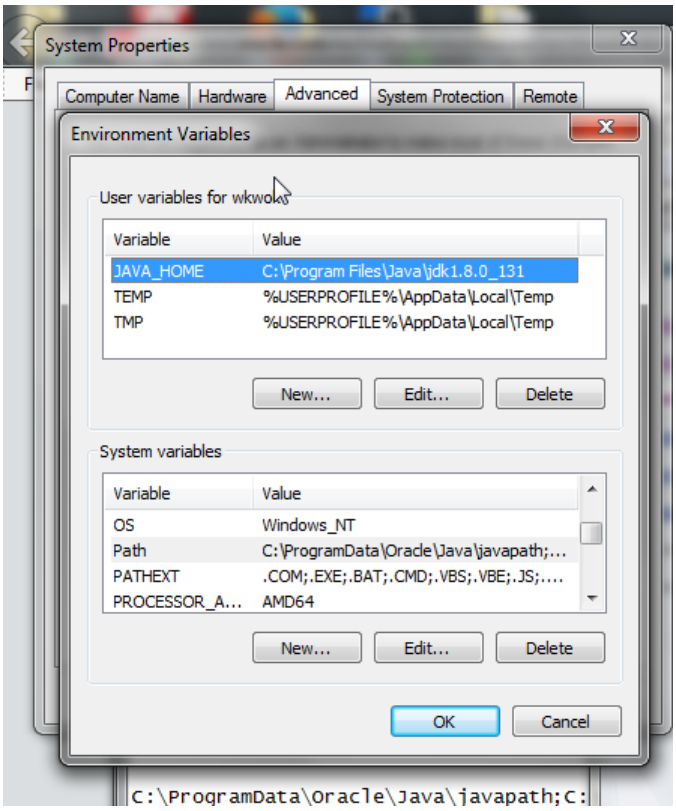
http://www.oracle.com/technetwork/middleware/weblogic/downloads/wls-main-097127.html

Run the installer accepting defaults for most part and create a base domain as a part of the setup process. Note the user/pwd provided during installation as this will be used later to log into the Weblogic Console for configuration and administration. Also note the Console URL & Domain folder location at the end of the installation.

Download weblogic from the link above. The latest version for Windows is 12.2.1.2

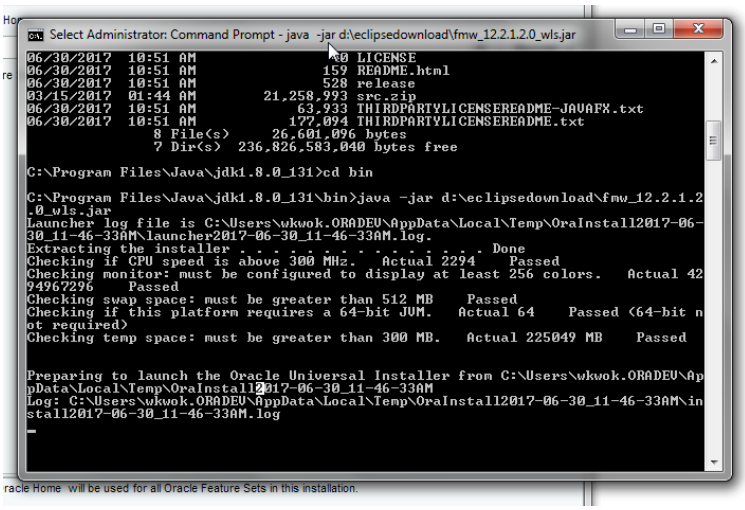


Ensure you have the JDK with JAVA_HOME set in your local PC.In this case, we used latest one JDK1.8.0_131.

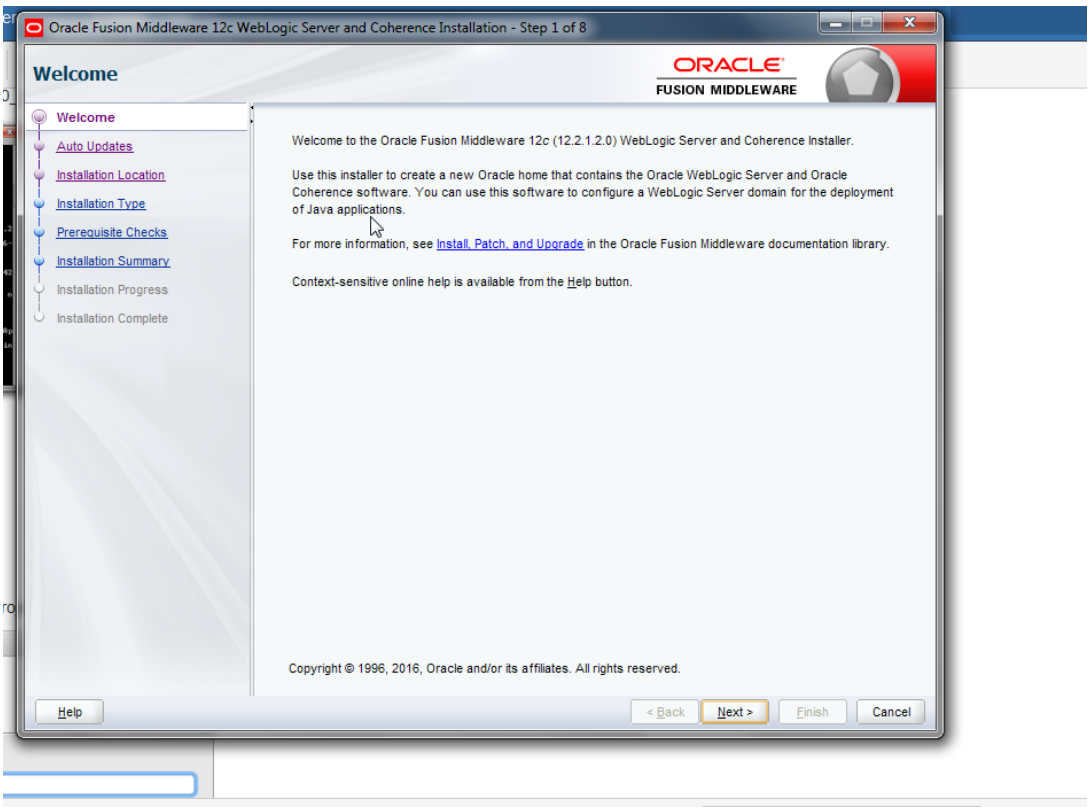


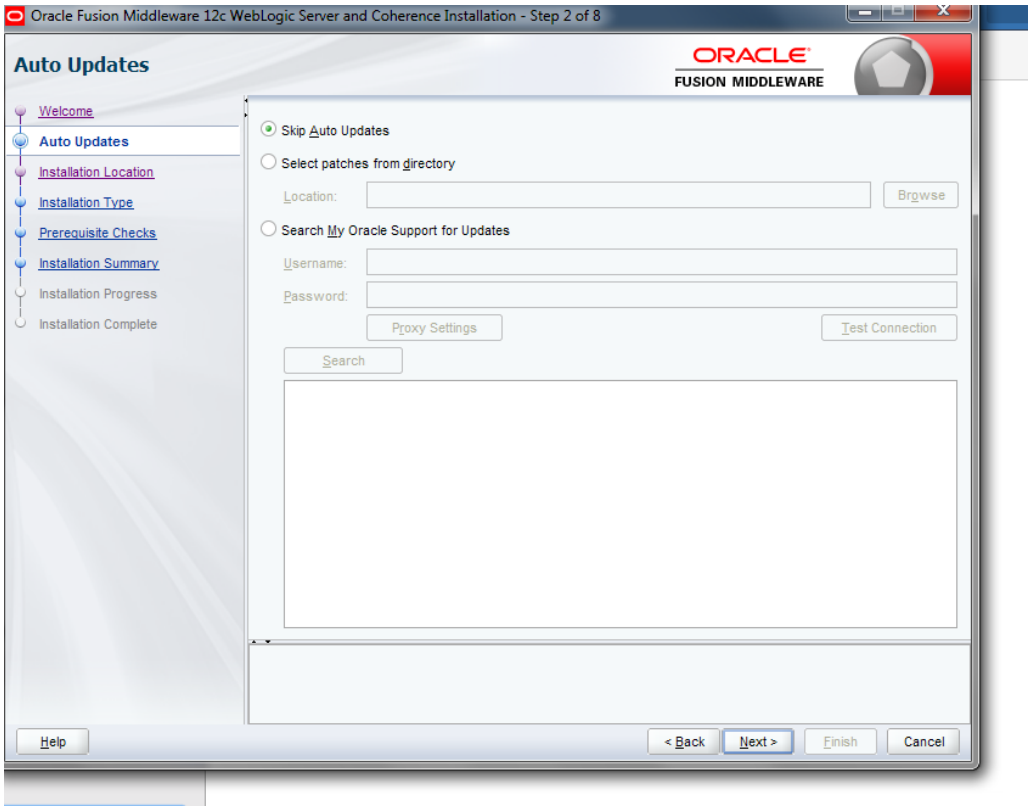
Run command prompt as administrator with command below:

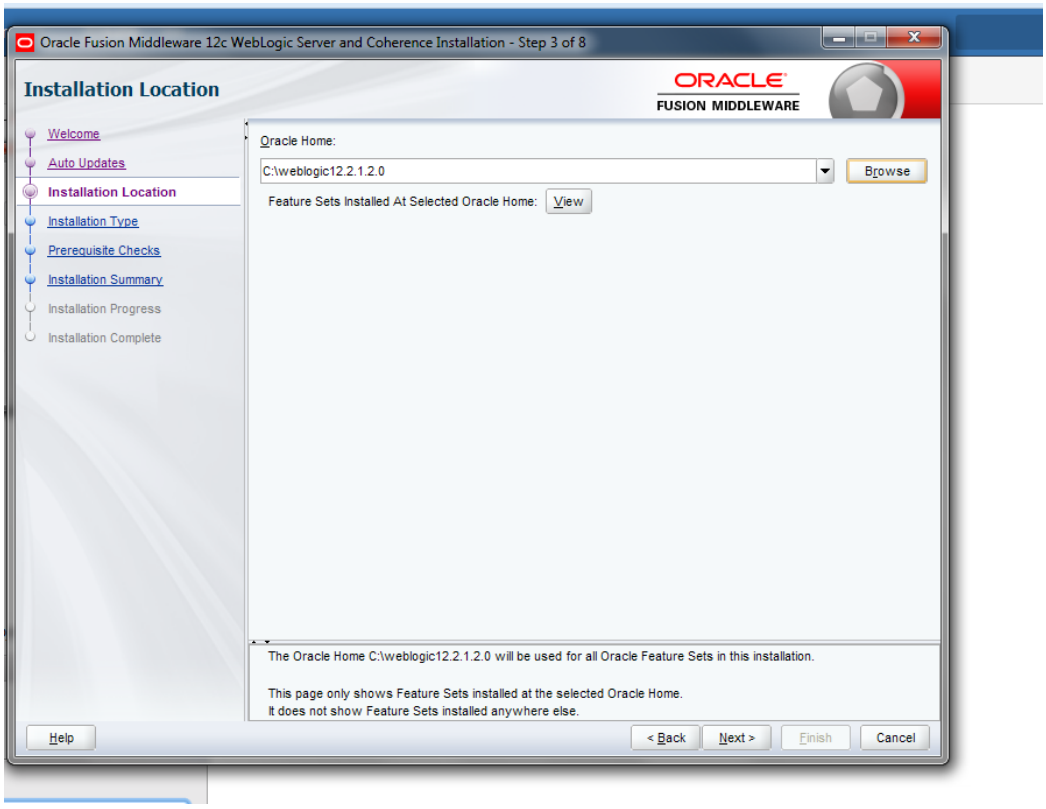
java -jar <drive letter>:\eclipsedownload\fmw_12.2.1.2.0_wls.jar

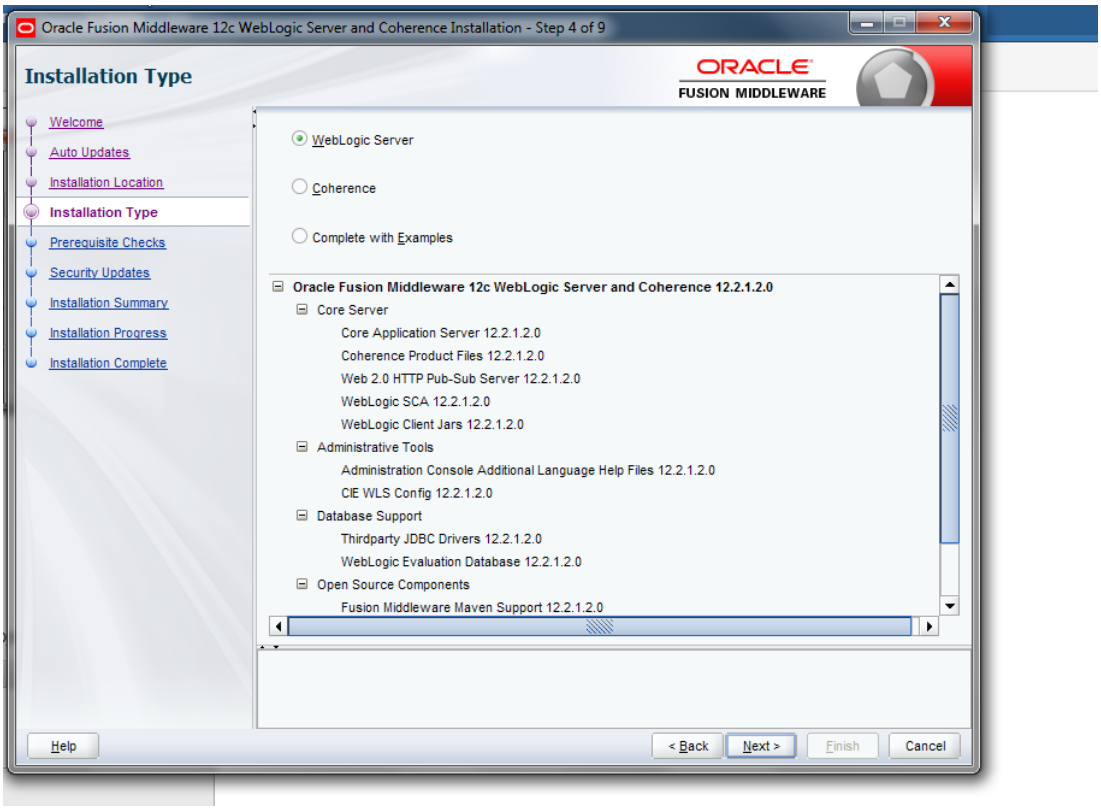


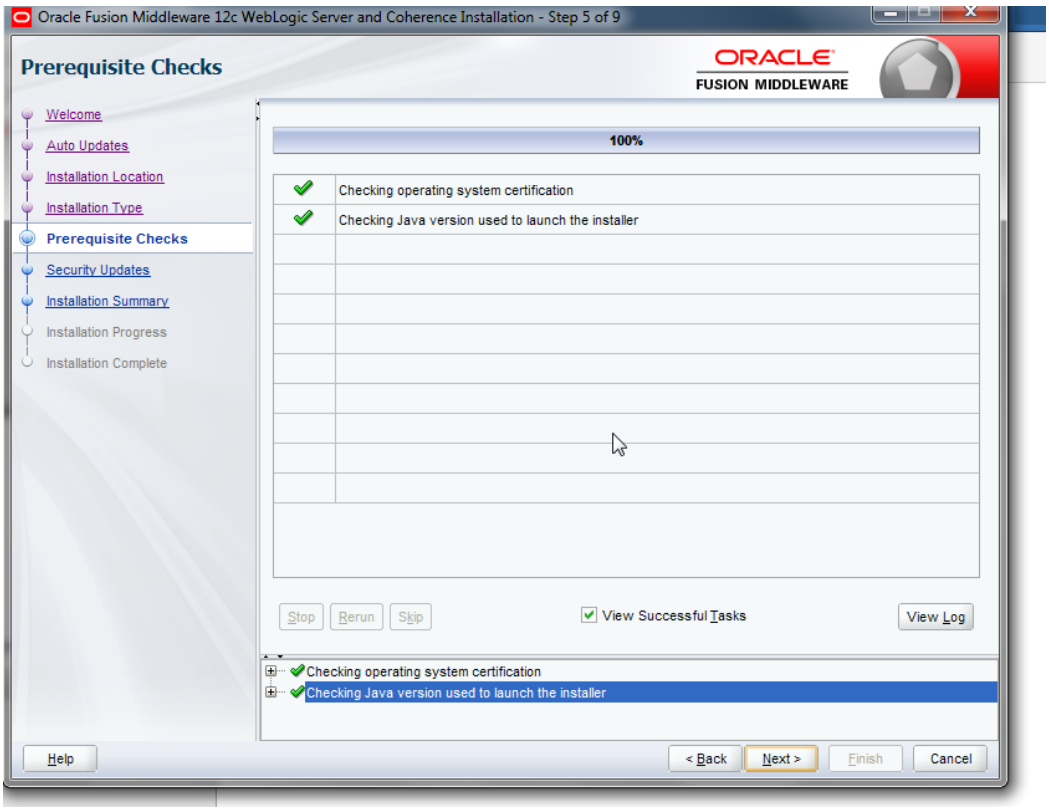
Welcome screen shown and go thru each step by clicking Next button

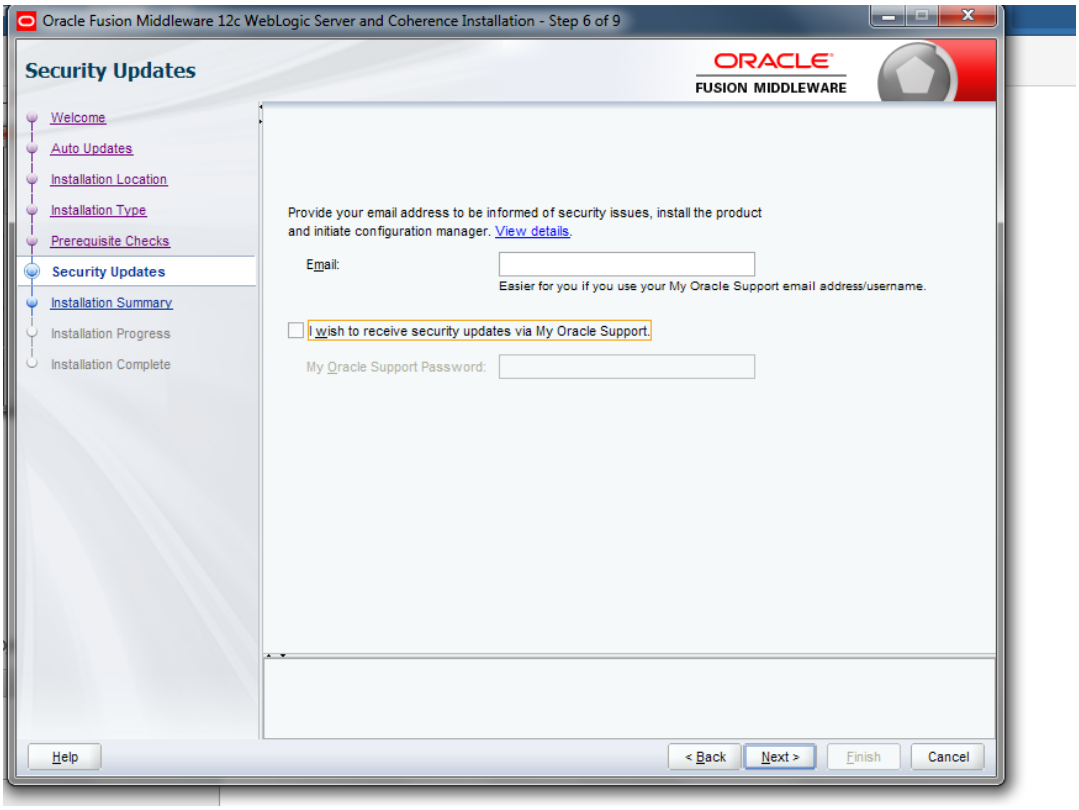


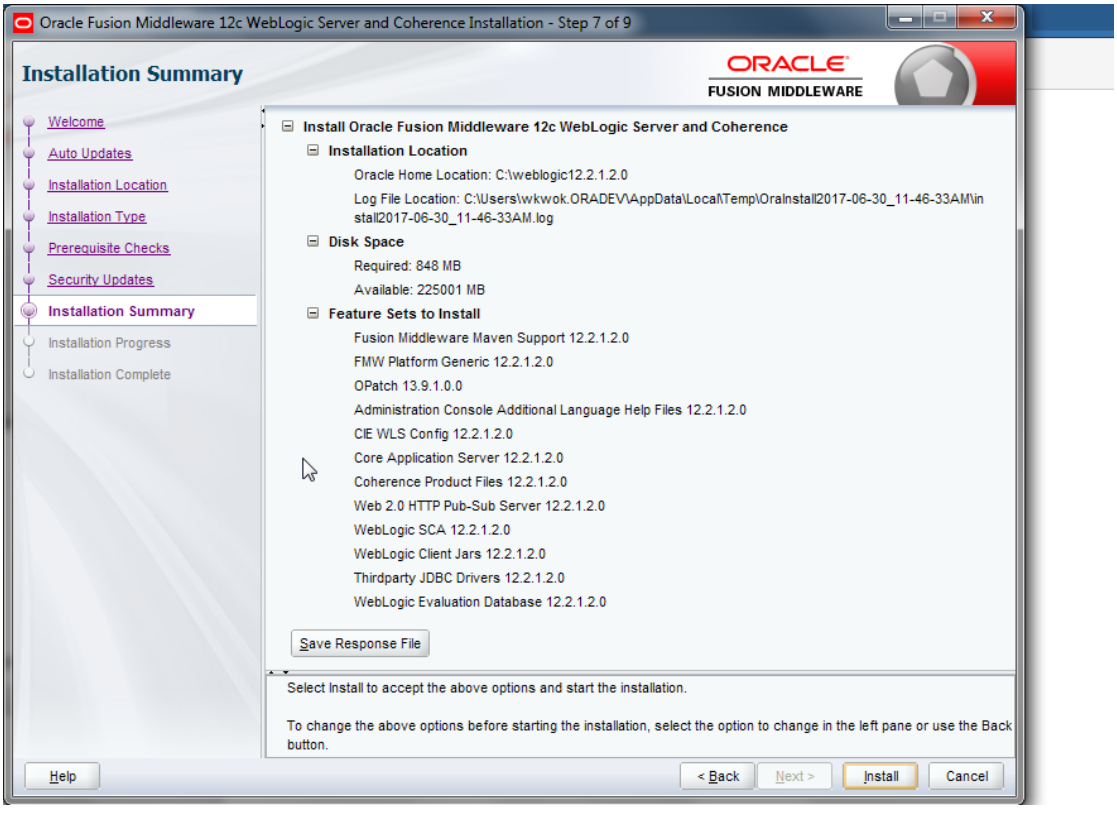

Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 2 of 8
Auto Updates
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Installation Summary
Installation Progress
Installation Complete
Skip Auto Updates
Select patches from directory
Location:
Browse
Search My Oracle Support for Updates
Username:
Password:
Proxy Settings
Test Connection
Search
Help
< Back
Next >
Finish
Cancel

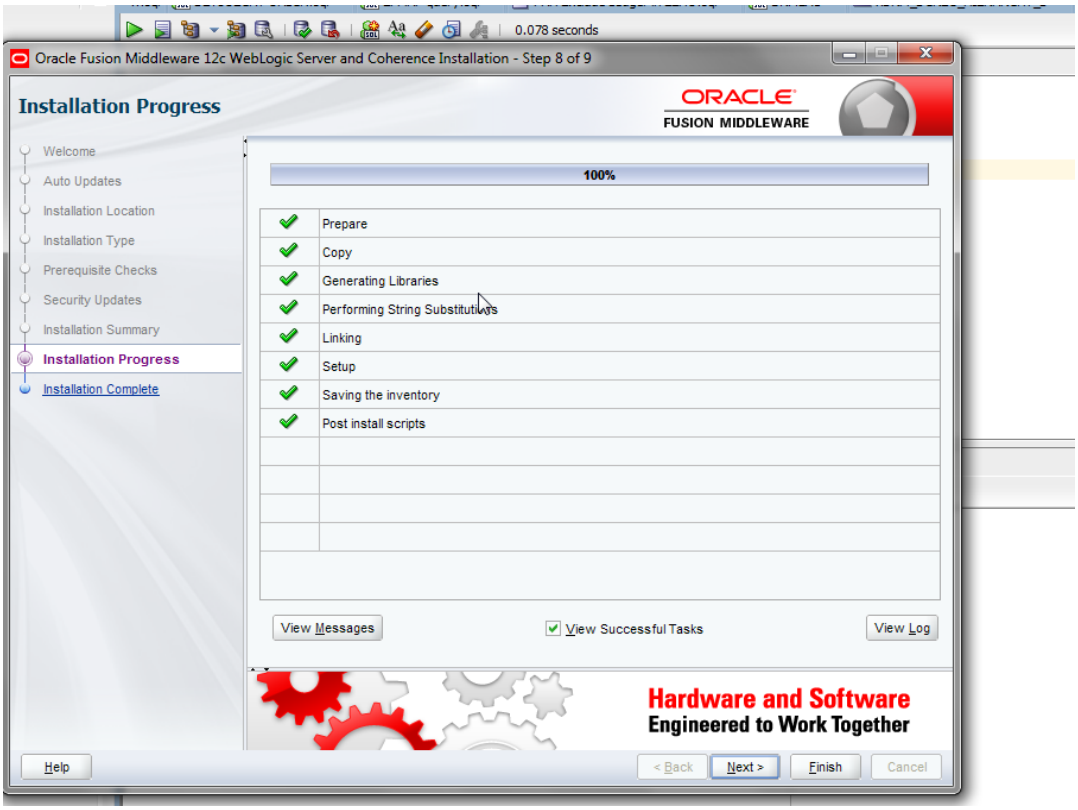

Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 3 of 8
Installation Location
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Installation Summary
Installation Progress
Installation Complete
Oracle Home:
C:\weblogic12.2.1.2.0
Browse
Feature Sets Installed At Selected Oracle Home:
View
The Oracle Home C:\weblogic12.2.1.2.0 will be used for all Oracle Feature Sets in this installation.
This page only shows Feature Sets installed at the selected Oracle Home.
It does not show Feature Sets installed anywhere else.
Help
< Back
Next >
Finish
Cancel

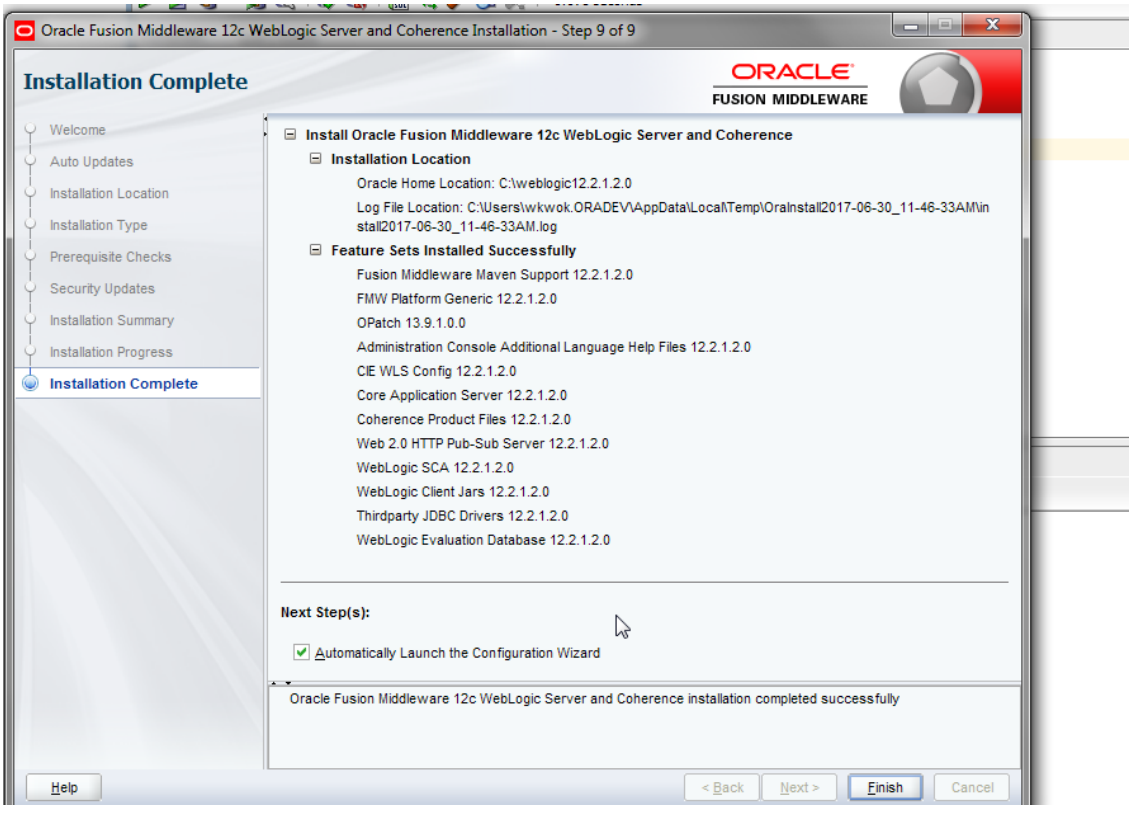

Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 4 of 9
Installation Type
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Security Updates
Installation Summary
Installation Progress
Installation Complete
WebLogic Server
Coherence
Complete with Examples
Oracle Fusion Middleware 12c WebLogic Server and Coherence 12.2.1.2.0
Core Server
Core Application Server 12.2.1.2.0
Coherence Product Files 12.2.1.2.0
Web 2.0 HTTP Pub-Sub Server 12.2.1.2.0
WebLogic SCA 12.2.1.2.0
WebLogic Client Jars 12.2.1.2.0
Administrative Tools
Administration Console Additional Language Help Files 12.2.1.2.0
CIE WLS Config 12.2.1.2.0
Database Support
Thirdparty JDBC Drivers 12.2.1.2.0
WebLogic Evaluation Database 12.2.1.2.0
Open Source Components
Fusion Middleware Maven Support 12.2.1.2.0
Help
< Back
Next >
Finish
Cancel


Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 5 of 9
Prerequisite Checks
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Security Updates
Installation Summary
Installation Progress
Installation Complete
100%
Checking operating system certification
Checking Java version used to launch the installer
Stop
Rerun
Skip
View Successful Tasks
View Log
Checking operating system certification
Checking Java version used to launch the installer
Help
< Back
Next >
Finish
Cancel

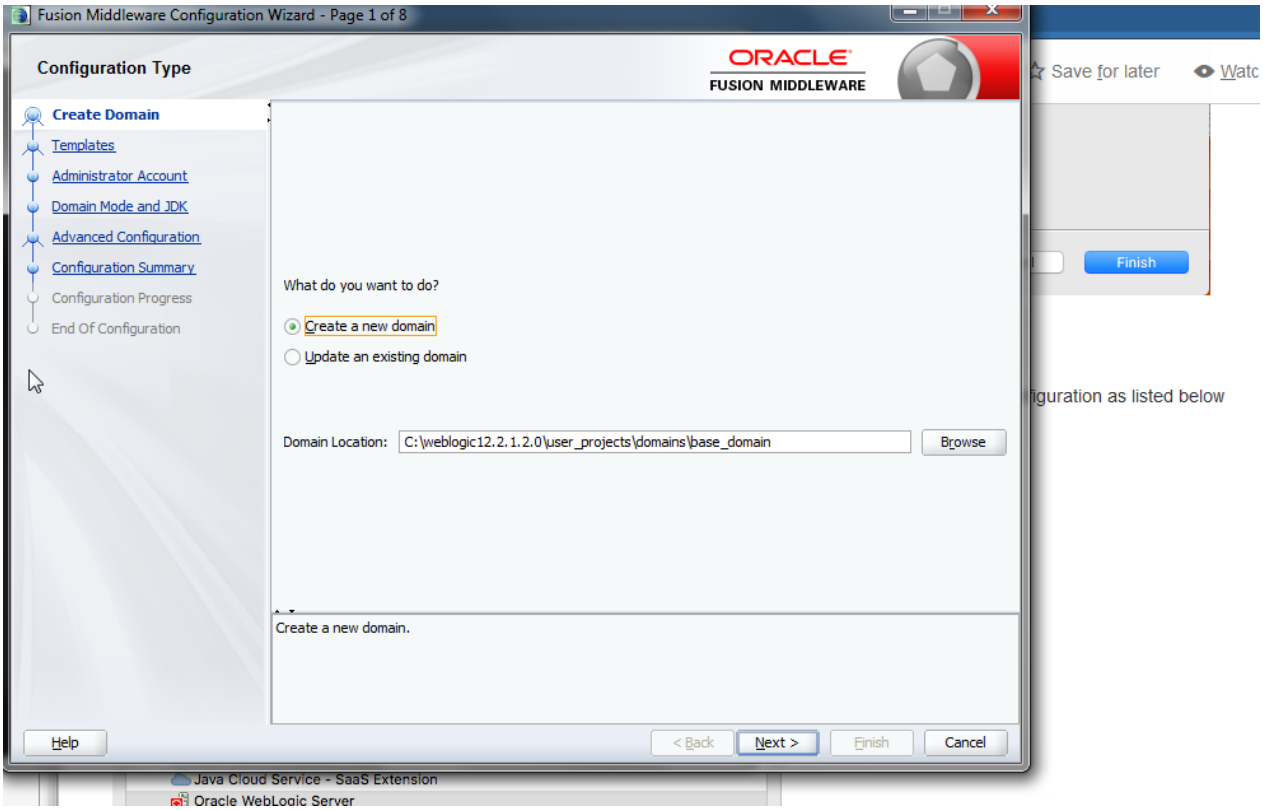

Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 6 of 9
Security Updates
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Security Updates
Installation Summary
Installation Progress
Installation Complete
Provide your email address to be informed of security issues, install the product and initiate configuration manager. View details.
Email:
Easier for you if you use your My Oracle Support email address/username.
I wish to receive security updates via My Oracle Support.
My Oracle Support Password:
Help
< Back
Next >
Finish
Cancel

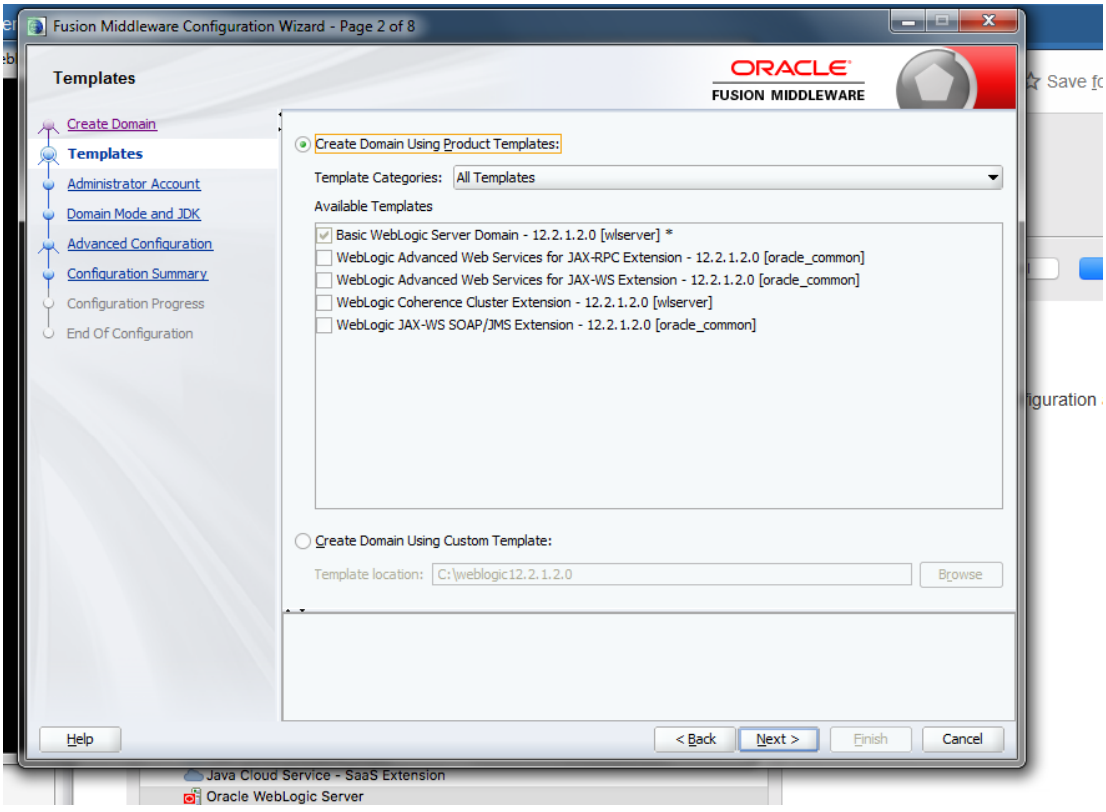

Oracle Fusion Middleware 12c WebLogic Server and Coherence Installation - Step 7 of 9
Installation Summary
ORACLE
FUSION MIDDLEWARE
Welcome
Auto Updates
Installation Location
Installation Type
Prerequisite Checks
Security Updates
Installation Summary
Installation Progress
Installation Complete
Install Oracle Fusion Middleware 12c WebLogic Server and Coherence
Installation Location
Oracle Home Location: C:\weblogic12.2.1.2.0
Log File Location: C:\Users\wkwok.ORADEV\AppData\Local\Temp\OraInstall2017-06-30_11-46-33AM\install2017-06-30_11-46-33AM.log
Disk Space
Required: 848 MB
Available: 225001 MB
Feature Sets to Install
Fusion Middleware Maven Support 12.2.1.2.0
FMW Platform Generic 12.2.1.2.0
OPatch 13.9.1.0.0
Administration Console Additional Language Help Files 12.2.1.2.0
CIE WLS Config 12.2.1.2.0
Core Application Server 12.2.1.2.0
Coherence Product Files 12.2.1.2.0
Web 2.0 HTTP Pub-Sub Server 12.2.1.2.0
WebLogic SCA 12.2.1.2.0
WebLogic Client Jars 12.2.1.2.0
Thirdparty JDBC Drivers 12.2.1.2.0
WebLogic Evaluation Database 12.2.1.2.0
Save Response File
Select Install to accept the above options and start the installation.
To change the above options before starting the installation, select the option to change in the left pane or use the Back button.
Help
< Back
Next >
Install
Cancel

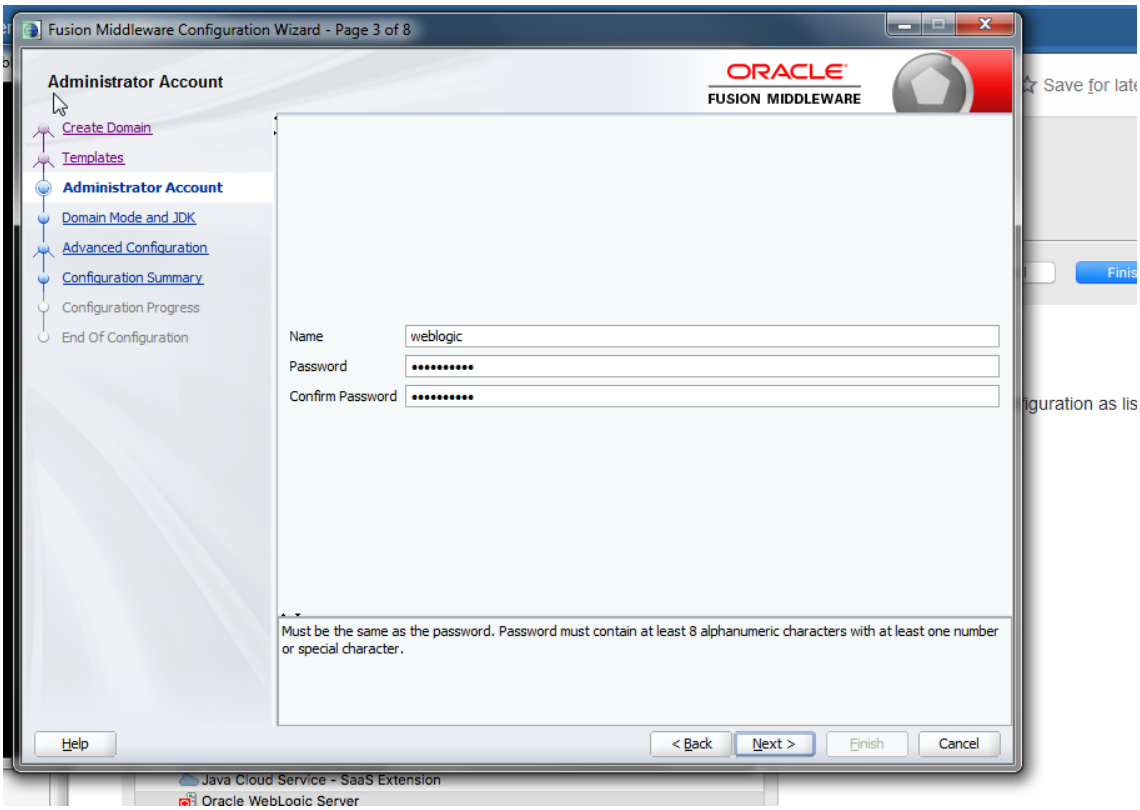


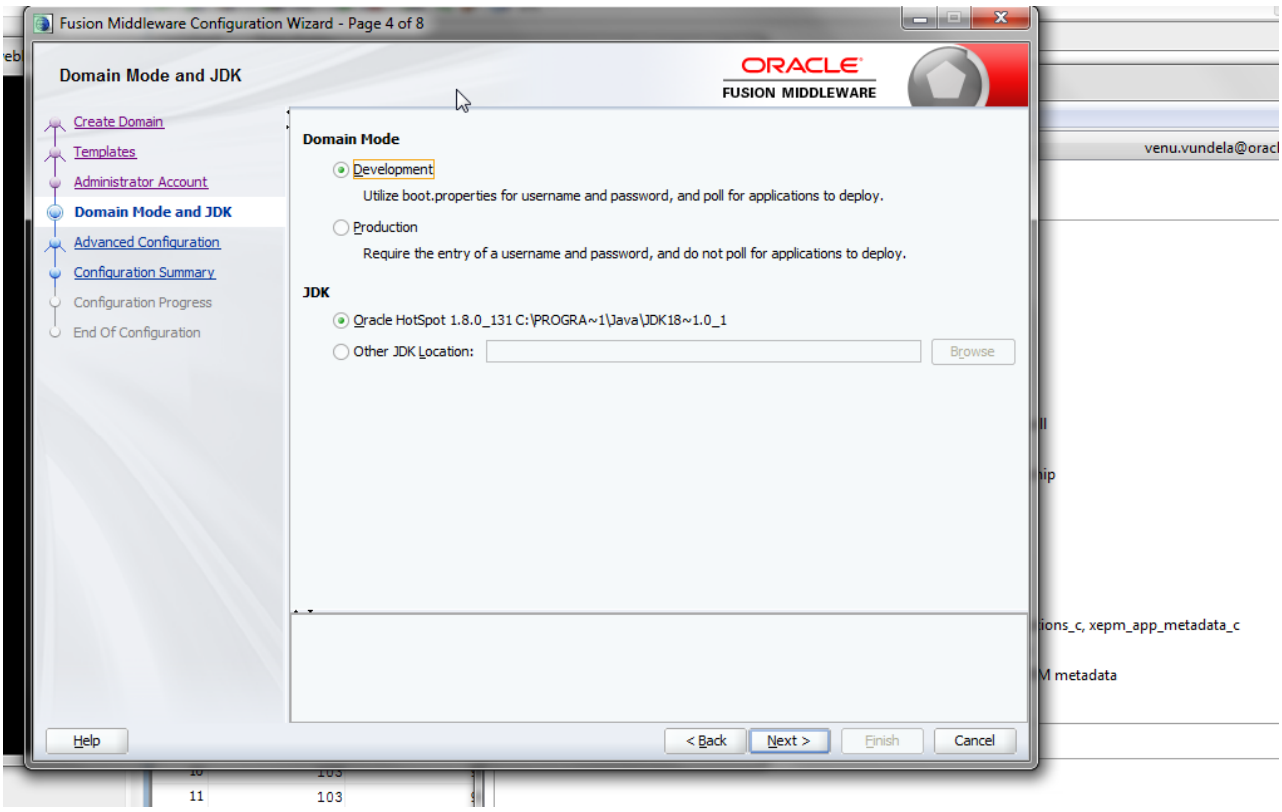


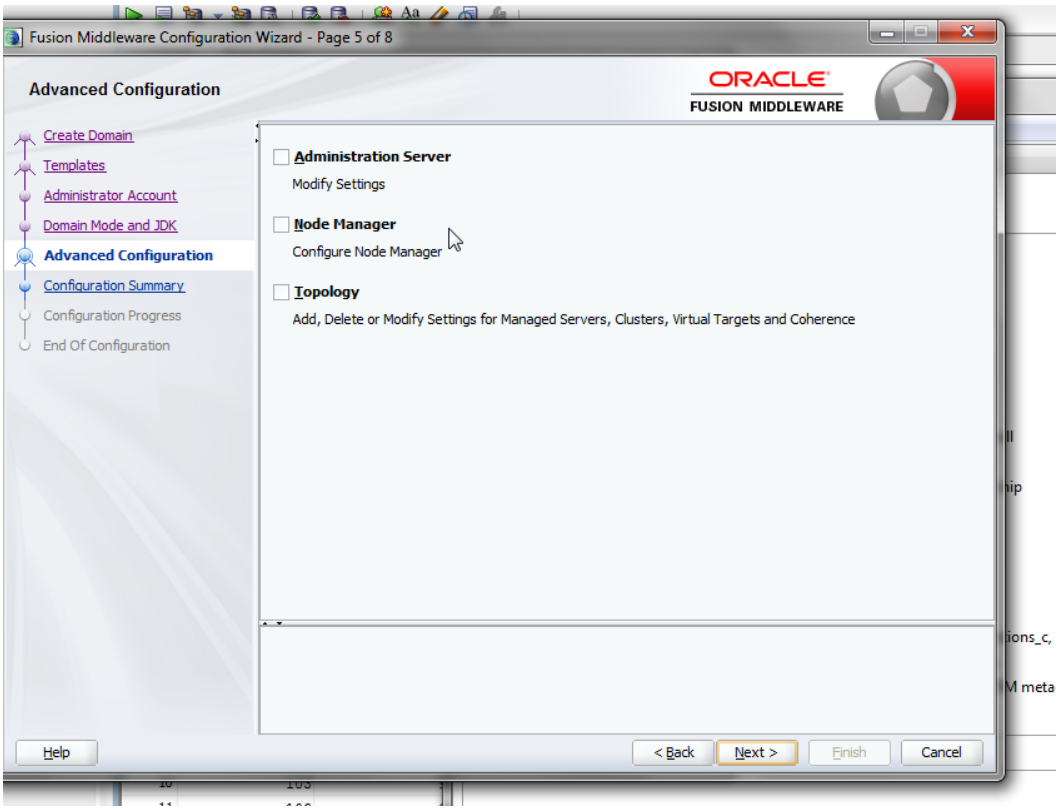
Create a weblogic domain as part of installation

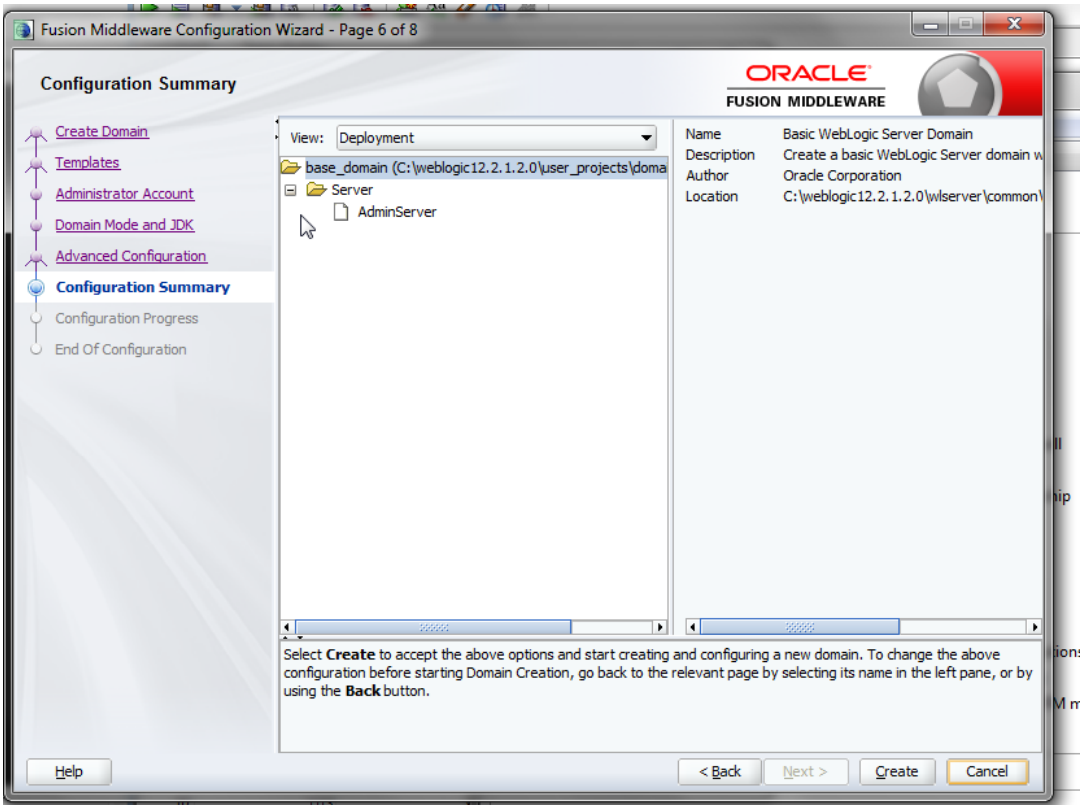

Fusion Middleware Configuration Wizard - Page 1 of 8
Configuration Type
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
What do you want to do?
Create a new domain
Update an existing domain
Domain Location:
C:\weblogic12.2.1.2.0\user_projects\domains\base_domain
Browse
Create a new domain.
Help
< Back
Next >
Finish
Cancel
Java Cloud Service - SaaS Extension
Oracle WebLogic Server
Save for later
Finish
iguration as listed below

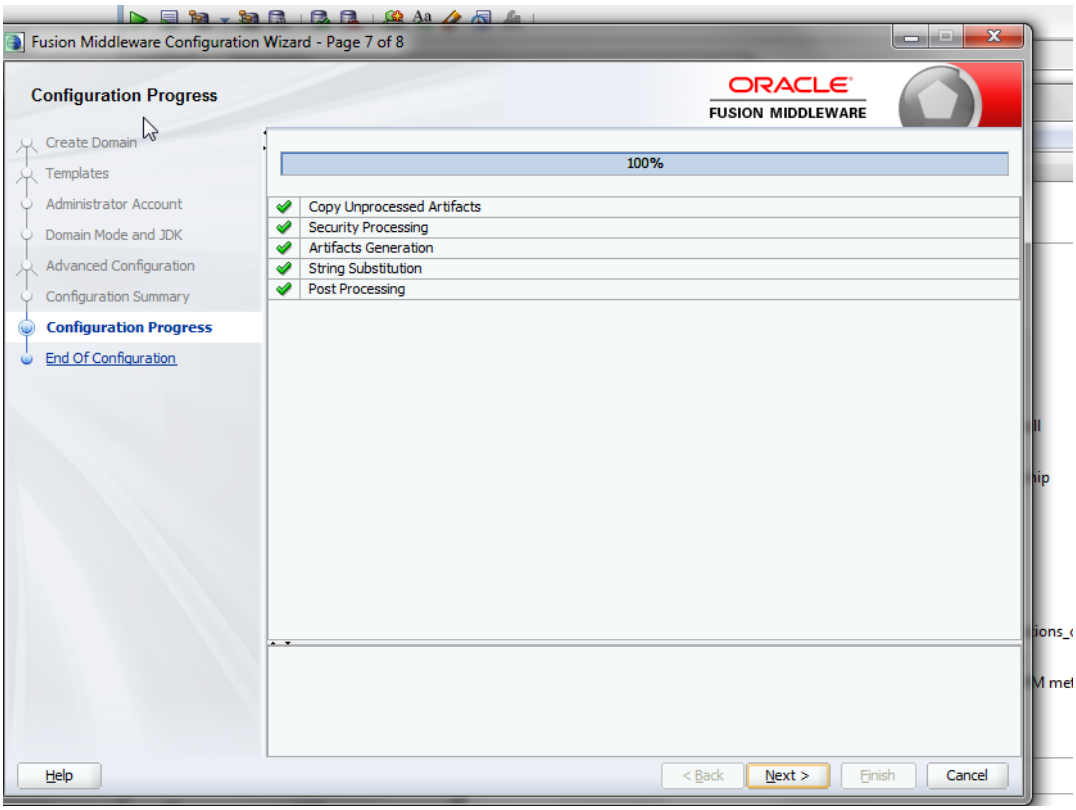

Fusion Middleware Configuration Wizard - Page 2 of 8
Templates
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
Create Domain Using Product Templates:
Template Categories:
All Templates
Available Templates
Basic WebLogic Server Domain - 12.2.1.2.0 [wlserver] *
WebLogic Advanced Web Services for JAX-RPC Extension - 12.2.1.2.0 [oracle_common]
WebLogic Advanced Web Services for JAX-WS Extension - 12.2.1.2.0 [oracle_common]
WebLogic Coherence Cluster Extension - 12.2.1.2.0 [wlserver]
WebLogic JAX-WS SOAP/JMS Extension - 12.2.1.2.0 [oracle_common]
Create Domain Using Custom Template:
Template location:
C:\weblogic12.2.1.2.0
Browse
Help
< Back
Next >
Finish
Cancel
Java Cloud Service - SaaS Extension
Oracle WebLogic Server
Save fo
iguration

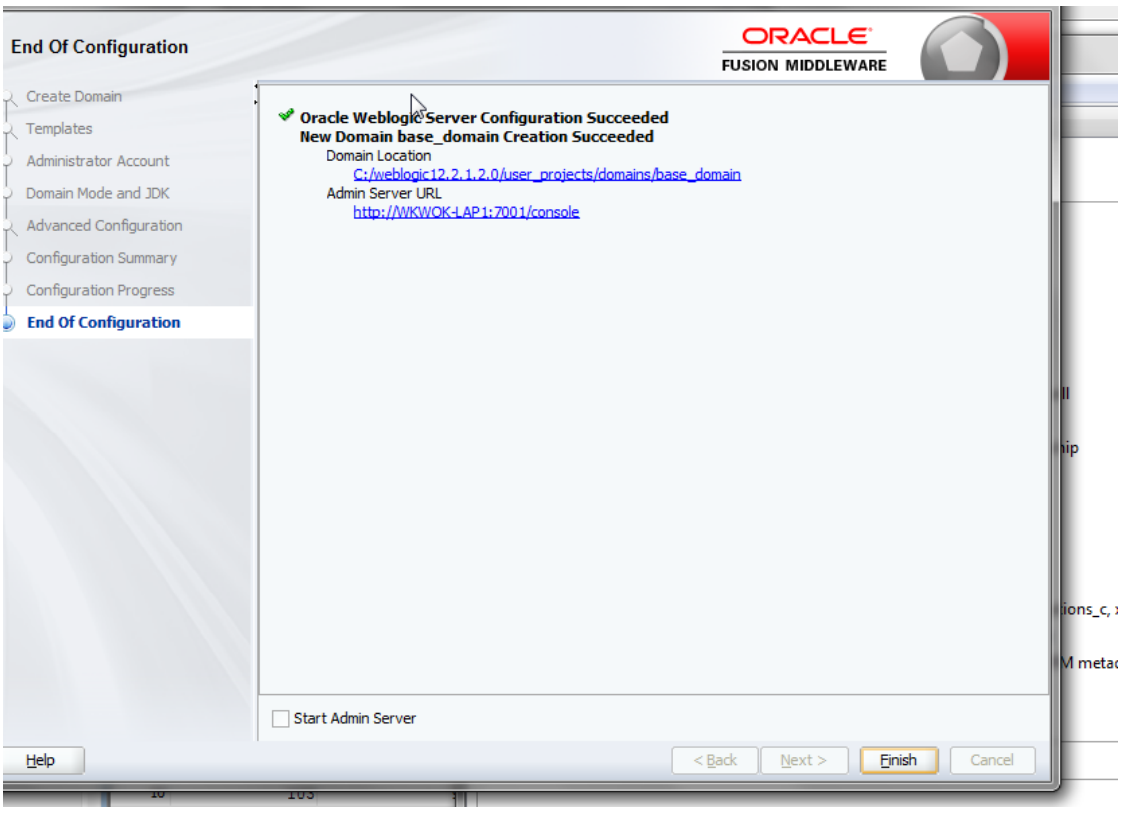

Fusion Middleware Configuration Wizard - Page 3 of 8
Administrator Account
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
Name
weblogic
Password
Confirm Password
Must be the same as the password. Password must contain at least 8 alphanumeric characters with at least one number or special character.
Help
< Back
Next >
Finish
Cancel
Java Cloud Service - SaaS Extension
Oracle WebLogic Server


Fusion Middleware Configuration Wizard - Page 4 of 8
Domain Mode and JDK
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
Domain Mode
Development
Utilize boot.properties for username and password, and poll for applications to deploy.
Production
Require the entry of a username and password, and do not poll for applications to deploy.
JDK
Oracle HotSpot 1.8.0_131 C:\PROGRA~1\Java\JDK18~1.0_1
Other JDK Location:
Browse
Help
< Back
Next >
Finish
Cancel


Fusion Middleware Configuration Wizard - Page 5 of 8
Advanced Configuration
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
Administration Server
Modify Settings
Node Manager
Configure Node Manager
Topology
Add, Delete or Modify Settings for Managed Servers, Clusters, Virtual Targets and Coherence
Help
< Back
Next >
Finish
Cancel


Fusion Middleware Configuration Wizard - Page 6 of 8
Configuration Summary
ORACLE
FUSION MIDDLEWARE
Create Domain
Templates
Administrator Account
Domain Mode and JDK
Advanced Configuration
Configuration Summary
Configuration Progress
End Of Configuration
View: Deployment
base_domain (C:\weblogic12.2.1.2.0\user_projects\doma
Server
AdminServer
Name Basic WebLogic Server Domain
Description Create a basic WebLogic Server domain w
Author Oracle Corporation
Location C:\weblogic12.2.1.2.0\wlserver\common\
Select Create to accept the above options and start creating and configuring a new domain. To change the above configuration before starting Domain Creation, go back to the relevant page by selecting its name in the left pane, or by using the Back button.
Help
< Back
Next >
Create
Cancel





Lombok

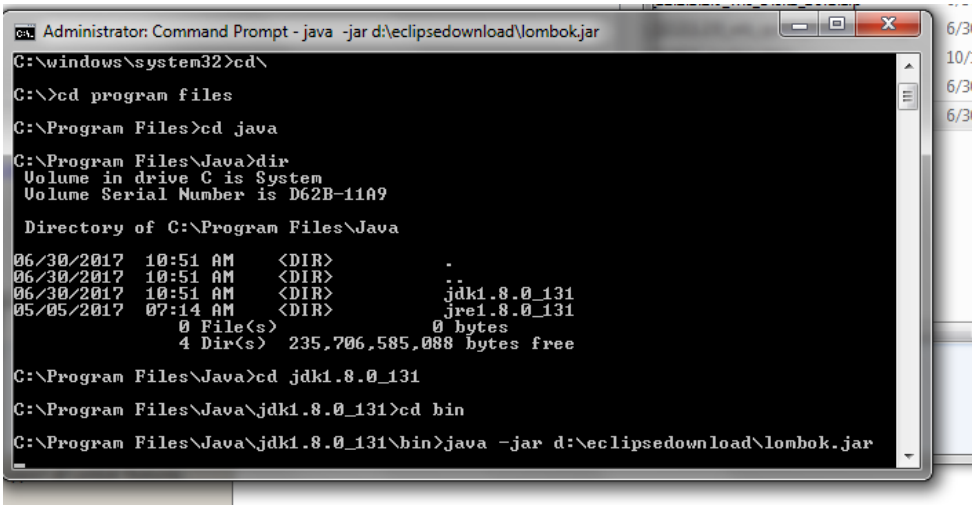
Download Lombok from the location below. 16.1.16 is the latest version as of writing this guide. While you are the website, checkout the video to see how cool lombok is.

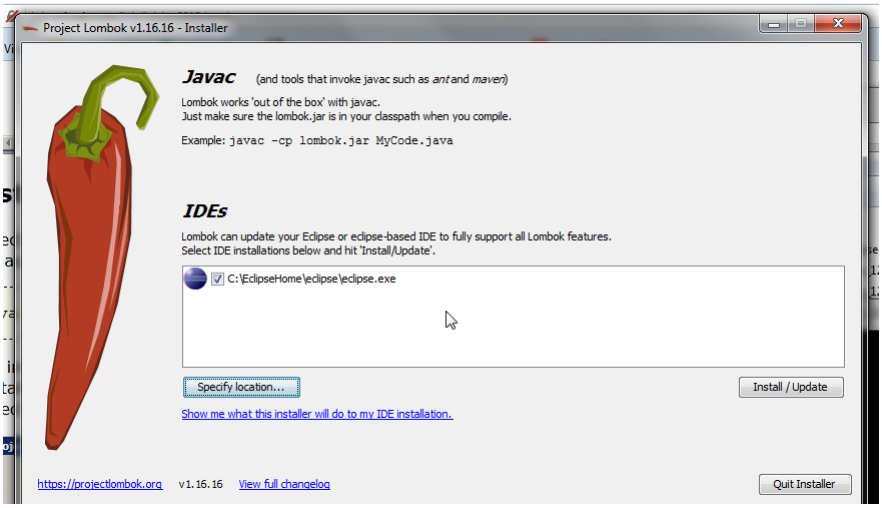
https://projectlombok.org/

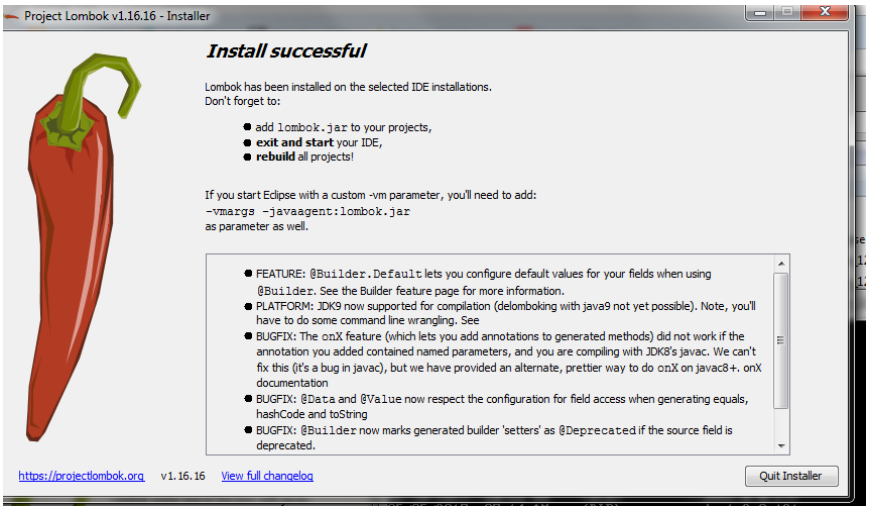
The installer is an executable jar that can automatically detects the Eclipse IDE installation location. If it doesn't detect automatically point the installer to the location of "eclipse" executable and complete installation. Restart Eclipse once the installation is complete.

Run java command in your JDK bin like below

java -jar <drive letter>:\lombok.jar







SQL Developer

TBD

Docker and Kubernetes

What is Docker?

Docker is a set of platform as a service products that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines. [https://en.wikipedia.org/wiki/Docker_(software)][https://www.docker.com/why-docker]

What is Kubernetes?

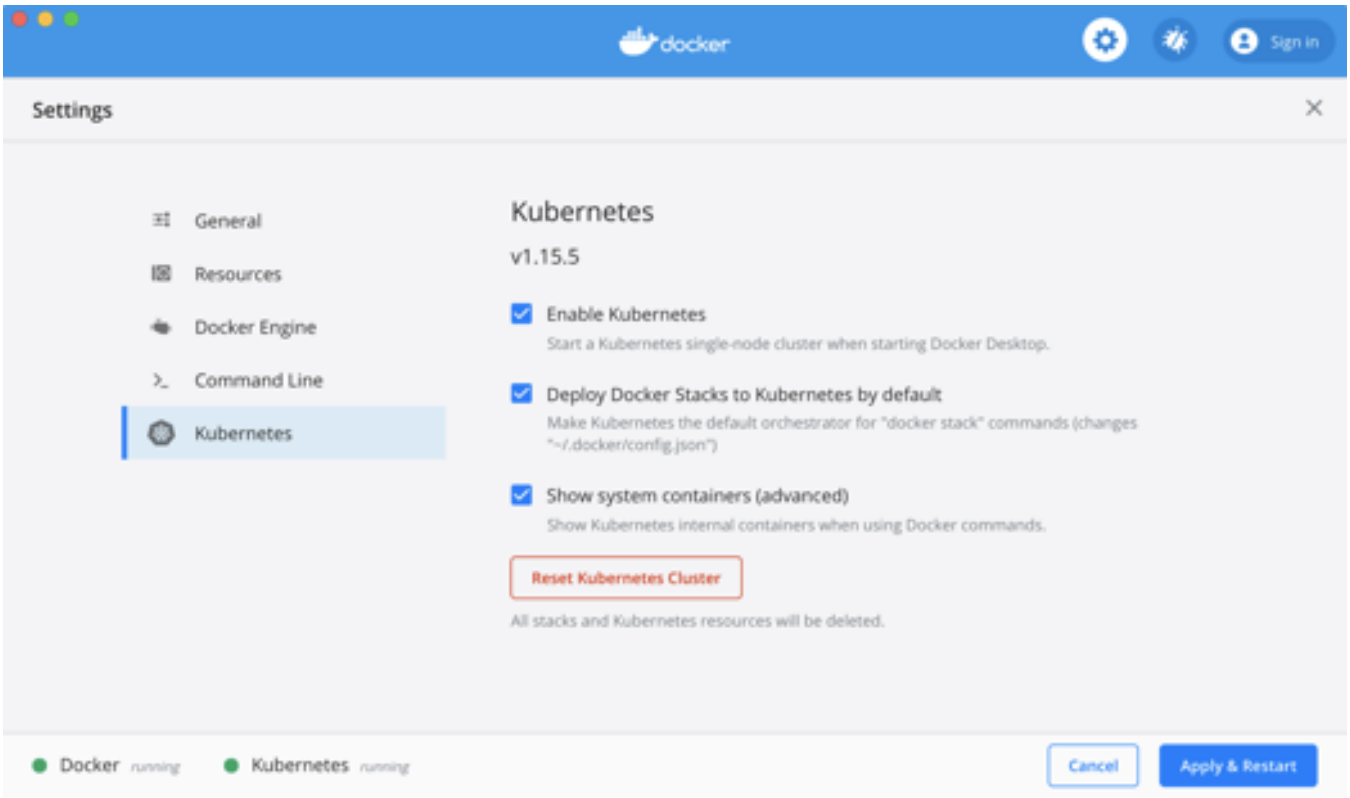
Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google, and is now maintained by the Cloud Native Computing Foundation. It aims to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts". It works with a range of container tools, including Docker. [https://en.wikipedia.org/wiki/Kubernetes][https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/]

Installing Docker Desktop for Mac and Windows

Download docker desktop (Docker.dmg v2.2.x ~700MB) from below link.

- https://hub.docker.com/editions/community/docker-ce-desktop-windows
- https://hub.docker.com/editions/community/docker-ce-desktop-mac/

To use docker desktop to manage kubernetes enable from Preferences>Kubernetes.



Test installation

```
$ which docker

/usr/local/bin/docker

$ docker --version

Docker version 19.03.8, build afacb8b

$ which kubectl

/usr/local/bin/kubectl

$ kubectl version

Client Version: version.Info{Major:"1", Minor:"17", GitVersion:"v1.17.3", GitCommit:"
06ad960bfd03b39c8310aaf92d1e7c12ce618213", GitTreeState:"clean", BuildDate:"2020-02-11T18:14:22Z", GoVersion:"
go1.13.6", Compiler:"gc", Platform:"darwin/amd64"}

Server Version: version.Info{Major:"1", Minor:"15", GitVersion:"v1.15.5", GitCommit:"
20c265fef0741dd71a66480e35bd69f18351daea", GitTreeState:"clean", BuildDate:"2019-10-15T19:07:57Z", GoVersion:"
go1.12.10", Compiler:"gc", Platform:"linux/amd64"}
```

Demo Docker

We will use “tutum/hello-world” [https://hub.docker.com/r/tutum/hello-world/] image to test docker deployments. It has Apache with a 'Hello World' page listening in port 80.

```
// run image “tutum/hello-world” and

// map container internal port “80” to container external port “30080”

// if the image is not present locally, docker will download the image from

// default docker registry (Docker Hub in this case).


$ docker run -d -p 30080:80 tutum/hello-world

Unable to find image 'tutum/hello-world:latest' locally

latest: Pulling from tutum/hello-world

Image docker.io/tutum/hello-world:latest uses outdated schema1 manifest format. Please upgrade to a schema2
image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-
schema-v1/

658bc4dc7069: Pull complete

a3ed95caeb02: Pull complete

af3cc4b92fa1: Pull complete

d0034177ece9: Pull complete

983d35417974: Pull complete

Digest: sha256:0d57def8055178aafb4c7669cbc25ec17f0acdab97cc587f30150802da8f8d85

Status: Downloaded newer image for tutum/hello-world:latest

b97534c7e569d851d38199e3107d15e23f350b2d563e6180d8c0659718660ef4


// now we can access the web service through curl or browser.

$ curl http://localhost:30080

<html>

<head>

    <title>Hello world!</title>

    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700' rel='stylesheet' type='text/css'>

    <style>
```

```
    body {
        background-color: white;
        text-align: center;
        padding: 50px;
        font-family: "Open Sans","Helvetica Neue",Helvetica,Arial,sans-serif;
    }

    #logo {
        margin-bottom: 40px;
    }
    </style>
</head>
<body>
    <img id="logo" src="logo.png" />
    <h1>Hello world!</h1>
    <h3>My hostname is b97534c7e569</h3>    </body>
</html>


// listing all images stored locally
$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
tutum/hello-world                   latest              31e17b0746e4        4 years ago         17.8MB


// listing all containers
$ docker container ls
CONTAINER
ID        IMAGE                                 COMMAND                  CREATED             STATUS
PORTS                  NAMES
b97534c7e569        tutum/hello-world                   "/bin/sh -c 'php-fpm…"   17 minutes ago      Up 17
minutes       0.0.0.0:30080->80/tcp   heuristic_hofstadter


// stop running container with container-id
$ docker stop b97534c7e569
b97534c7e569


// remove/delete containers and images
$ docker system prune
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache
```

```
Are you sure you want to continue? [y/N]

// container information using container-name
$ docker inspect --format '{{ .NetworkSettings.IPAddress }}' heuristic_hofstadter
172.17.0.2

$ docker inspect heuristic_hofstadter

$ docker kill heuristic_hofstadter
heuristic_hofstadter

// running container with “rm” will remove the container once the container is stopped.
$ docker image rm tutum/hello-world
```

Demo Kubernetes

We will use “tutum/hello-world” [https://hub.docker.com/r/tutum/hello-world/] image to create Kubernetes deployment and service.

Deployment will create 3 replicas of our application and service will provide load balancing to the pods.

We will create a yml file which will contain both Deployment and Service specification separated by “---"

The outcome of this demo will be, when we will hit the endpoints curl http://localhost:30080, we will get response from different container.

The range of valid ports is 30000-32767 to expose services on Kubernetes cluster for NodePort.

```
$ mkdir kuberneted-demo
$ cd kuberneted-demo/
$ nano k8.yml
$ ls
k8.yml
$ cat k8.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-web-server-hello-world
  labels:
    app: demo-web-server-hello-world
spec:
  replicas: 3
  selector:
    matchLabels:
      app: demo-web-server-hello-world
```

```
  template:
    metadata:
      labels:
        app: demo-web-server-hello-world
    spec:
      containers:
        - name: demo-web-server-hello-world-containers
          image: tutum/hello-world:latest
          # for local image
          # imagePullPolicy: Never
          env:
          - name: SPRING_PROFILES_ACTIVE
            value: local
          ports:
            - containerPort: 80



---


apiVersion: v1
kind: Service
metadata:
  name: demo-web-server-hello-world
spec:
  ports:
      - protocol: "TCP"
        # Port accessible inside cluster
        port: 9090
        # Port to forward to inside the pod
        targetPort: 80
        # Port accessible outside cluster
        nodePort: 30080
  selector:
    app: demo-web-server-hello-world
  type: NodePort


// create deployment and service
$ kubectl apply -f ./k8.yml
deployment.apps/demo-web-server-hello-world created
service/demo-web-server-hello-world created
```

```
$ kubectl get deploy -o wide

NAME                          READY   UP-TO-
DATE   AVAILABLE   AGE   CONTAINERS                              IMAGES                       SELECTOR

demo-web-server-hello-world   3/3     3            3           22s   demo-web-server-hello-world-
containers   tutum/hello-world:latest   app=demo-web-server-hello-world


$ kubectl get pods -o wide

NAME                                          READY   STATUS    RESTARTS   AGE   IP          NODE             N
OMINATED NODE   READINESS GATES

demo-web-server-hello-world-5dbcb58ff5-jk457   1/1     Running   0          56s   10.1.0.98   docker-
desktop   <none>           <none>

demo-web-server-hello-world-5dbcb58ff5-tsncl   1/1     Running   0          56s   10.1.0.96   docker-
desktop   <none>           <none>

demo-web-server-hello-world-5dbcb58ff5-x9g27   1/1     Running   0          56s   10.1.0.97   docker-
desktop   <none>           <none>


$ kubectl get service -o wide

NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE   SELECTOR

demo-web-server-hello-world   NodePort    10.110.150.15   <none>        9090:30080/TCP   83s   app=demo-web-
server-hello-world

kubernetes                    ClusterIP   10.96.0.1       <none>        443/TCP          11d   <none>


$ curl http://localhost:30080

<html>

<head>

     <title>Hello world!</title>

     <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700' rel='stylesheet' type='text/css'>

     <style>

     body {

         background-color: white;

         text-align: center;

         padding: 50px;

         font-family: "Open Sans","Helvetica Neue",Helvetica,Arial,sans-serif;

     }


     #logo {

         margin-bottom: 40px;

     }

     </style>

</head>

<body>

     <img id="logo" src="logo.png" />

     <h1>Hello world!</h1>
```

```
     <h3>My hostname is demo-web-server-hello-world-7b68b88bfd-rk4j7</h3>                <h3>Links found</h3>

                        <b>DEMP_WEB_SERVER_HELLO_WORLD</b> listening in 9090 available at tcp://10.97.60.111:
9090<br />

                             <b>KUBERNETES</b> listening in 443 available at tcp://10.96.0.1:443<br />

                    </body>

</html>


// run below command and observe the hostname in the output.

// the hostname will keep changing since the request is processed by different application instance.

$ while true; do curl http://localhost:30080; sleep 1; done;



//NOTE: Killing individual PODS will not kill the it as

// Kubernetes will make sure to required number pods as per specification.


// delete deployment and service

$ kubectl delete deployment demo-web-server-hello-world

deployment.extensions "demo-web-server-hello-world" deleted


$ kubectl delete service demo-web-server-hello-world

service "demo-web-server-hello-world" deleted
```

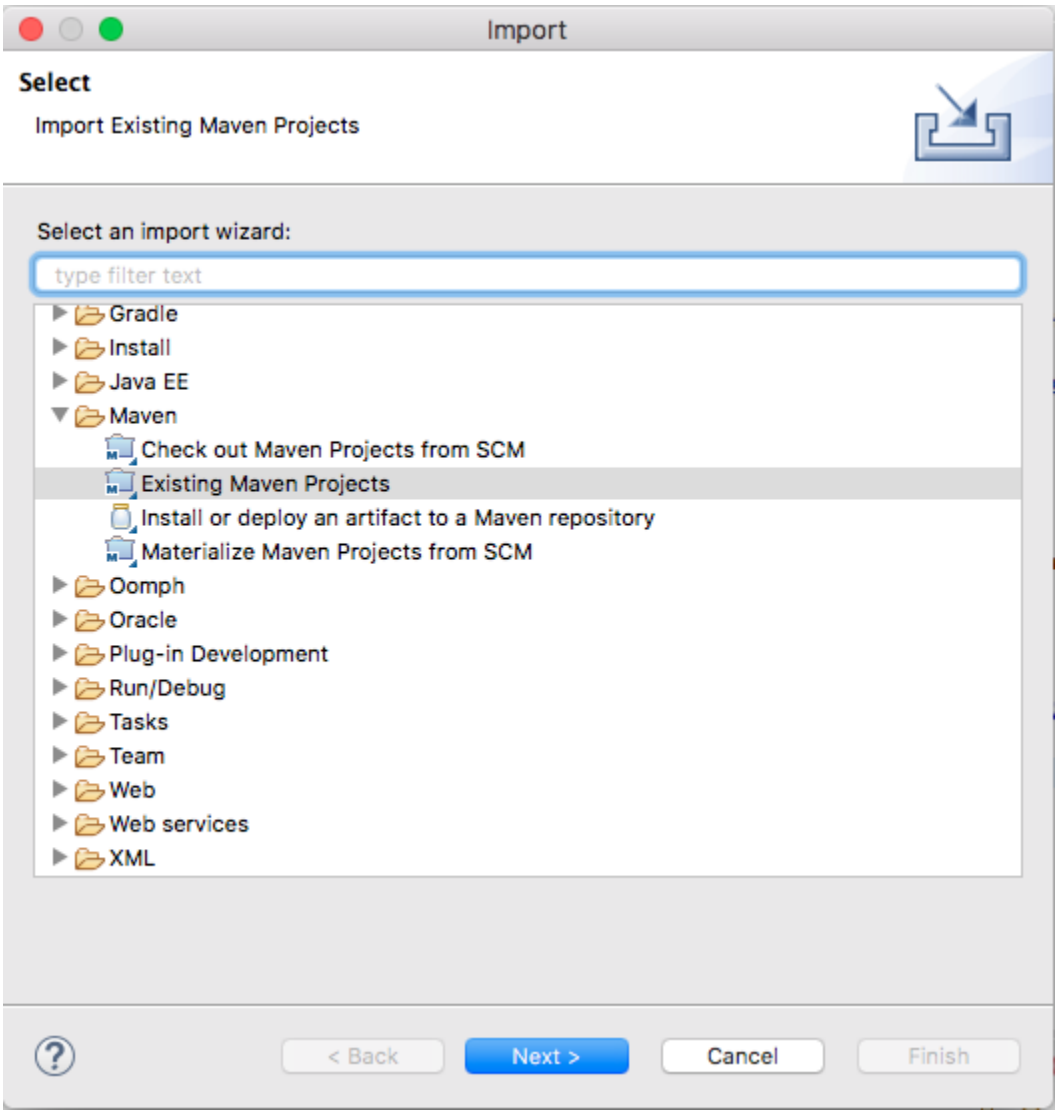
References

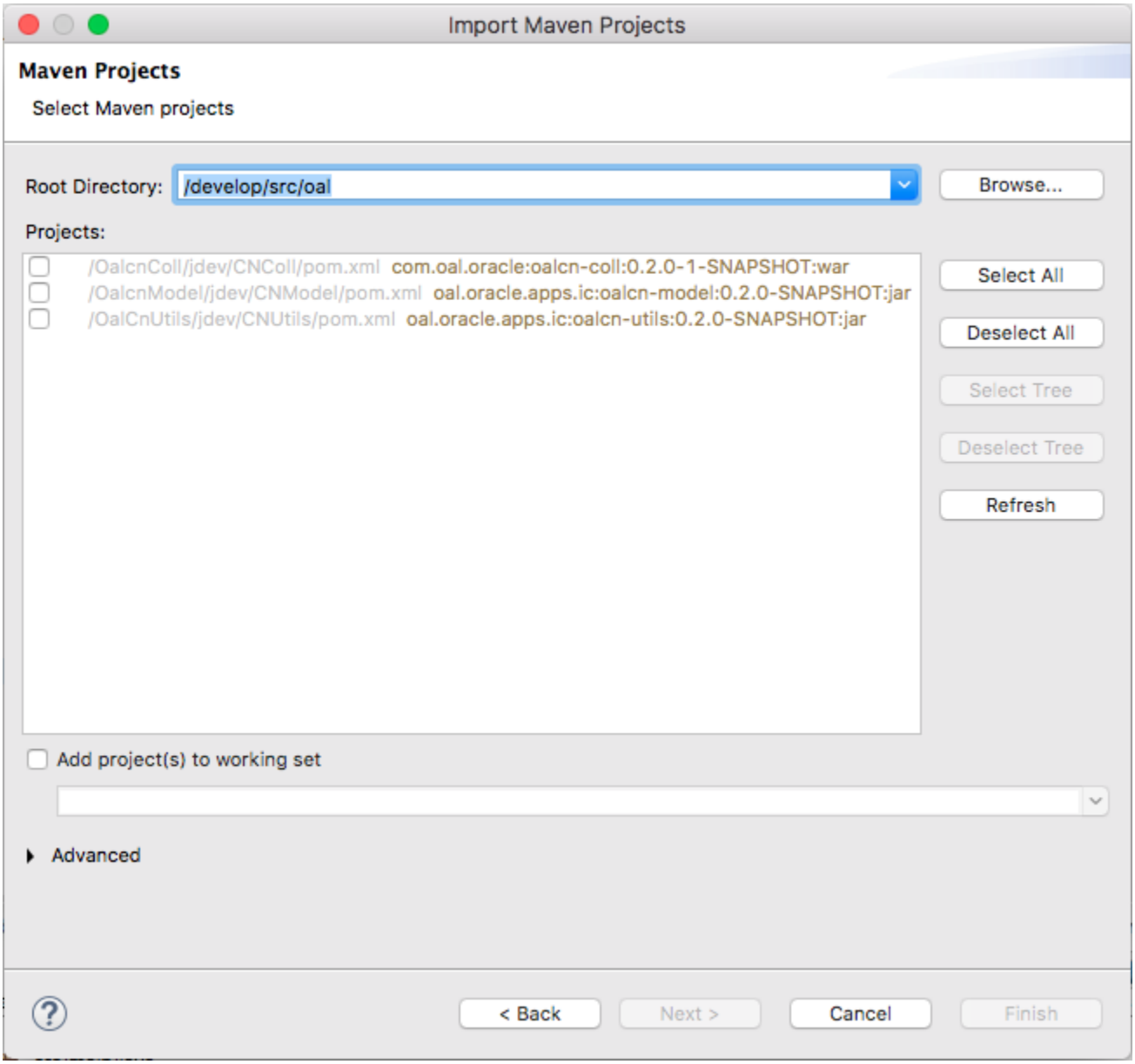
- https://stackoverflow.com/questions/55462654/cant-access-minikube-loadbalancer-service-from-host-machine
- https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.10/#service-v1-core
- https://confluence.oraclecorp.com/confluence/display/OALMS/Migrating+from+JCS+to+OKE
- https://kubernetes.io/docs/home/
- https://docs.docker.com/

Configuration

Importing & Building Projects in Eclipse

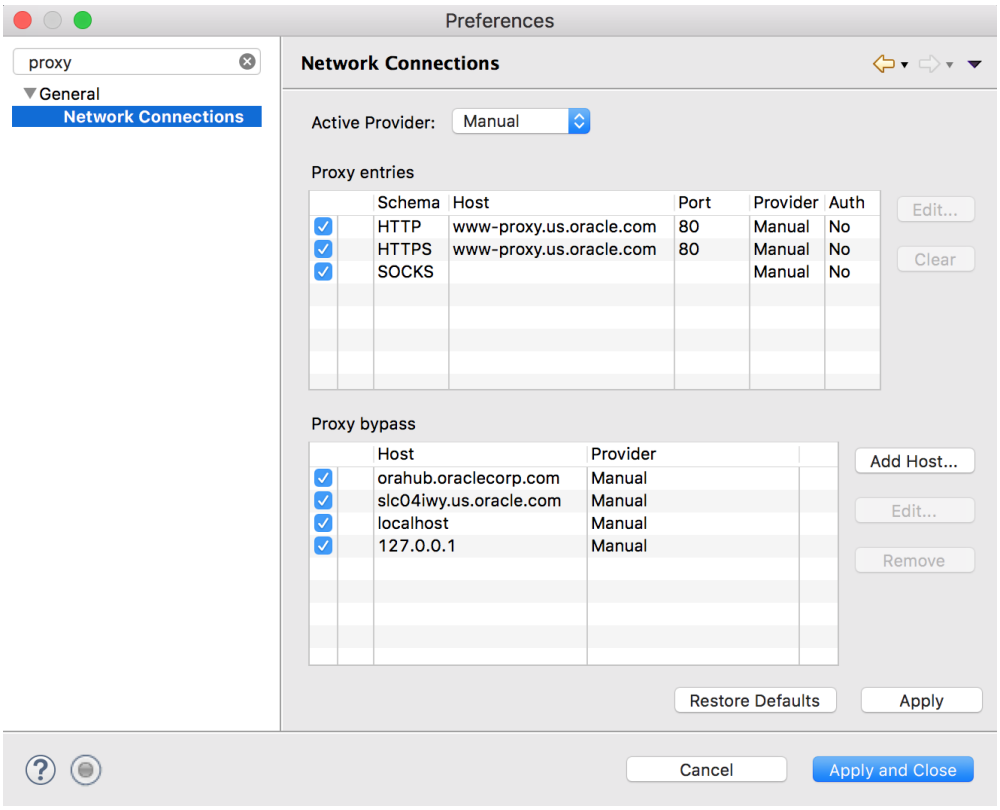
Import OalCnColl, OalCnUtils & OalCnUtils as Maven Projects using the respective pom.xml files. All projects can be imported together if they have the same common parent folder


Import
Select
Import Existing Maven Projects
Select an import wizard:
type filter text
Gradle
Install
Java EE
Maven
Check out Maven Projects from SCM
Existing Maven Projects
Install or deploy an artifact to a Maven repository
Materialize Maven Projects from SCM
Oomph
Oracle
Plug-in Development
Run/Debug
Tasks
Team
Web
Web services
XML
< Back
Next >
Cancel
Finish



Proxy Configuration

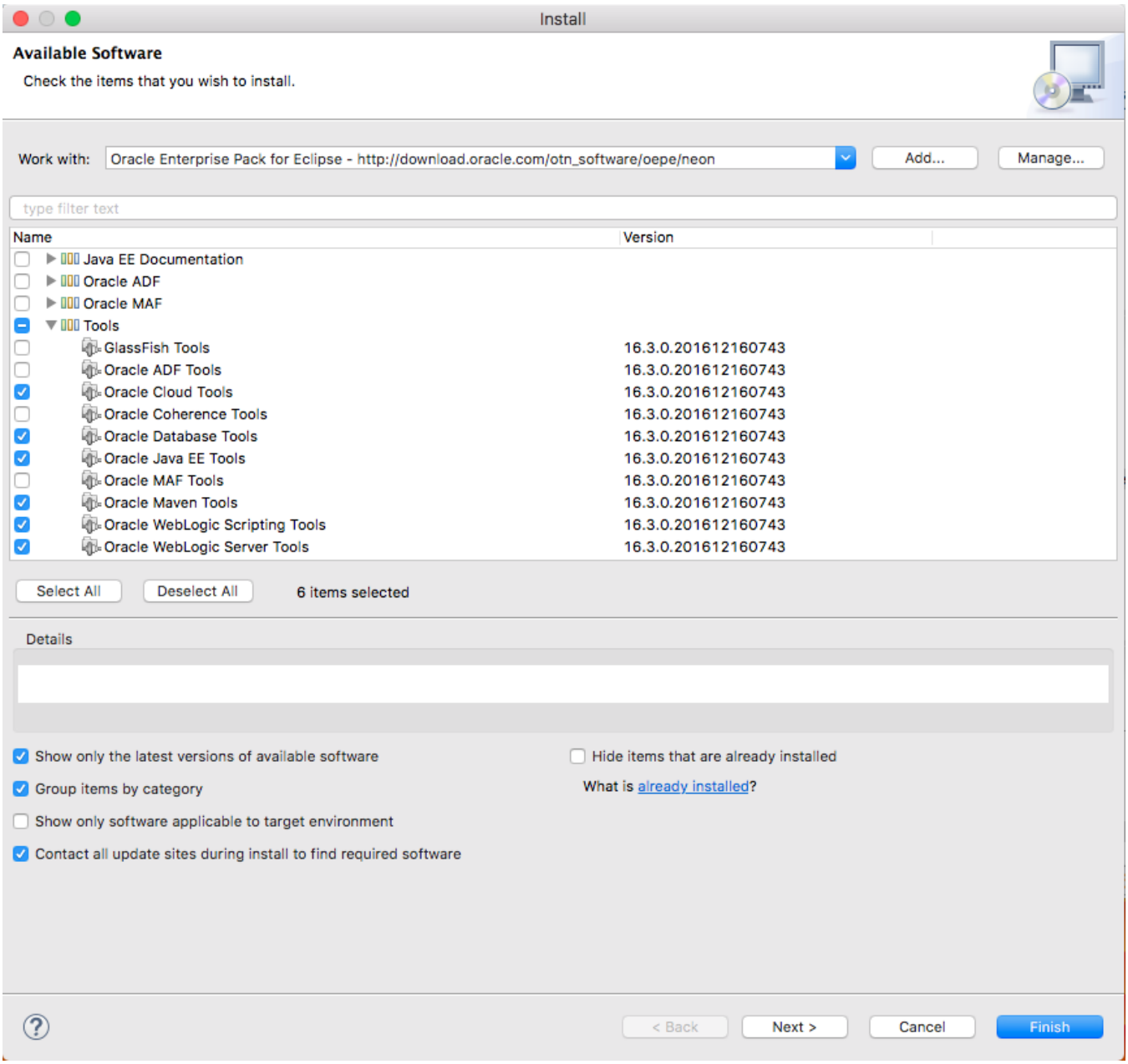
Configure the Proxy Eclipse Preferences as noted in the screenshot below. This will allow Eclipse to download plugins and updates from the marketplace.



Eclipse Weblogic Integration

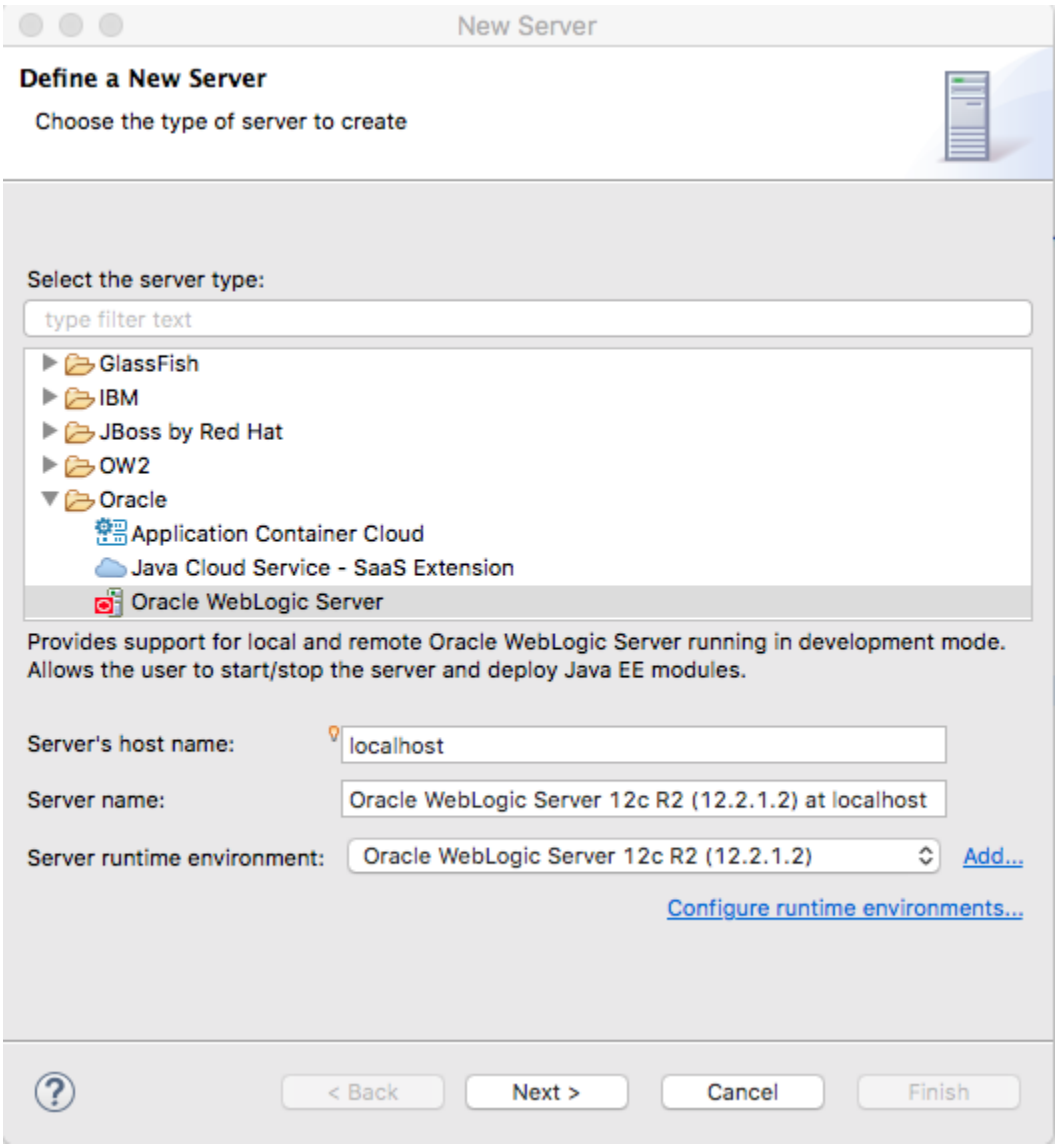
Install the Oracle Enterprise Pack for Eclipse using the below mentioned site (from Help Install Software). The required components are highlighted in the below screenshot.

Oracle Enterprise Pack for Eclipse - http://download.oracle.com/otn_software/oepe/neon (you can put in Oracle Enterprise Pack for Eclipse - http://download.oracle.com/otn_software/oepe/neon and click Add)

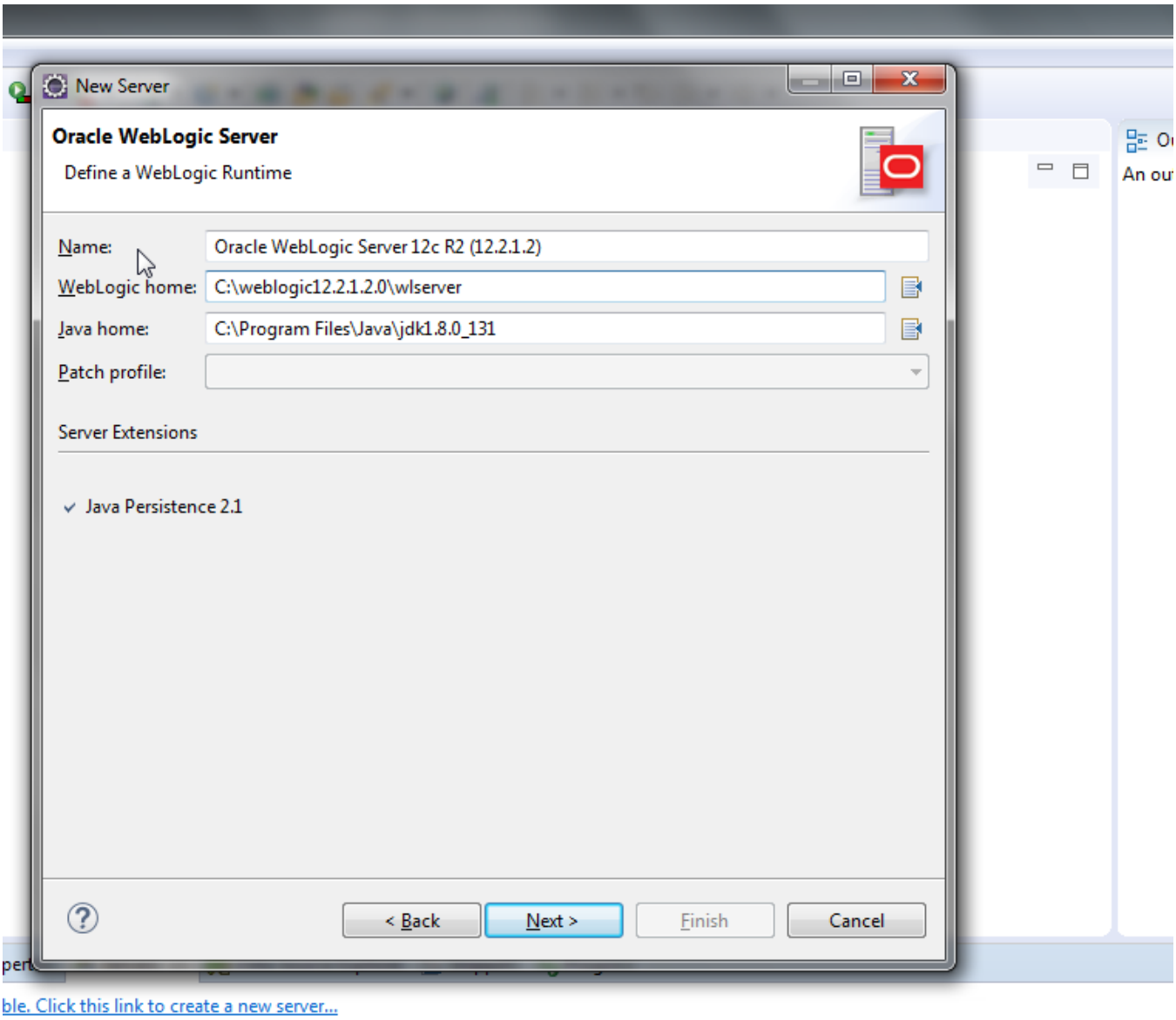


Creating a Server

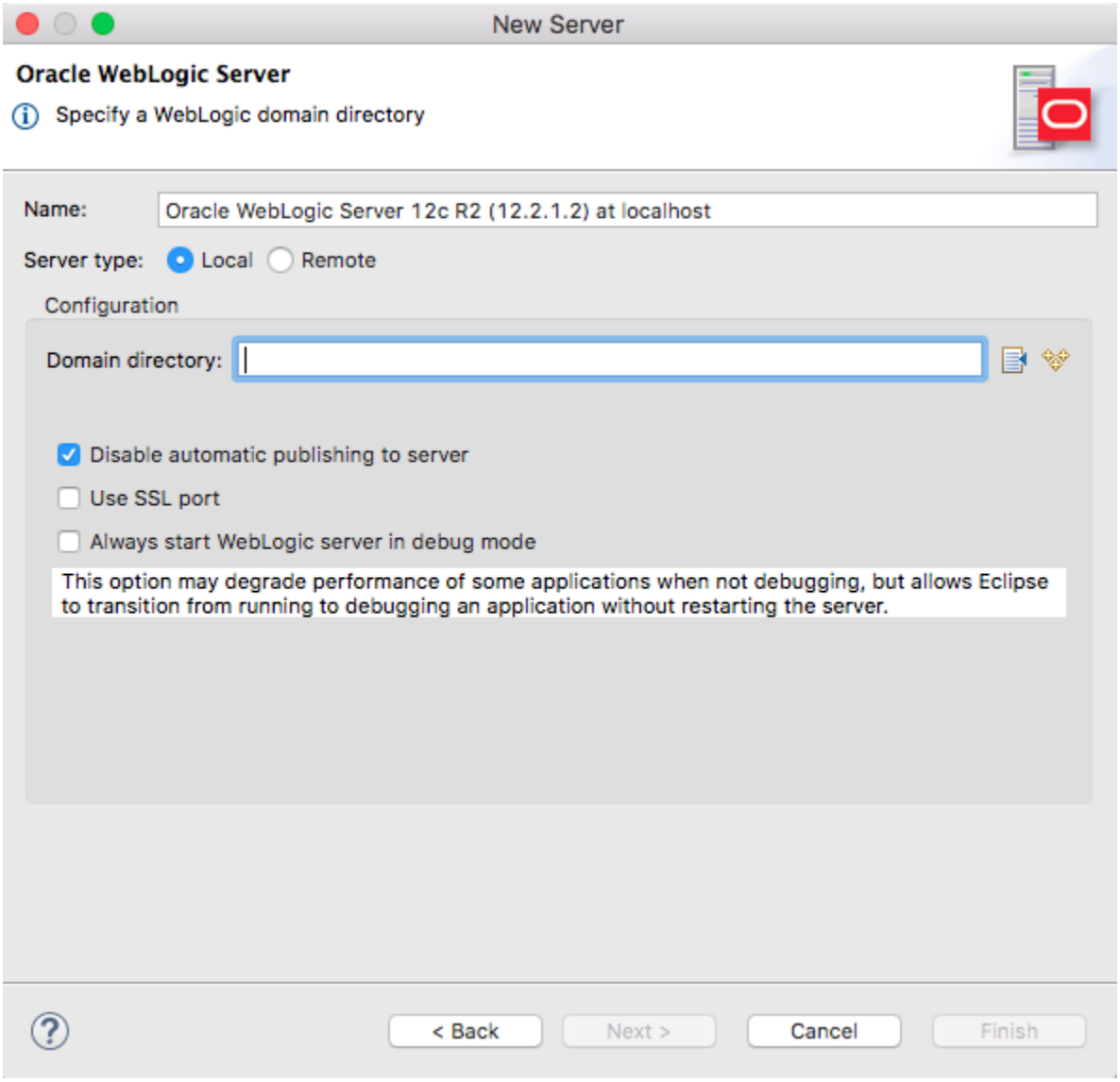
From the Window Menu open the Servers view. In the Servers view, start by creating a new Server and use the configuration as listed below



In case, you need have multiple runtime environments. Set up like below:



Next > Specify the base domain directory and finish the setup



Running/Debugging on Server

Right Click on the OalCnColl project and choose Run/Debug on Server option. Follow the default prompts and this should start the Weblogic Server and Deploy the Application. There is an option to use defaults everytime a Run/Debug is selected. Once checked, the prompts should not appear again.

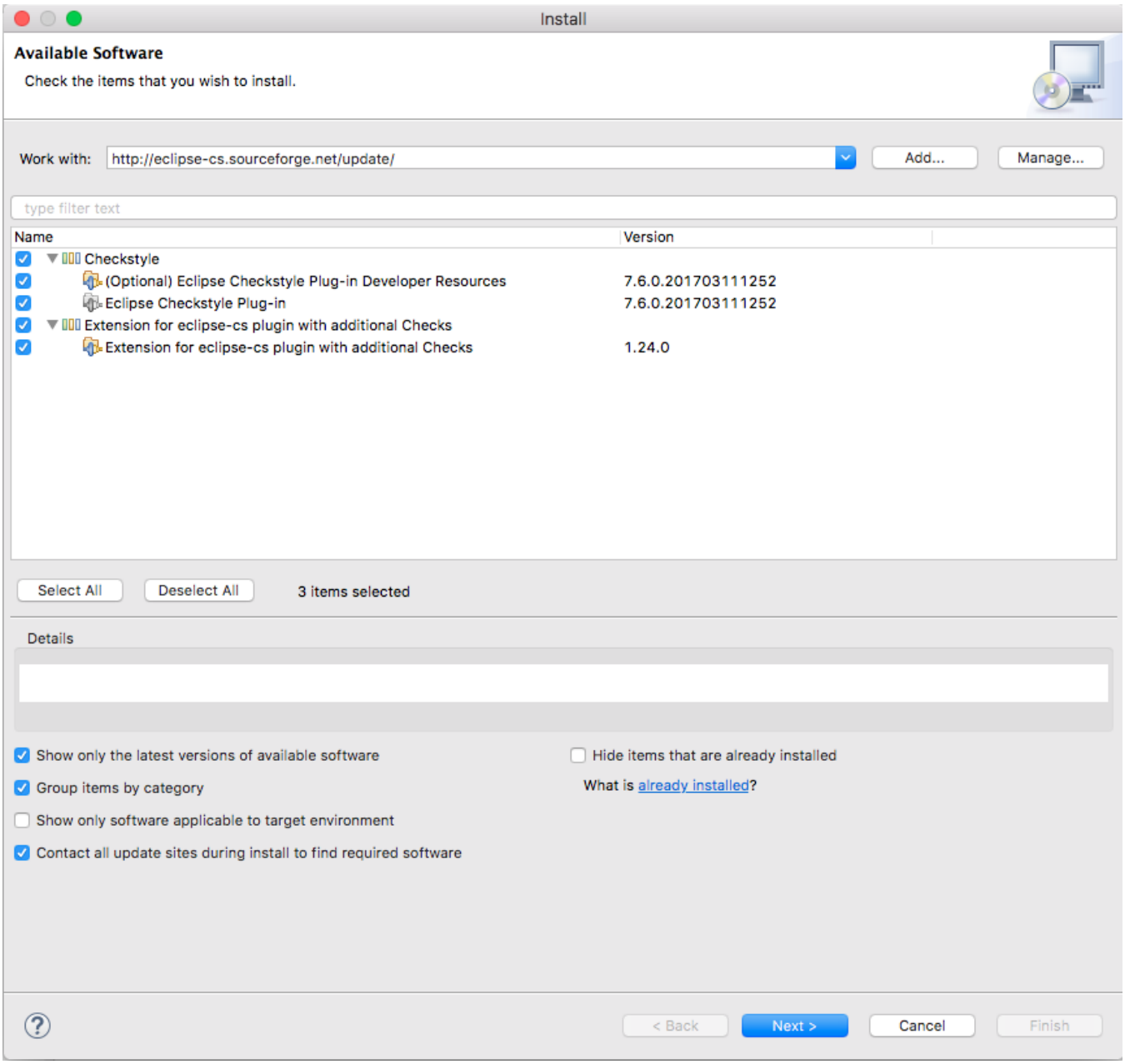
Sonarlint Setup

From the Eclipse Market Place, install Sonarlint for static code analysis. Once installed, restart Eclipse and open the "SonarLint On-The-Fly" & "SonarLint Rule Description" views.

CheckStyle Setup

Install Checkstyle from the Install New Software menu and following the below configuration to install CheckStyle static code analysis plugin.

Link to use as of 18 Sep 2018 `https://checkstyle.org/eclipse-cs/update`



Checkstyle Custom Configuration

Checkstyle is performed based on a series of rules that can be added, removed or modified according to the development needs. Our set of rules are a modified version of the Checkstyle's default XML.

We have different set of rules with a few changes depending on the project they are performed on. In this case the rules vary between OalCnUtils and OalCnModel.

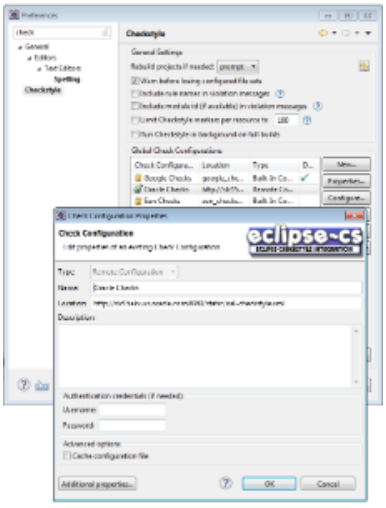
This is the default XML file from Checkstyle:

Some of these rules are not applied on our projects, for example:

Rule	Explanation
<module name="JavadocMethod"> <property name=" allowMissingPropertyJavadoc" value=" true" /> </module>	This allows us to skip comments on getters and setters since it is redundant to explain what this methods do. However if the getter/setter implements a different logic apart from just retrieving or setting then this is marked as an error on the result XML.
<module name="JavadocType"/>	Removed due to discussion with our team that not all classes are required to have Javadoc comments.
<module name="JavadocVariable"/>	Removed due to number of variables that were self explanatory. However we are introducing javadoc comments on important variables that need explaining.
<module name="JavadocStyle"/>	Removed due to limitations on comment formatting.
<module name="LineLength"> <property name="max" value="120" /> </module>	Added additional characters permitted per line. All of our most used code editors allow 120 characters without presenting formatting errors or unreadable code.
<module name="HiddenField" />	Removed due to some classes that are autogenerated and use the same local variable names of the class attributes.
<module name="MagicNumber"> <property name="ignoreAnnotation" value="true" /> </module>	Modified to ignore numbers in annotations, most of them being limit characters on attributes.
<module name="Regexp"> <property name="format" value=" System\.out\.println" /> <property name="illegalPattern" value="true" /> <property name="ignoreComments" value="true" /> </module>	Regular expression to find and restrict use of System.out.print.* where logging should occur.
<module name="Regexp"> <property name="format" value=" printStackTrace" /> <property name="illegalPattern" value="true" /> <property name="ignoreComments" value="true" /> </module>	Regular expression to find and restrict use of printStackTrace.* where logging should occur.

<module name="RegexpSinglelineJava"> <property name="format" value="\t"/> <property name="message" value="Indentation should be performed with spaces only."/> <property name="ignoreComments" value="true"/> </module>	Indentation should be applied with 4 spaces per \t. This was accorded by our team to standardize the format when using different IDEs.

Checkstyle used by Oal:
http://slc15ekv.us.oracle.com:8080/static/oal-checkstyle.xml



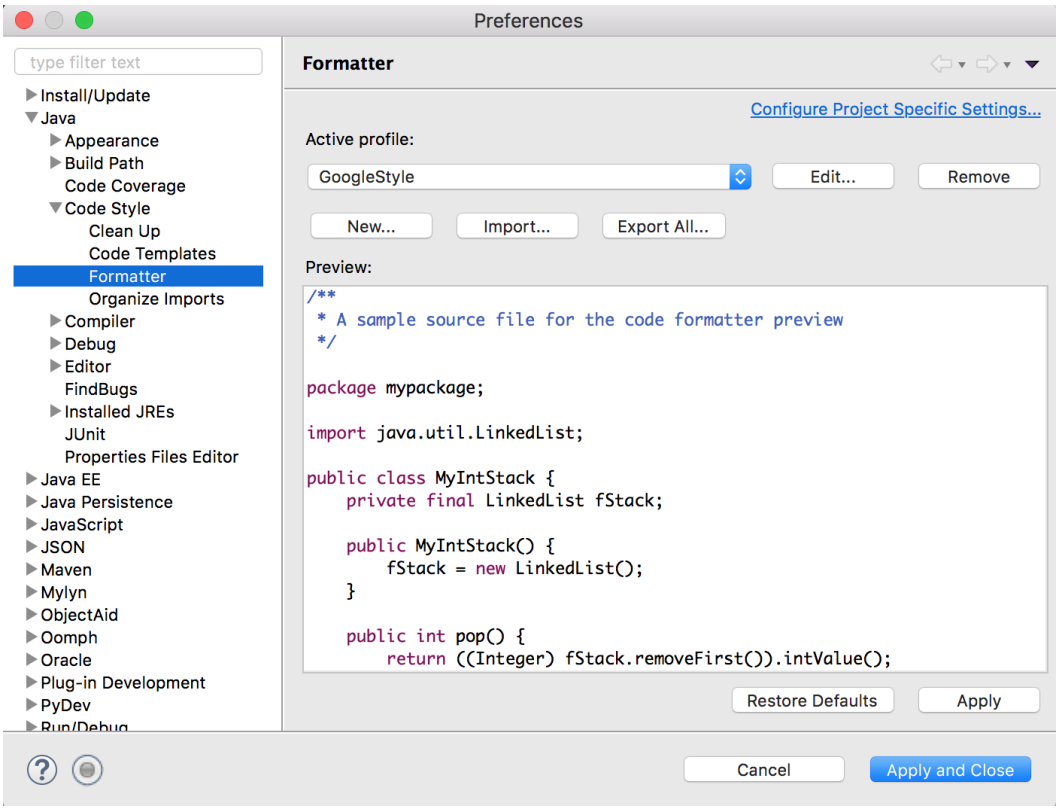
Note: The above server hosting Checkstyle rules will have to be added to Proxy exclusion list or Proxy may have to be disabled for the IDE to be able to use the Rules.

Sonar Custom Configuration (TBD)

Coding Standards

Formatting

We intend to use the Google Java Coding Style for formatting our code. The style ruleset has been updated for the lines to be wrapped at 120 columns instead of 100. Download the attached XML and import it using Eclipse/IntelliJ Preferences.



Updated the rule sets for the formatter and below are the updated links:
Intellij: http://slc15ekv.us.oracle.com:8080/static/oal-intellij-formatter-v2.xml
Eclipse: http://slc15ekv.us.oracle.com:8080/static/oal-eclipse-formatter-v2.xml

~~http://slc15ekv.us.oracle.com:8080/static/oal-eclipse-formatter.xml~~

Note: The above server hosting Formatter rules will have to be added to Proxy exclusion list or Proxy may have to be disabled for the IDE to be able to use the Rules.

Code Templates

TBD

Import Order

TBD

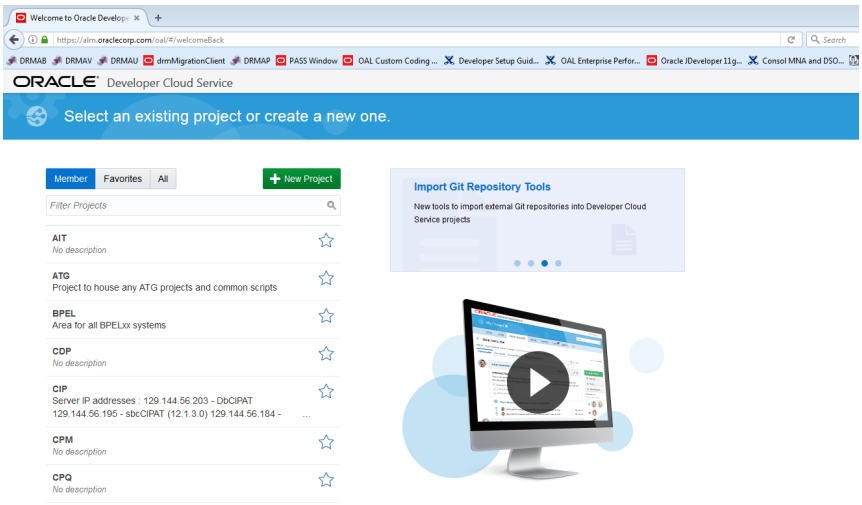
Cloning Repository From Oracle Developer Cloud Service (ODCS)

To setup a local copy of git repository from ODCS, perform following steps on your personal machine. Before you can clone repository, you would need following:

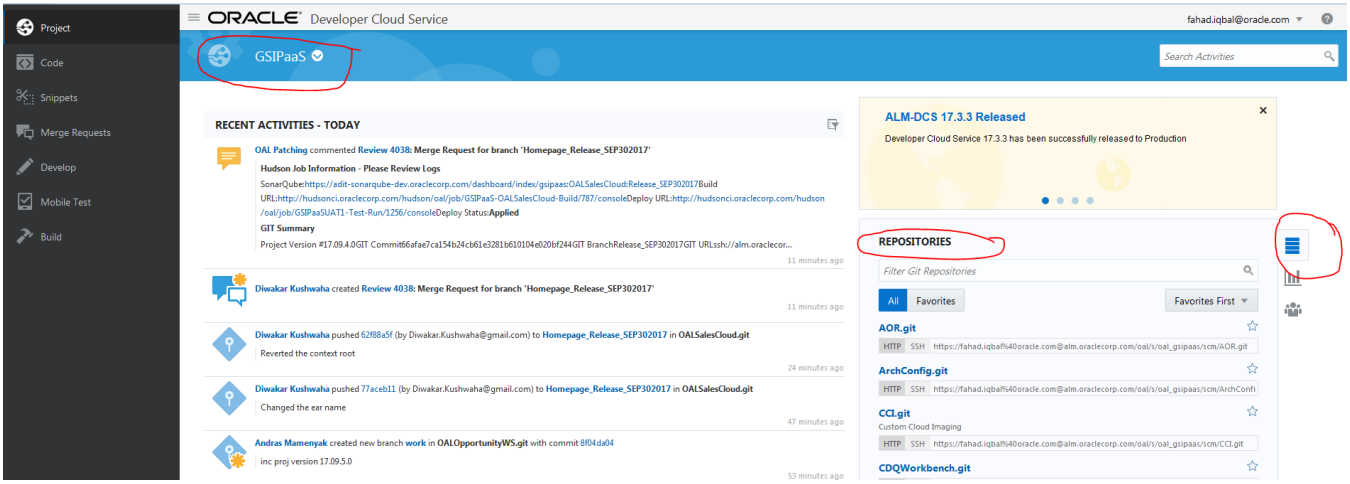
1. Git would need to be installed.
2. Access to ODCS. If you do not have access, then request for following from OIM:
    a. oal.DEVELOPER_USER (ALM (Production))
    b. oal.DEVELOPER_USER (ALM (Stage))

Find GIt Repository to Clone

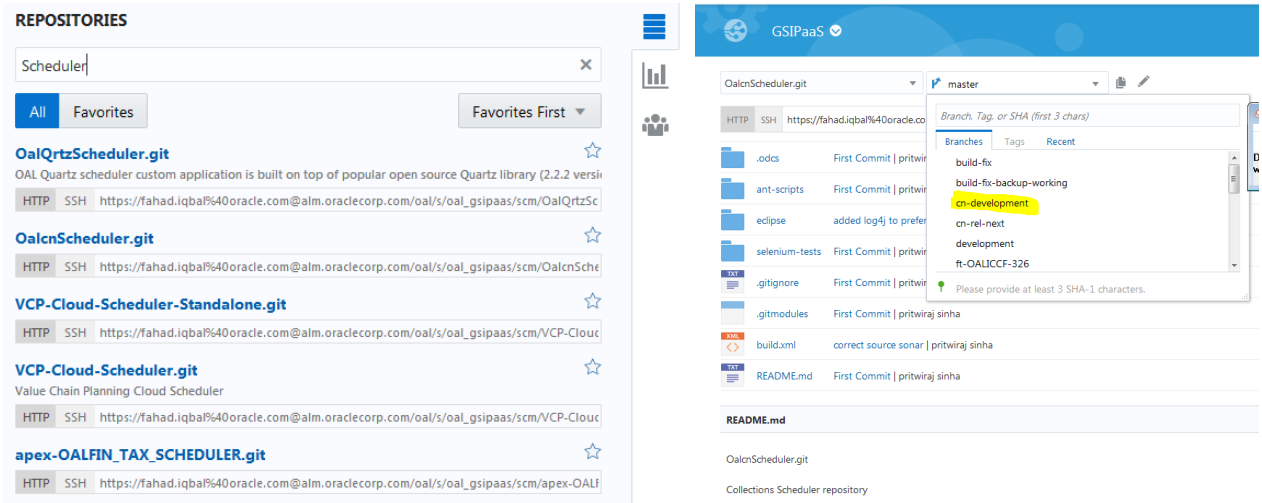
To find the link to git repository (a.k.a service) to clone, login to ODCS using your SSO.



Select relevant project from the list of projects available. This document uses GSIPaaS for demo.



Project page lists available repositories on the right pane of page. Use Repository Filter to search for desired repository and click on it to open description page.



Next page displays the components available in the selected repository. Select the HTTP link and copy. This link will be used to clone repository into local desktop. Note that SSH link can also be used for cloning but that will need generating SSH key. See more details for using SSH link here.

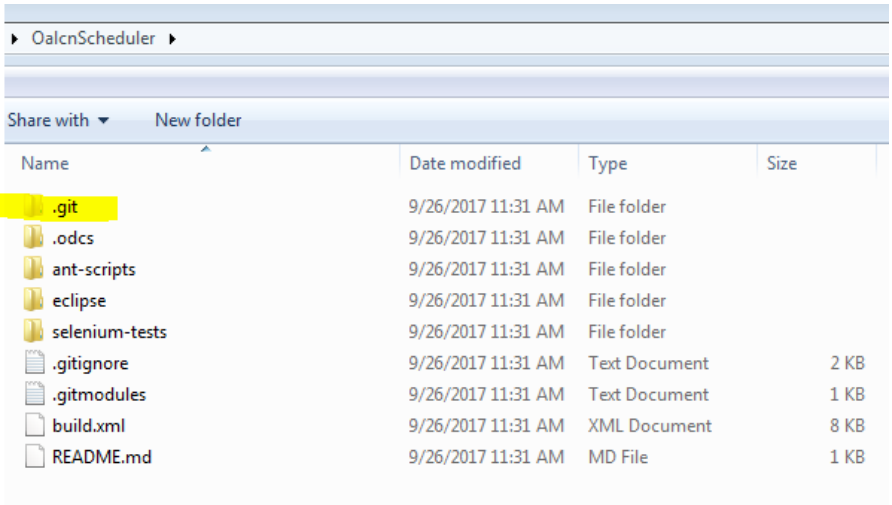
Clone Repository Using GIT

Start GIT Bash on your local desktop and navigate to the folder where you would like to clone the repository and execute following git command. Note "***-b***" is for branch, which in this case is *cn-development*.

```
git clone -b <branch> <remote_repo>
```

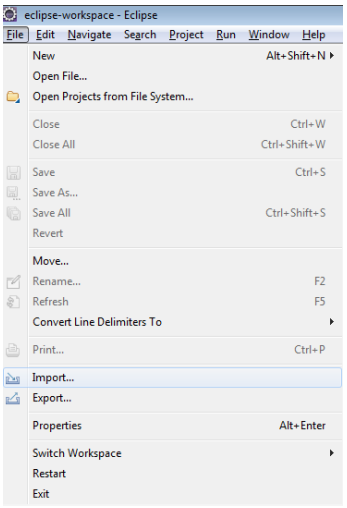
```
$ git clone -b cn-development https://fahad.iqbal%40oracle.com@alm.oraclecorp.com/oal/s/oal_gsipaas/scm/OalcnScheduler.git
Cloning into 'OalcnScheduler'...
remote: Counting objects: 1629, done
remote: Finding sources: 100% (2296/2296)
remote: Getting sizes: 100% (1452/1452)
remote: Compressing objects: 100% (4831/4831)
remote: Total 2296 (delta 692), reused 2294 (delta 691)
Receiving objects: 100% (2296/2296), 411.50 KiB | 0 bytes/s, done.
Resolving deltas: 100% (692/692), done.
```

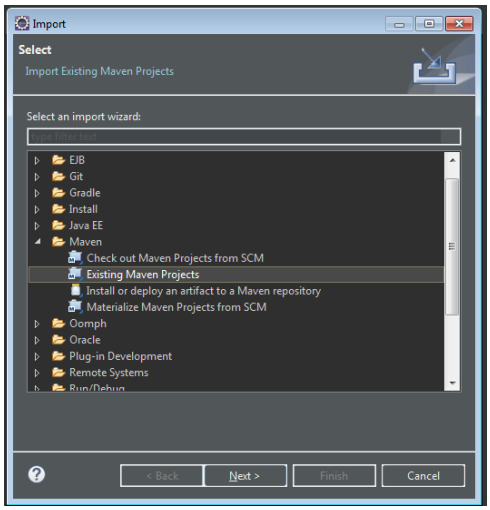
Once the command is successfully completed navigate to local folder to validate if the repository has been cloned. Note that *.git* folders are hidden folders (will need windows settings to be changed to make visible).



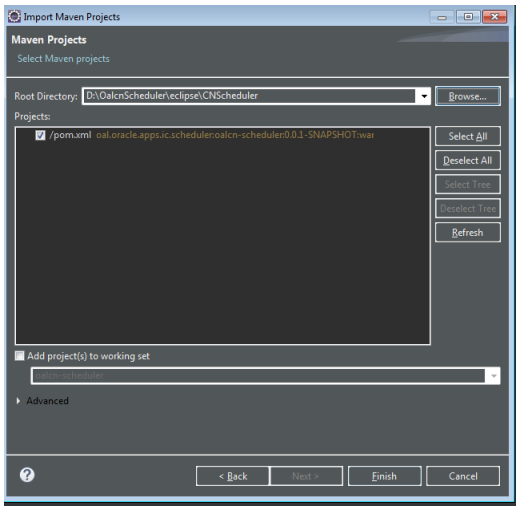
Importing Project into Eclipse

Once the repository have been cloned, import the project into eclipse as shown below.





Select Import from **eclipse > Edit > Import.** On next window select "**Existing Maven Project**"



Provide the path to project within the repository. The project is stored under "**eclipse**" folder within repository, as shown in screenshot and click **Finish**.