SAP NetWeaver
How-To Guide

How to set up a Stand-alone Mobiliser Development Environment

Windows Version

Applicable Releases:

Mobiliser 5.1

Simmaco Ferriero

Senior Specialist @ Rapid Innovation Group (RIG)

Version 1.11

December 2012

Document History

Document Version	Description
1.11	Moved some slides from chapter 4.2 to 4.1 because they were in awrong position.
1.10	Corrected some typos and added a page on Nexus
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Description
	Caution
	Important
	Note
	Recommendation or Tip
	Example

Table of Contents

1. Business Scenario 1

2. Background Information 1

2.1 What is …? 2

2.1.1 Apache Maven 2

2.1.2 Sonatype Nexus 2

2.1.3 CollabNet Subversion Edge 3

2.1.4 Jenkins 3

2.1.5 Eclipse 3

3. Prerequisites 4

4. Step-by-Step Procedure 7

4.1 Install Sonatype Nexus 7

4.2 Install CollabNet Subversion Edge 20

4.3 Upload the source code on Subversion 31

4.4 Install Apache Maven 35

4.5 Install Jenkins 42

4.6 Install Oracle Express XE 58

4.7 Install the Mobiliser database 66

4.8 Run Mobiliser 73

4.9 Install Eclipse 80

5. Appendix 97

1. Business Scenario

When a customer or a SAP Partner wants to enhance the Mobiliser source code, it's necessary to set up a development environment containing all the needed resources. In this guide, we will see how to set up a server containing all the needed components for development and how to configure the client part to start enhancing the code.

2. Background Information

A lot of technologies will be involved in this walkthrough: we will talk about Apache Maven, Sonatype Nexus, Subversion, Jenkins and Eclipse. It would be appreciated if the reader of this document would already have knowledge about these components: it would definitely speed up the execution of all the steps. However, we will try to give some very basic information about each one of the components involved, mainly we will look into what they do and how they integrate with other components.

For our scope, we are going to configure a virtual machine equipped with:

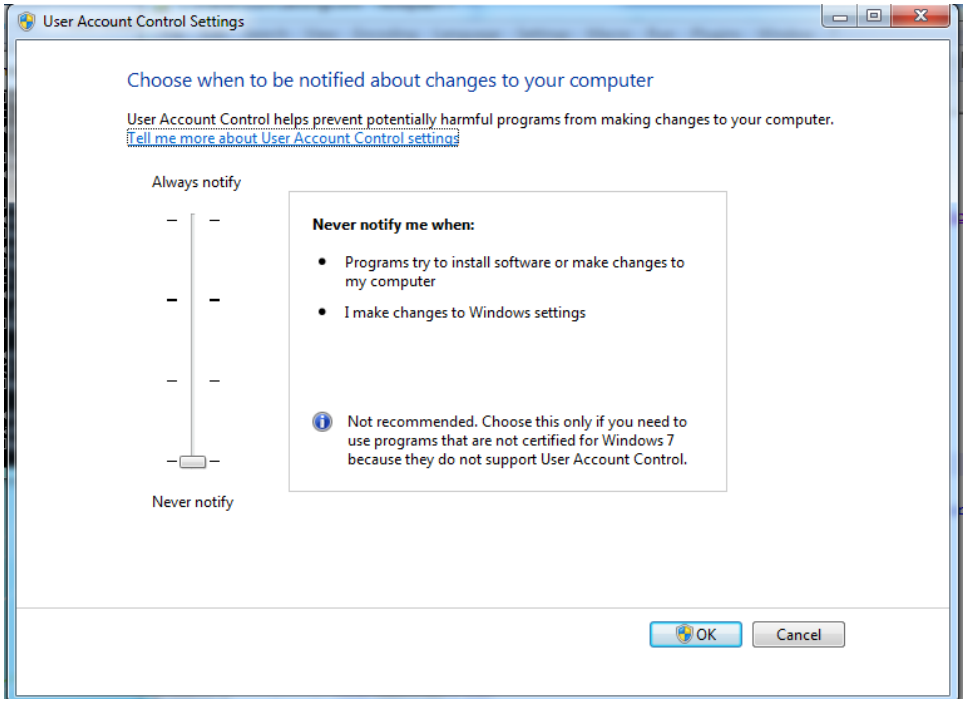
- Windows 7 Ultimate 64bit
- At least 2 GB of memory (4 GB recommended)
- 50 GB of free space on the disk

The same steps can be done on a 32bit system, paying attention to download the software specific for that version.

Note

On our system, we have disabled the UAC (User Access Control) settings by decreasing the security level. This will allow us to avoid running certain programs or commands by requesting administrator's rights. Therefore, throughout this guide you will not see any "Run as administrator…" type of command. Of course, this is not mandatory.

In case you want to do the same, you can lower the UAC security level by accessing this screen:



2.1 What is …?

Before starting, here's a short introduction to all the main software components we are covering in this paper.

2.1.1 Apache Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven uses an XML file to describe the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain, well-defined tasks such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository. This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated. This is an example of POM.XML file:

```
<project>
  <!-- model version is always 4.0.0 for Maven 2.x POMs -->
  <modelVersion>4.0.0</modelVersion>

  <!-- project coordinates, i.e. a group of values which
       uniquely identify this project -->

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>

  <!-- library dependencies -->

  <dependencies>
    <dependency>

      <!-- coordinates of the required library -->

      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>

      <!-- this dependency is only used for running and compiling tests -->

      <scope>test</scope>

    </dependency>
  </dependencies>
</project>
```

When you run the command "mvn package" in the same folder where this POM is, the package is build and all the dependencies are downloaded from the repositories and linked to the project.

2.1.2 Sonatype Nexus

Nexus is a repository manager. It can handle a lot of repositories providing one single entry point to Maven for all the components it needs when building packages. You can use it for storing your own artifacts or for caching external artifacts.

An artifact is essentially a .jar file that can be stored and downloaded from a repository and included in your own project as an external component. A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR. Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. These three together (group ID, artifact ID and the version string) uniquely identify the artifact. All project's dependencies are specified as artifacts.

2.1.3 CollabNet Subversion Edge

Subversion is an open source version control system. Founded in 2000 by CollabNet, Inc., the Subversion project and software have seen incredible success over the past decade. Subversion has enjoyed and continues to enjoy widespread adoption in both the open source arena and the corporate world. There are many distributions of it, but we will use the one from CollabNet called Subversion Edge. This particular version includes a web server and a user interface so it's relatively easy to configure and to manage. Other distributions are mainly based on command line tools.

2.1.4 Jenkins

Jenkins is an open source integration tool written in Java. The project was derived from Hudson. Jenkins provides continuous integration services for software development mainly in Java. It's a server-based system running on Apache Tomcat. It supports a lot of version control systems and among these there is the one we have chosen, CollabNet Subversion Edge. It allows you to build software package automatically, by scheduling builds with any frequency and collecting the results.

2.1.5 Eclipse

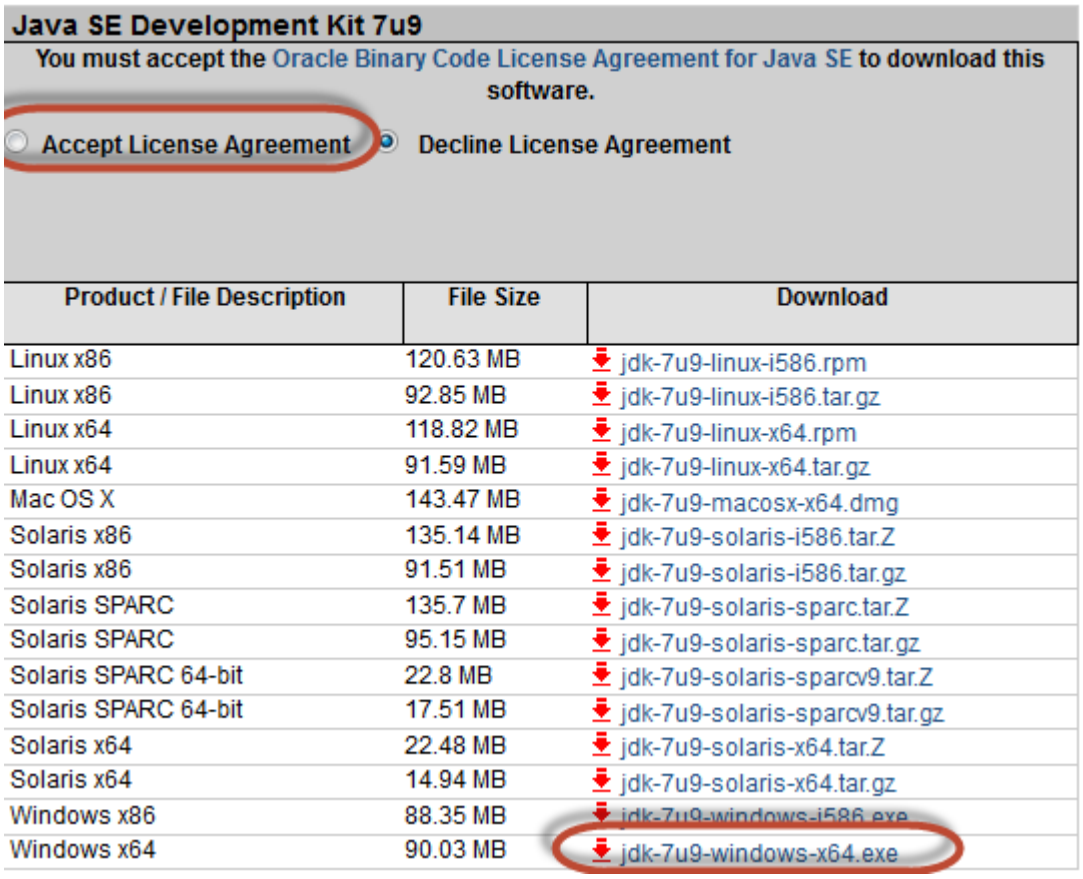
Eclipse is an integrated development environment (IDE) that allows developer to create their applications with many programming languages like Java, Python and so on. It's particularly used for Java and Android development. There are many distributions of it. For the purpose of this paper, we are using the Juno distribution, but even Indigo is acceptable.

3. Prerequisites

Here it is a list of all the software components and settings that are needed throughout this guide.

- **Java Development Kit (JDK) 7 64bit** – it can be downloaded from the following web site: http://www.oracle.com/technetwork/java/javase/downloads/index.html. We can download the latest version of JDK by clicking on the following button:

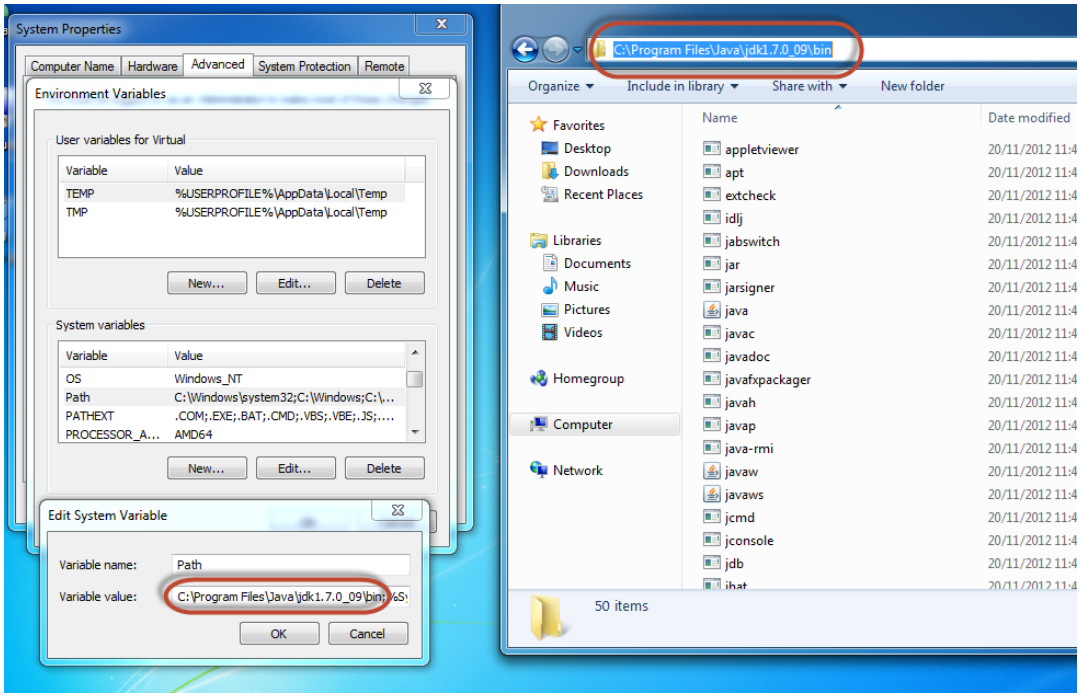
After accepting the License Agreement, you can download the required version. In our case, we are going to download the 64 bit version for Windows.



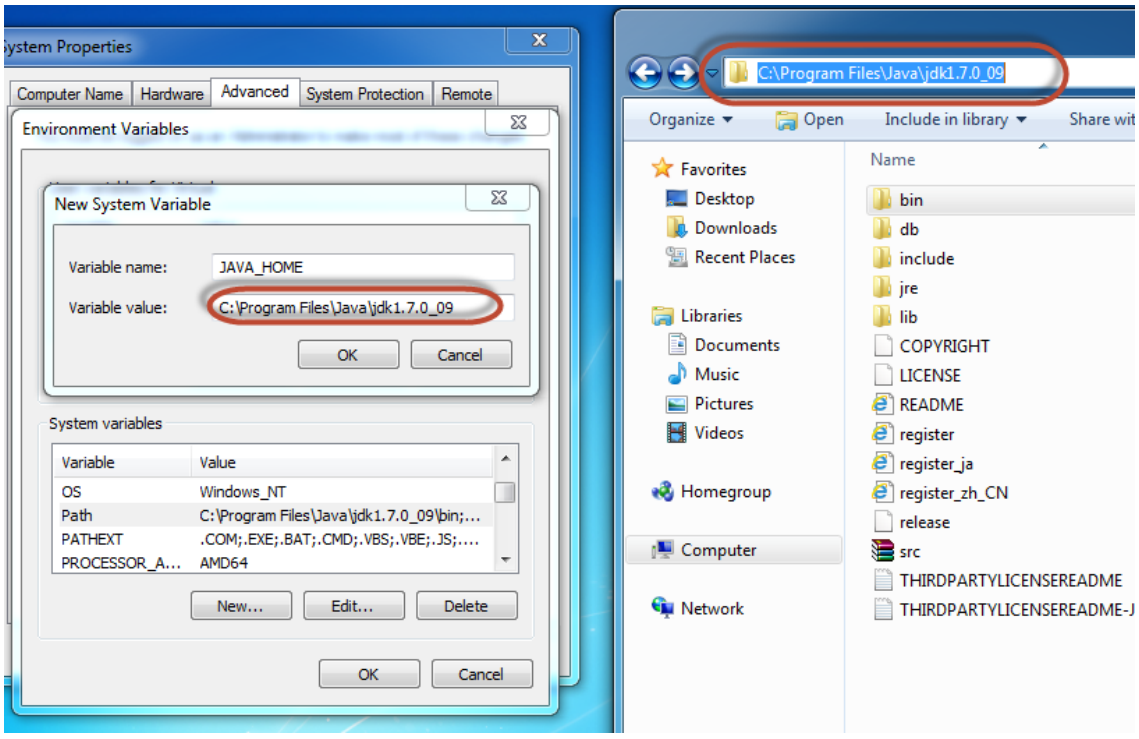
The installation of Java is quite straightforward. After downloading the software, you can install it using the default settings.

- **Some new environment variables** need to be already in place before starting this guide. Specifically, we have to:
    - Extend the system **PATH** variable with the JDK folder
    - Create a new variable named **JAVA_HOME** to point to the folder where Java is installed

In order to change this, you have to go on **Control Panel → System and Security → System**, open the Advanced system settings, click on the **Environment variable** button and edit the **PATH** variable by adding the path to the BIN folder in your Java installation just at the beginning of this variable, as depicted in the following picture:



Then you need to create a new variable called **JAVA_HOME** that points to the superior folder shown in the previous screen, as follows



- **A valid account to the Sybase Nexus repository is needed**: https://repo.paybox.net/nexus/index.html#welcome

- **The complete Mobiliser source code**

- **Notepad++** - it can be downloaded from the following web site: http://notepad-plus-plus.org/
  It will be used for editing the XML files.

In this guide we will use the following naming convention:

Variable	Description	Value in this document
WCMD	Windows Command Shell	Run it as Start → Run → Cmd
<nexus_folder>	The folder where Nexus is installed	C:\nexus-2.2-01
<server_name>	The name of the machine where you install the software.	virtual-pc
<repository_name>	The name of the repository inside Subversion	custom
<windows_user_name>	The name of the Windows user who logs in and installs the software	Virtual
<windows_user_password>	The password of the Windows user who logs in and installs the software	Password
<local_user_path>	The Windows path for the current user profile	C:\Users\virtual

4. Step-by-Step Procedure

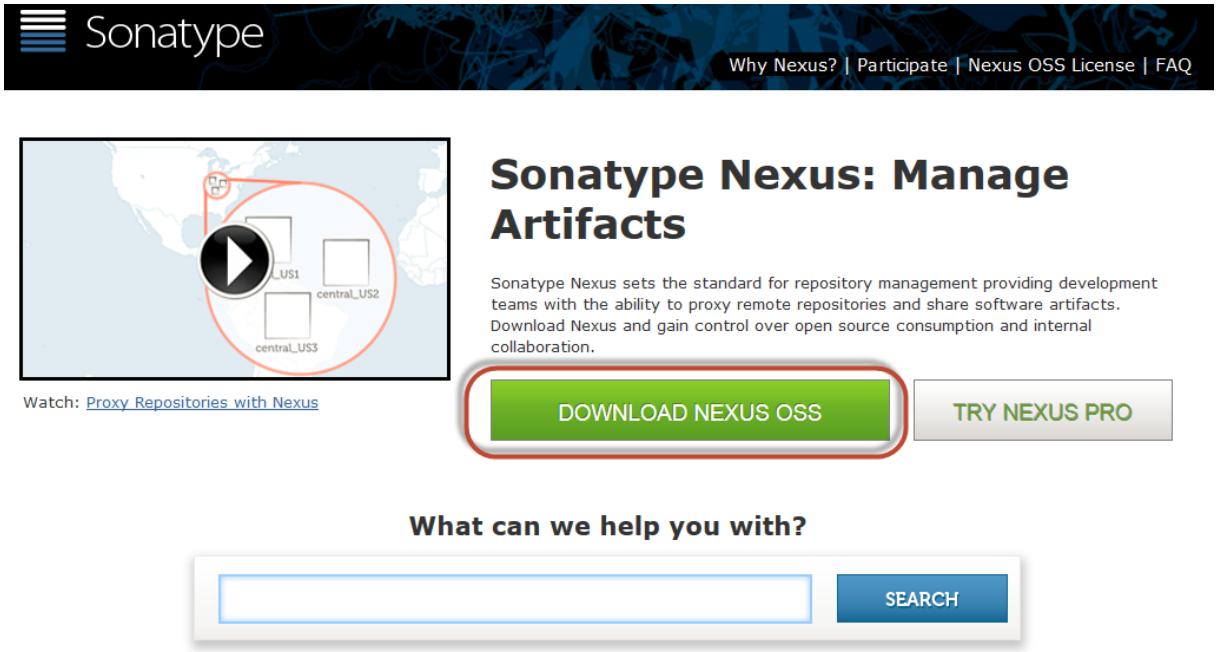
Our installation will be made by the following steps:

1) Installing Nexus
2) Installing CollabNet Subversion Edge
3) Uploading the source code to Subversion
4) Installing Apache Maven
5) Installing Jenkins
6) Installing Oracle Express XE
7) Creating and installing the Mobiliser database
8) Running Mobiliser
9) Installing Eclipse

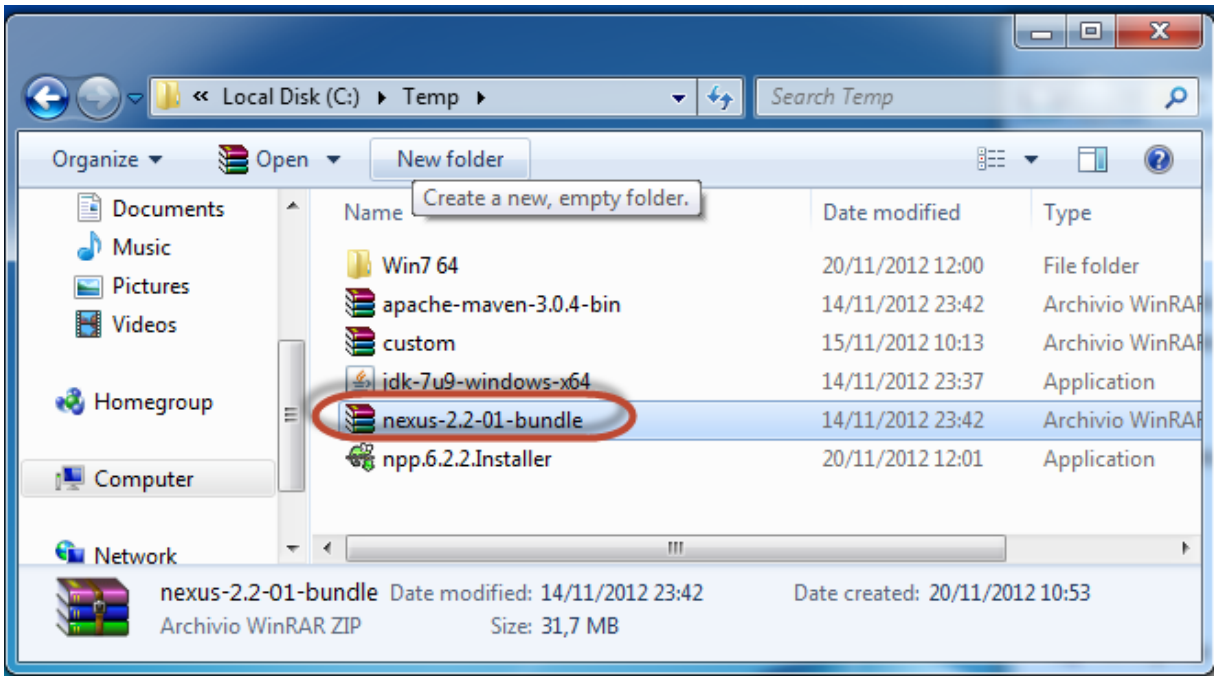
4.1 Install Sonatype Nexus

1. Sonatype Nexus is a repository manager for Maven artifacts. It can be downloaded from the following web site: http://www.sonatype.org/nexus.

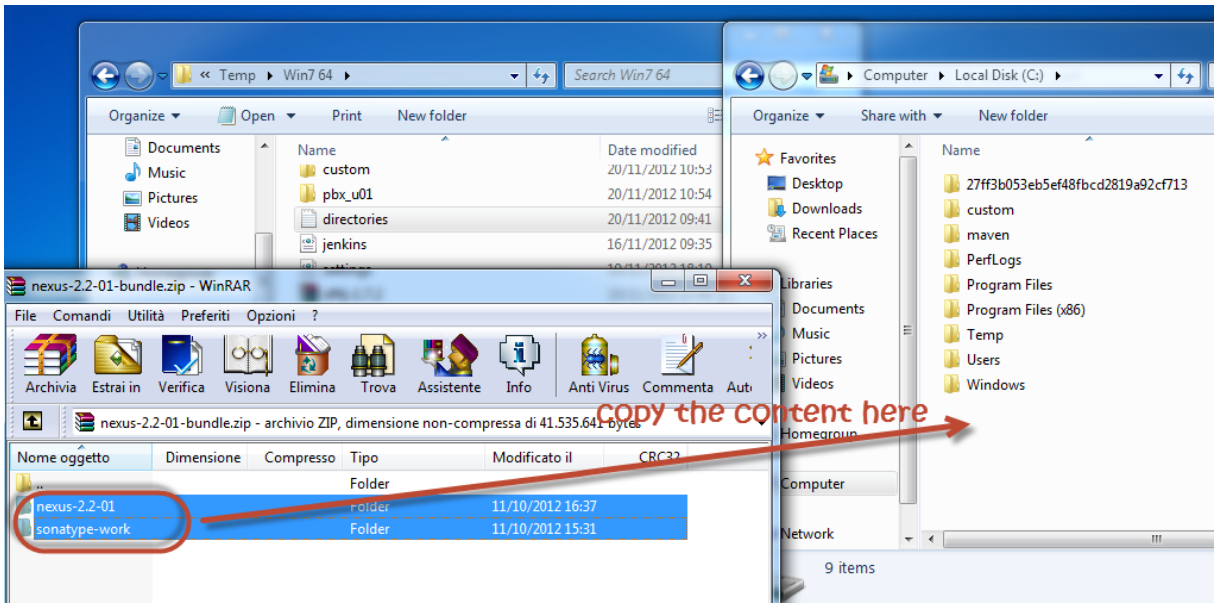
   Download it by clicking on **Download Nexus OSS**.



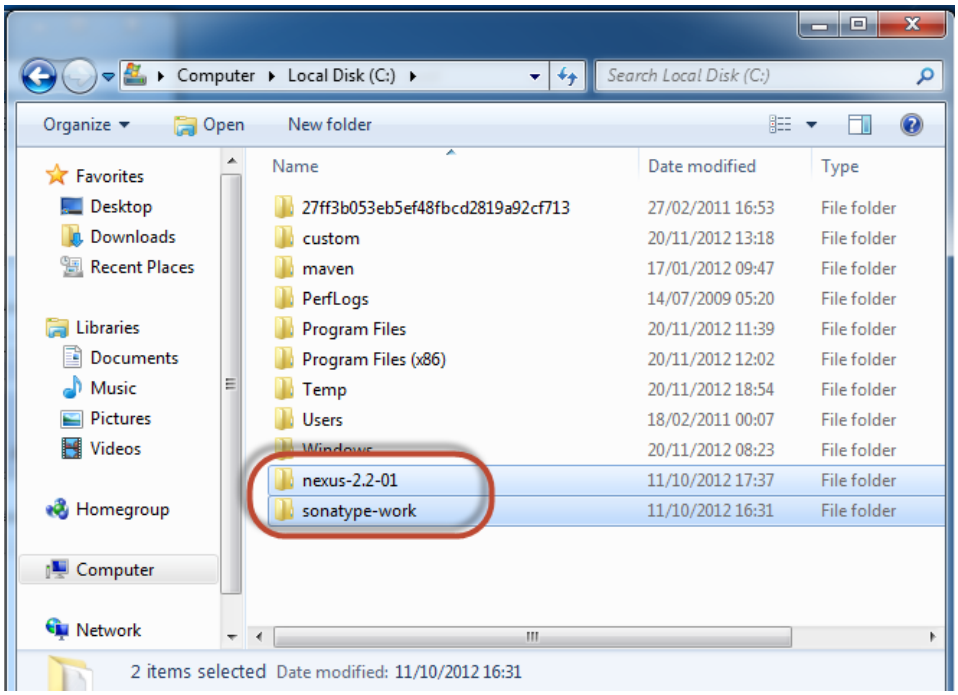
2. After downloading the file you can start the installation, by unpacking the downloaded file.



3. For ease of use, we can put the content of this package directly on the C: drive. You will have to copy 2 folders.

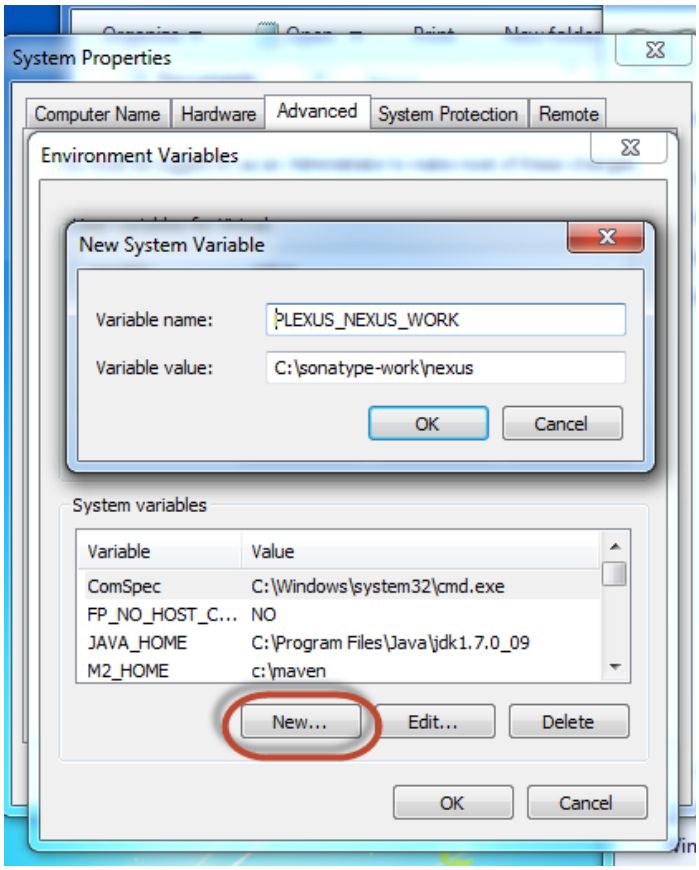


4. After you have extracted the 2 folders, we need to set up a variable inside the system



5. The variable to setup is the following:

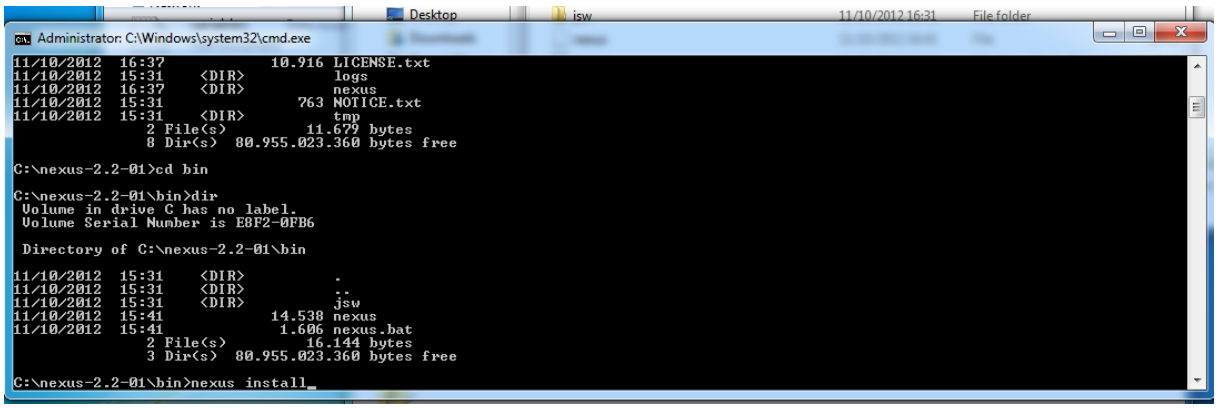
PLEXUS_NEXUS_WORK=C:\sonatype-work\nexus



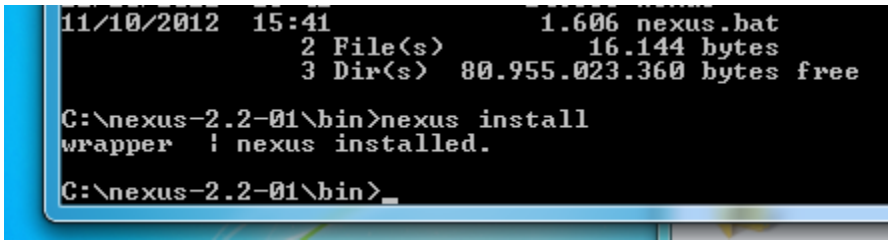
6. Now open a WCMD and go in the <nexus_folder>\bin with the command CD <nexus_folder>\bin. Once inside this folder, run the following command:

   nexus install

   This will install Nexus as a service.

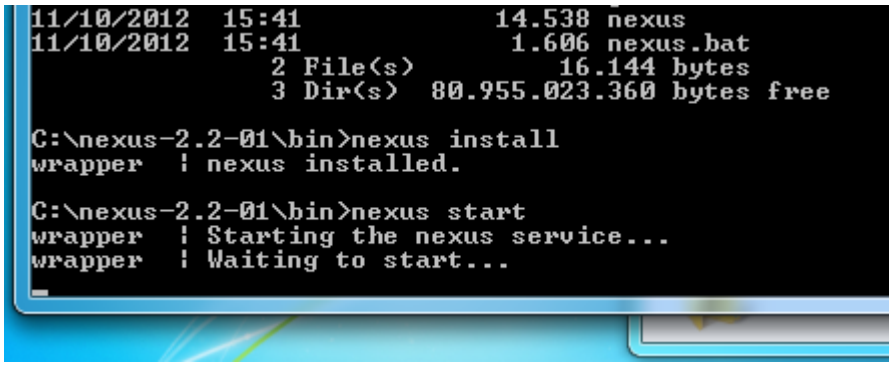


7. If the Nexus service has been successfully installed, the message shown will be same as below.

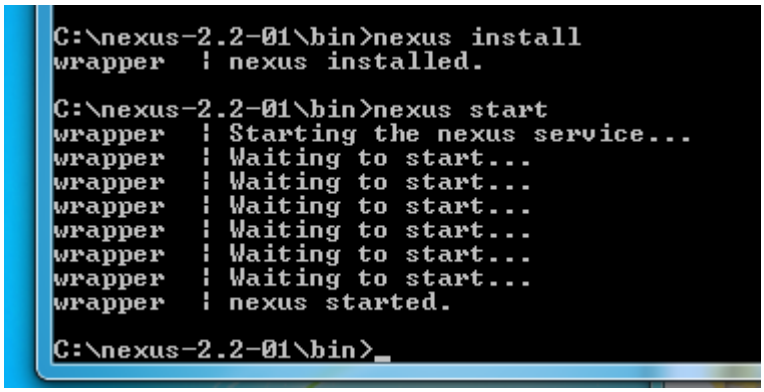


8. You can now start it using the following command:

   nexus start



9. After a few seconds you will receive the notification that the service is started successfully.

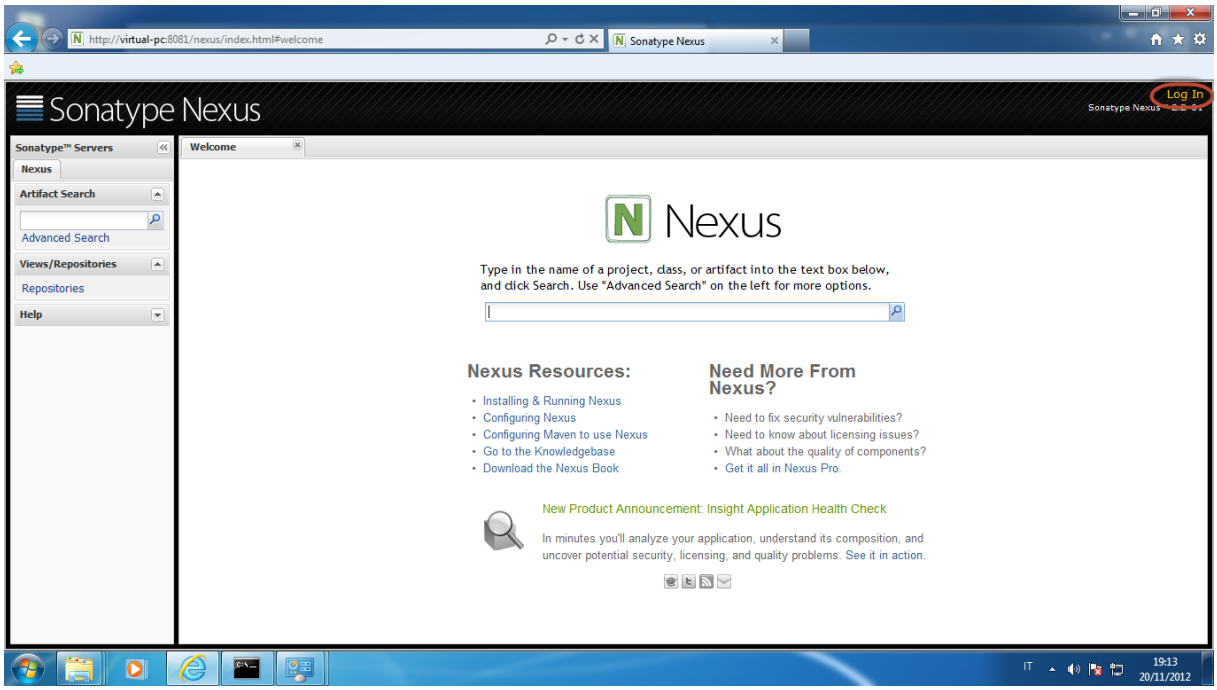


10. In order to access the new service you will need to open the Internet browser and type the following URL:

    http://<server_name>:8081/nexus

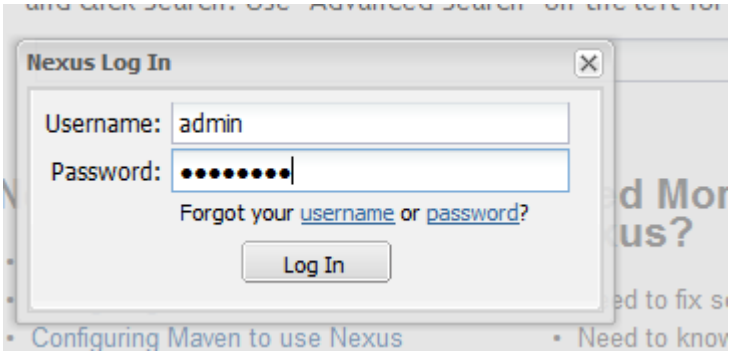
    By default the Nexus service will be installed on the port 8081. It can be changed, but for the scope of this document we will leave it as it is by default.

    The main page of Nexus is displayed; click on the **Log in** link on the top right corner of the screen. We will go inside Nexus in order to configure the system.

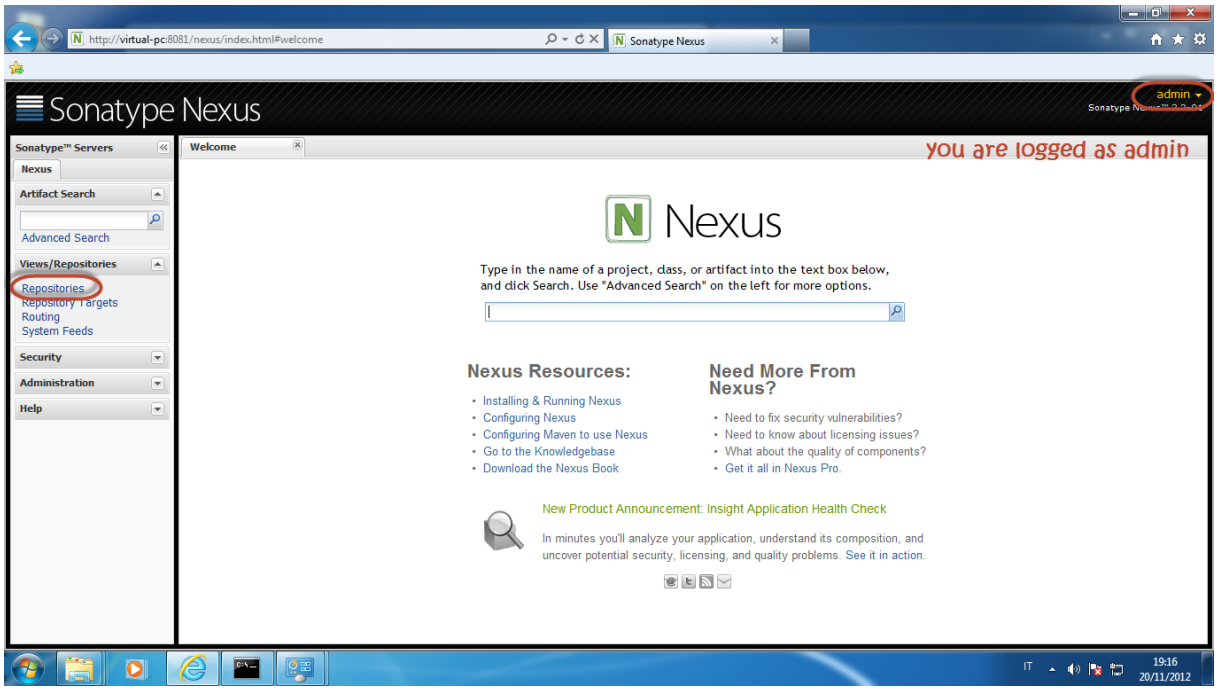


11. You will be requested to provide Nexus with a valid account. The default administrator account is:

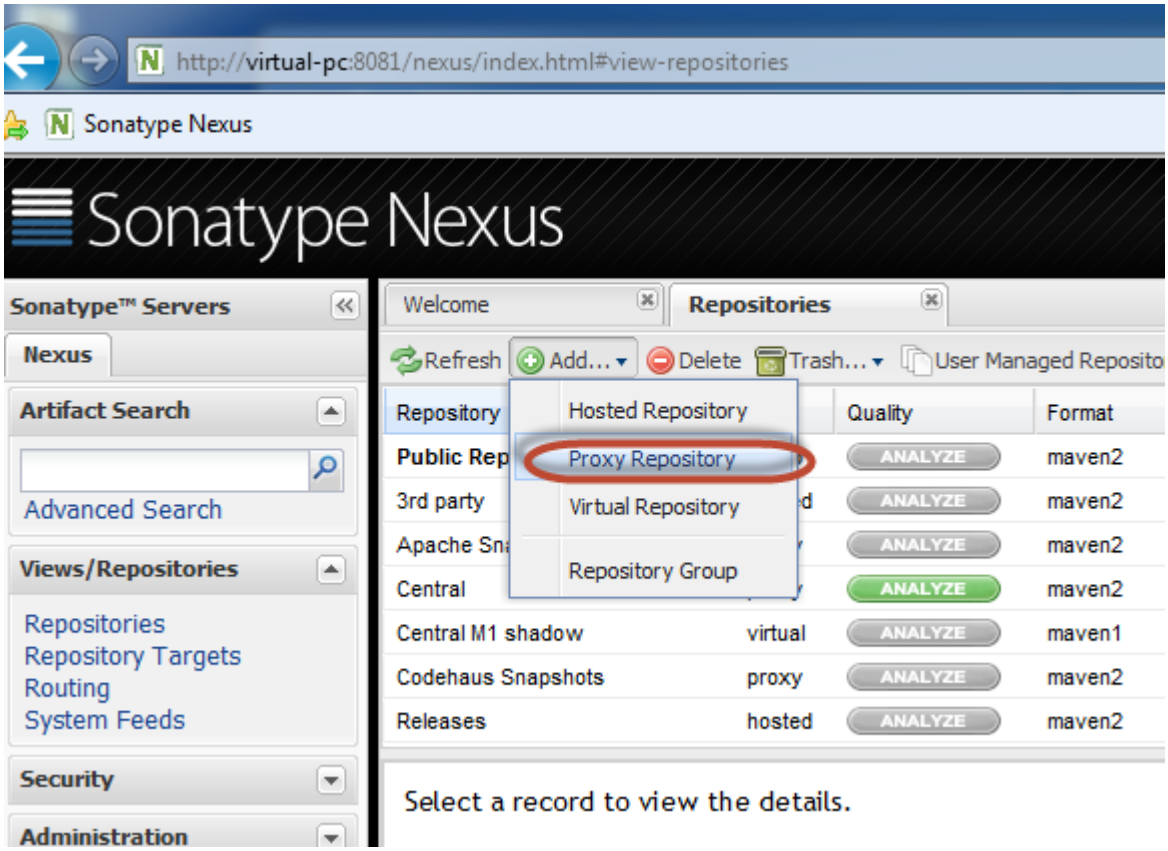
USER	PASSWORD
admin	admin123



12. After the login, click on the **Repositories** page:



13. We need to add a couple of new proxy repositories to connect with the Sybase central repository. Click on **Add → Proxy Repository**

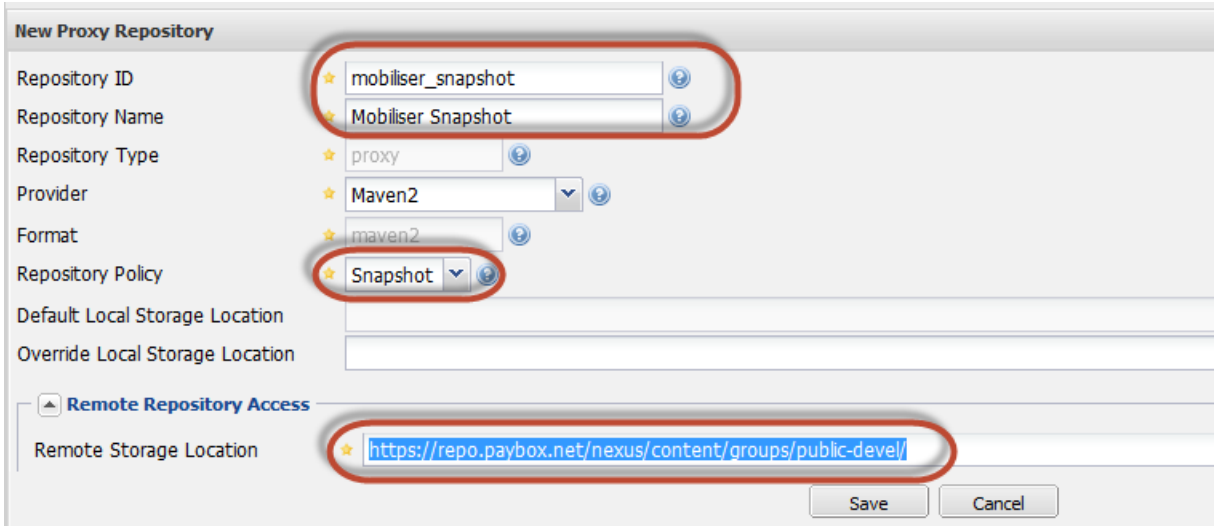


14. The URL address for the Sybase central repository is:

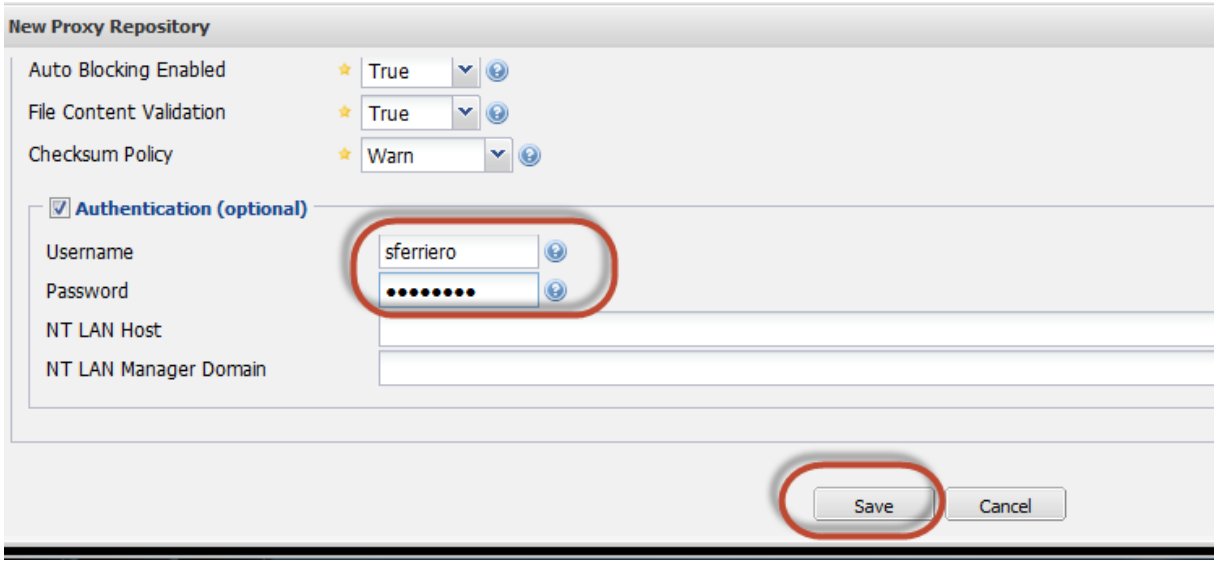
    https://repo.paybox.net/nexus/content/groups/public-devel/

    Type the following information:

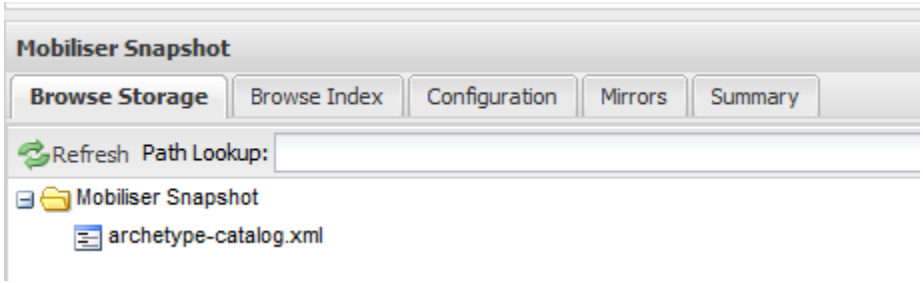
Field	Value
Repository ID	mobiliser_snapshot
Repository Name	Mobiliser Snapshot
Provider	Maven2
Repository Policy	Snapshot
Remote Storage Location	https://repo.paybox.net/nexus/content/groups/public-devel/



15. Enable the authentication and type the username and the password you used to register to the Sybase Central repository site, then click on **Save**.

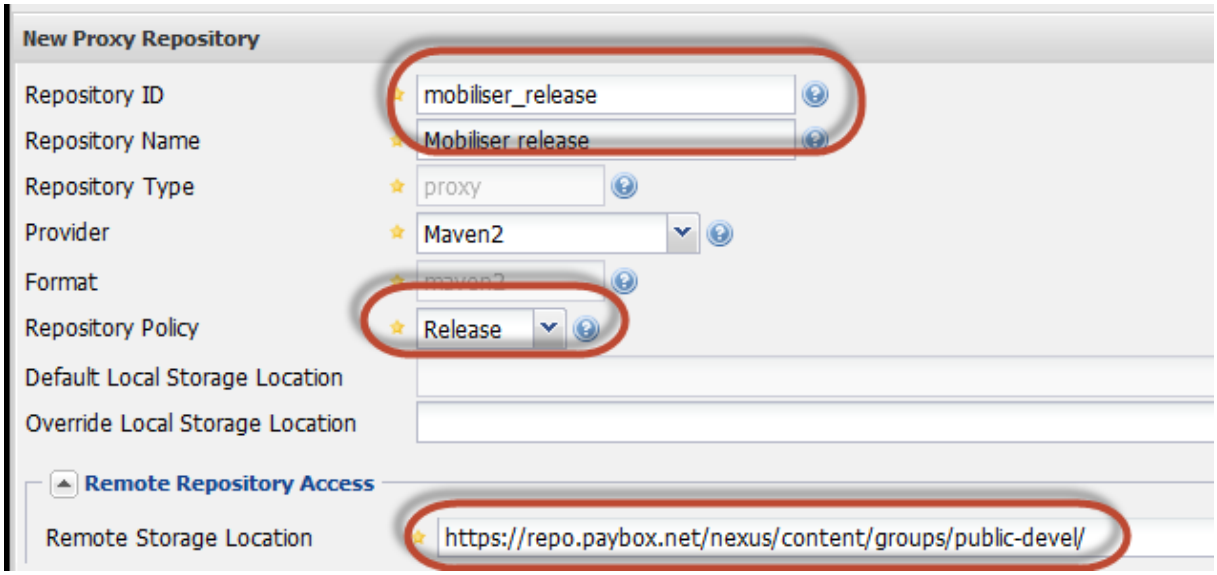


16. You should be able to see the following content in the **Browse Storage** tab.



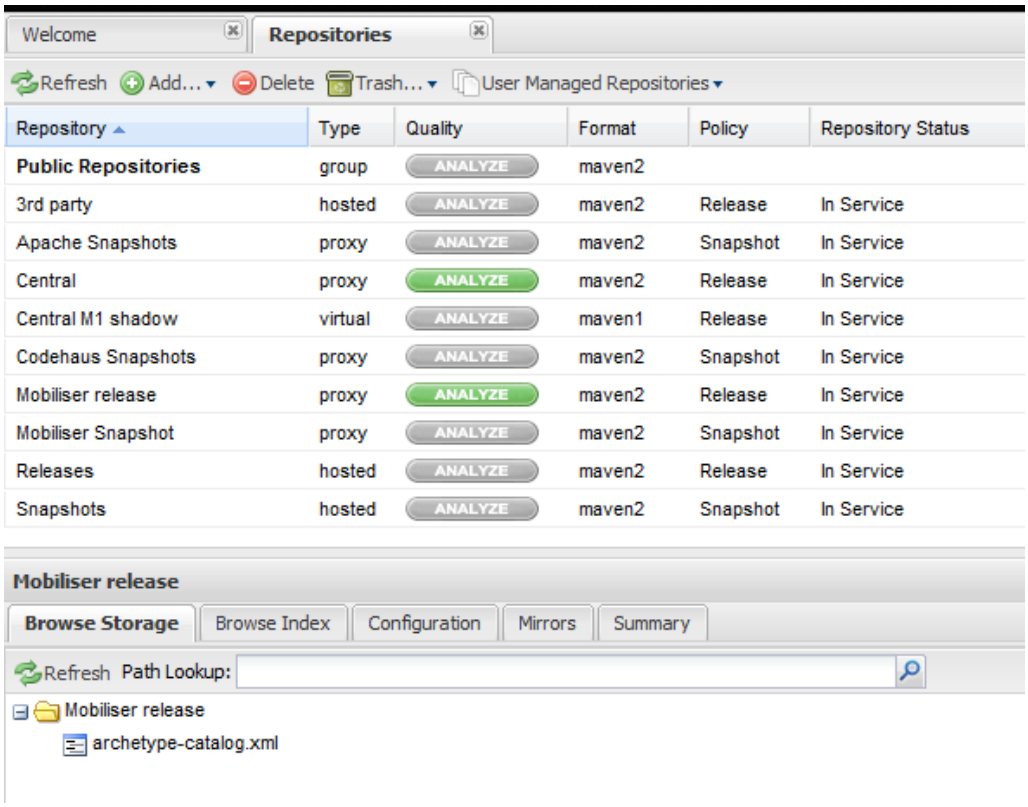
17. We have created a proxy for the snapshot and now we need to create the same for the release. Create a new proxy repository and fill in the following details:

Field	Value
Repository ID	mobiliser_release
Repository Name	Mobiliser Release
Provider	Maven2
Repository Policy	Release
Remote Storage Location	https://repo.paybox.net/nexus/content/groups/public-devel/

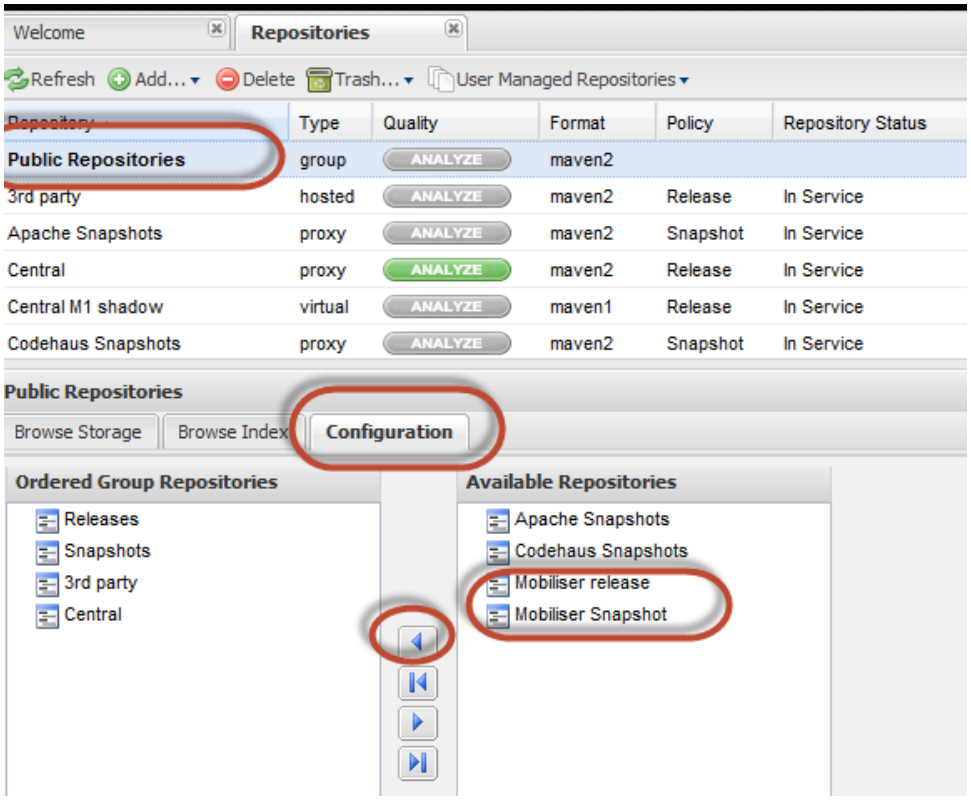


18. Enable the authentication and type the username and the password you used to register to the Sybase Nexus site, then click on **Save**.

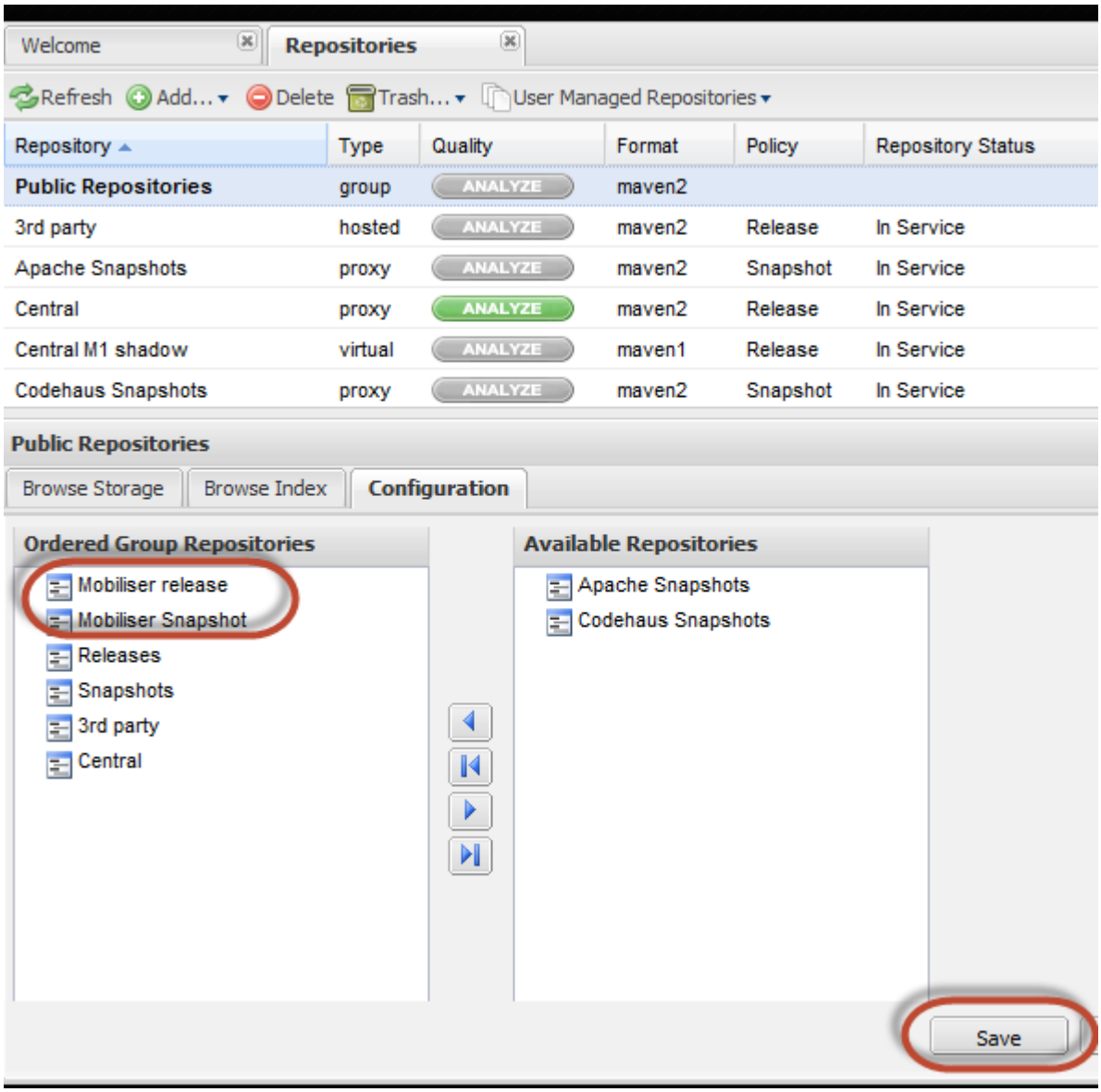
19. At the end you should have the two new repositories in the list.



20. Click on the **Public Repository** row and then on the **Configuration** tab. You will need to add the two new repositories to the public group:



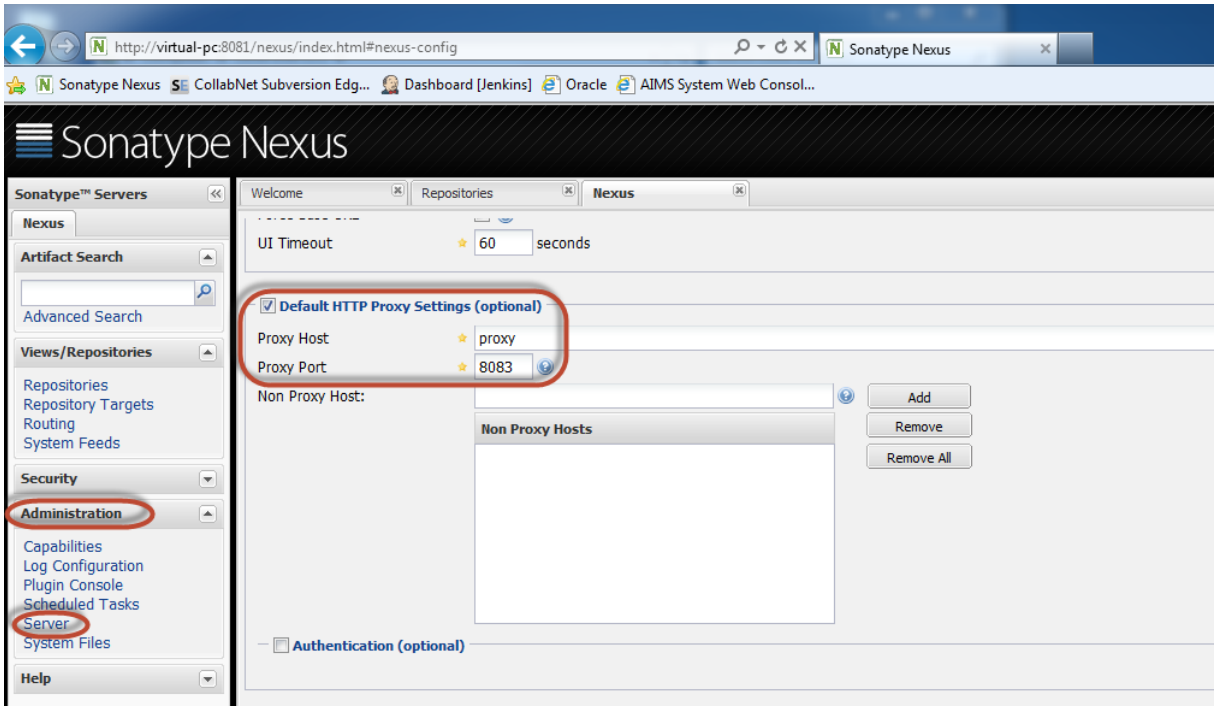
21. Put them on top of the list and DON'T FORGET to click on **Save**.



22. Please pay attention that if you are inside the SAP Network you may need to setup some further parameters inside the Server settings. In particular you need to define the proxy server to go out on the external network. Please proceed as follows. Go on **Administration → Server** and set the **Default HTTP Proxy Settings** with

Proxy Host	Proxy Port
proxy	8083

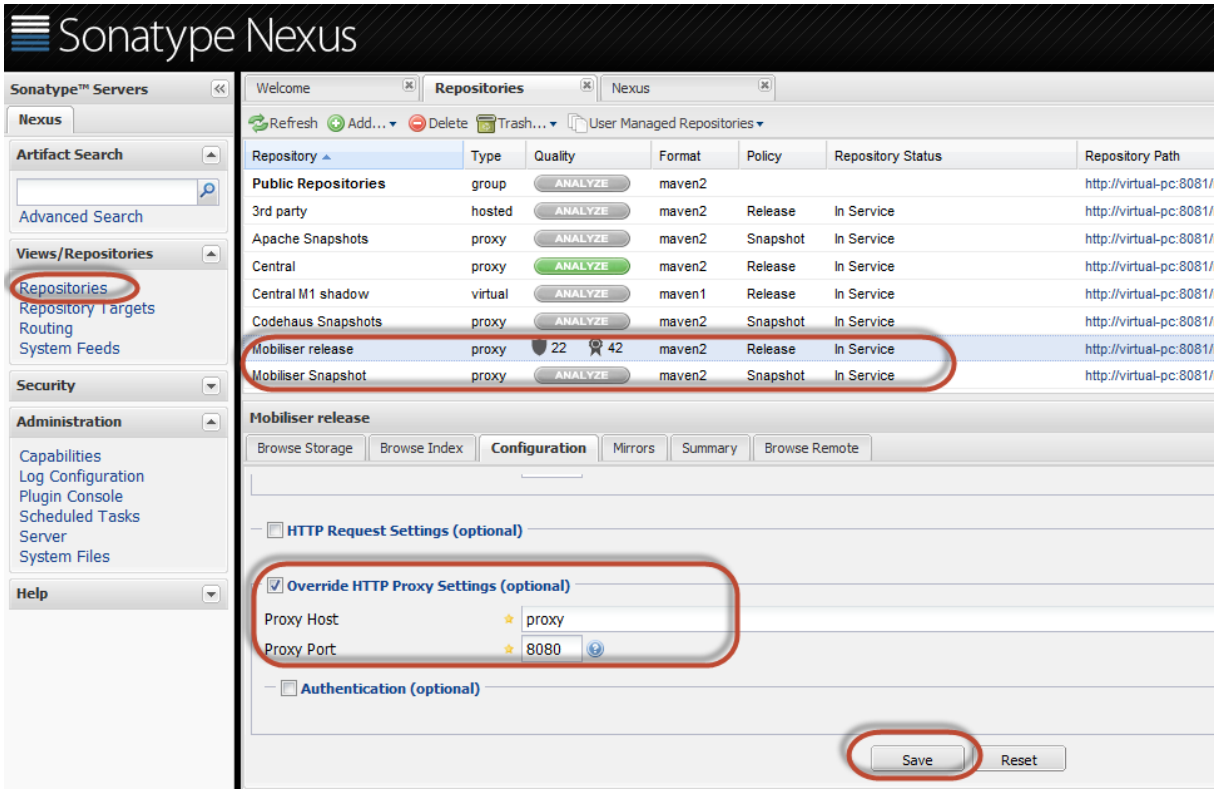
Then click on **Save**.



23. Now for each one of the two Mobiliser proxies we have set, change the **Override HTTP Proxy Settings** with the following values:

Proxy Host	Proxy Port
proxy	8080

Then click on **Save**.

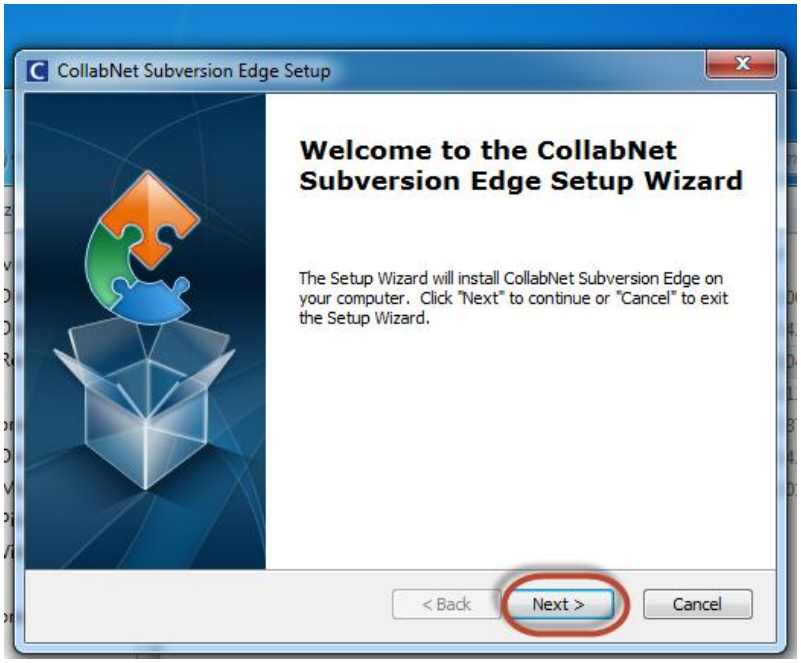


24. Just for your comfort, you may want to add the Nexus Web Site to the Favorites Bar of the browser.

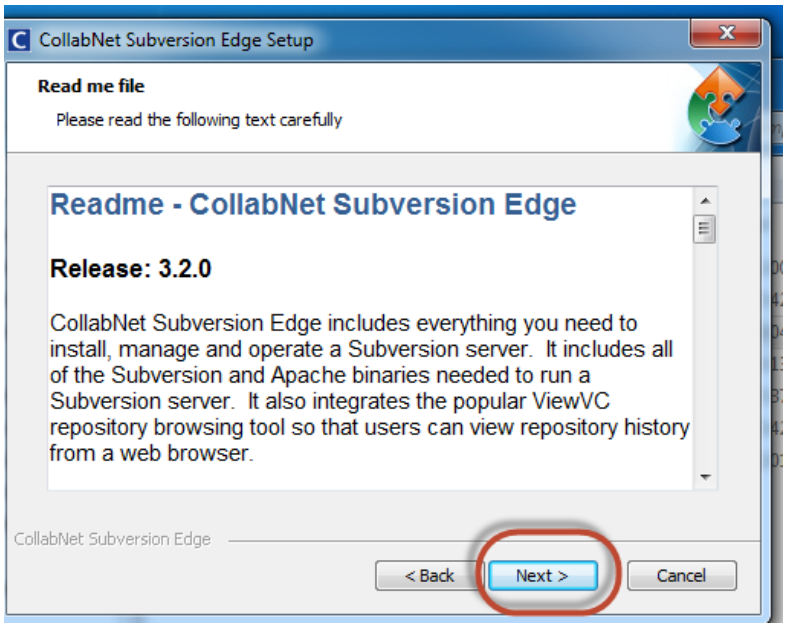
4.2 Install CollabNet Subversion Edge

1. This source management tool can be downloaded from the following web site: http://www.collab.net/downloads/subversion. For our test, since we are on a 64 bit system, we will download the 64 bit version of the software.

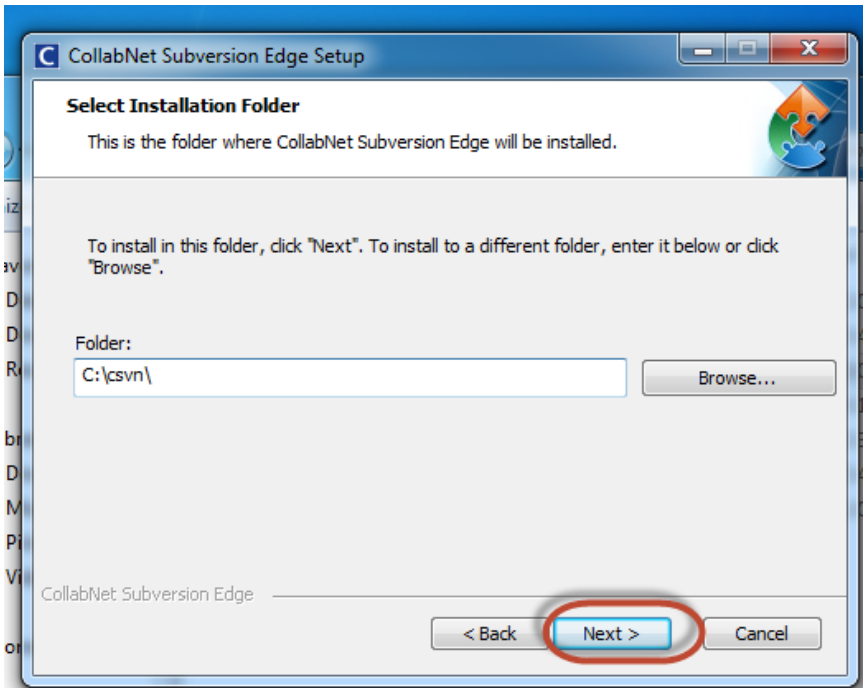
2. In order to download the software you will need to register, which can be done without charge. Once you have registered and downloaded, you can install it on your server. Start the installation and click on **Next**.



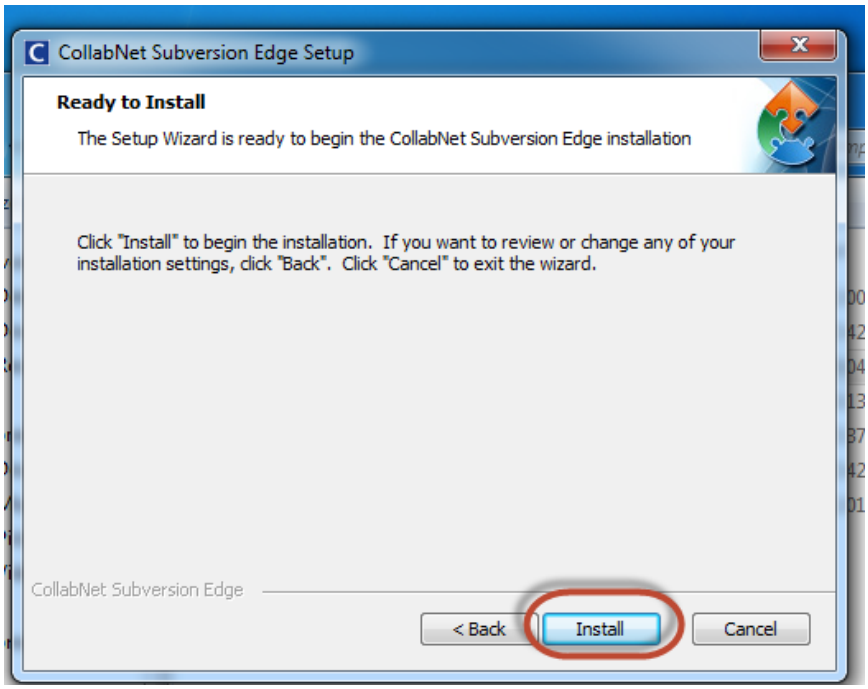
3. Click on **Next**.



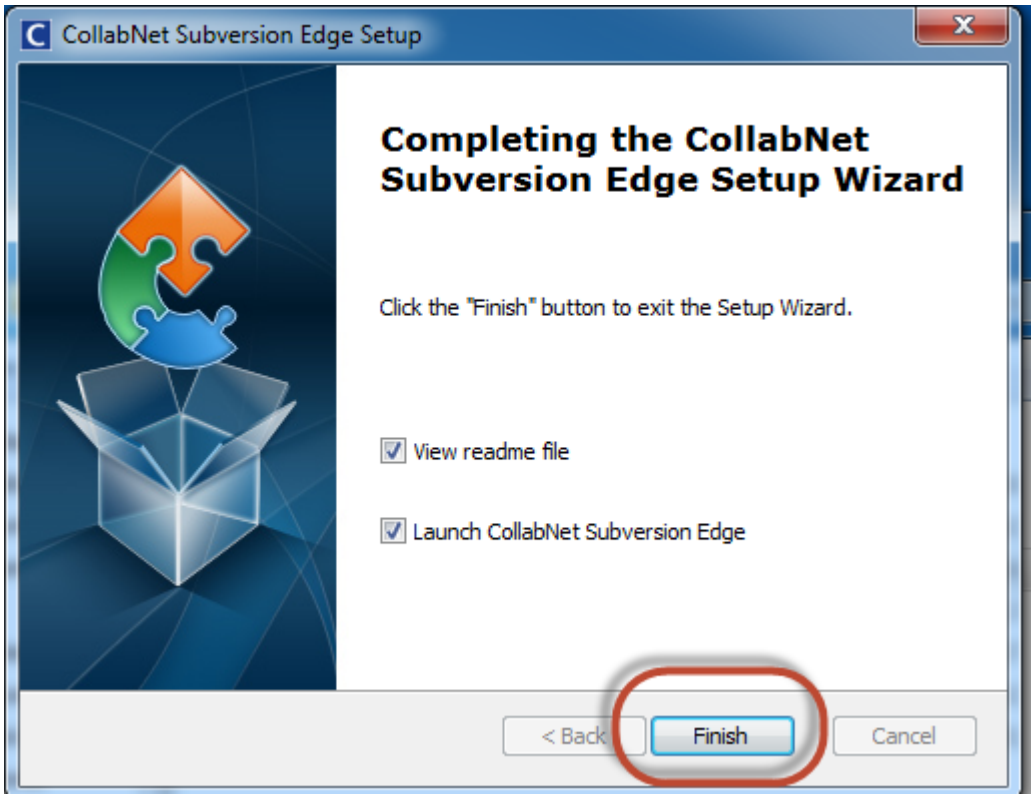
4. Use the proposed installation path and click on **Next**.



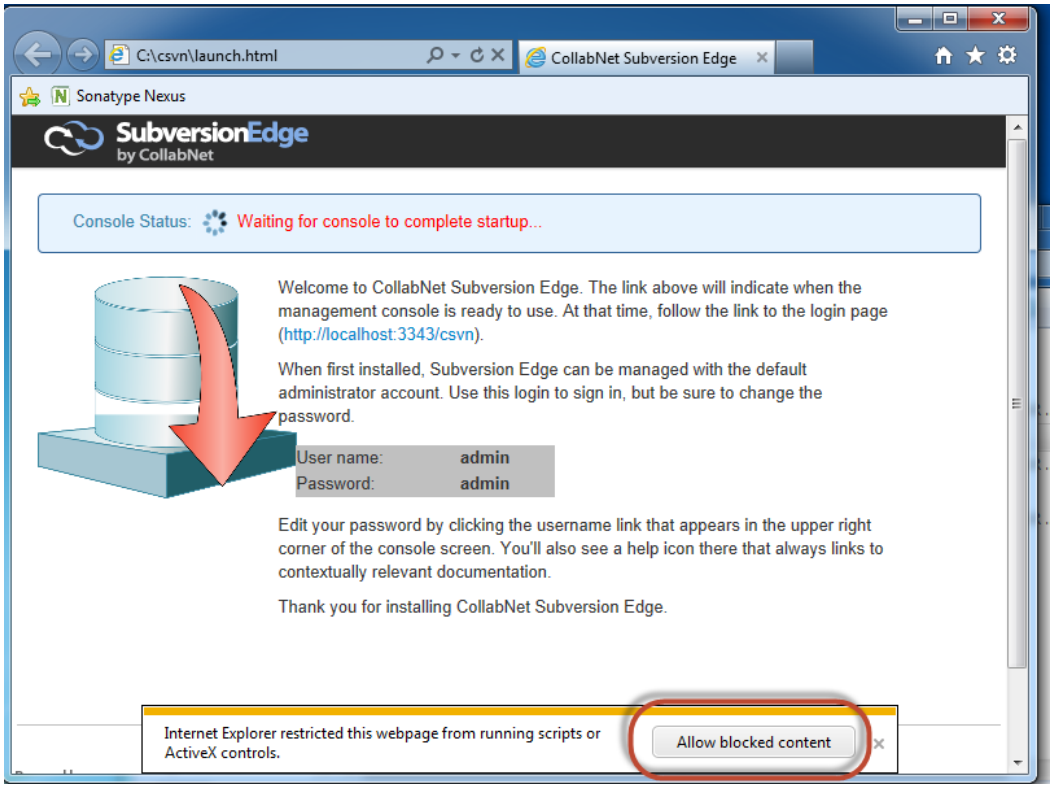
5. Click on **Install**.



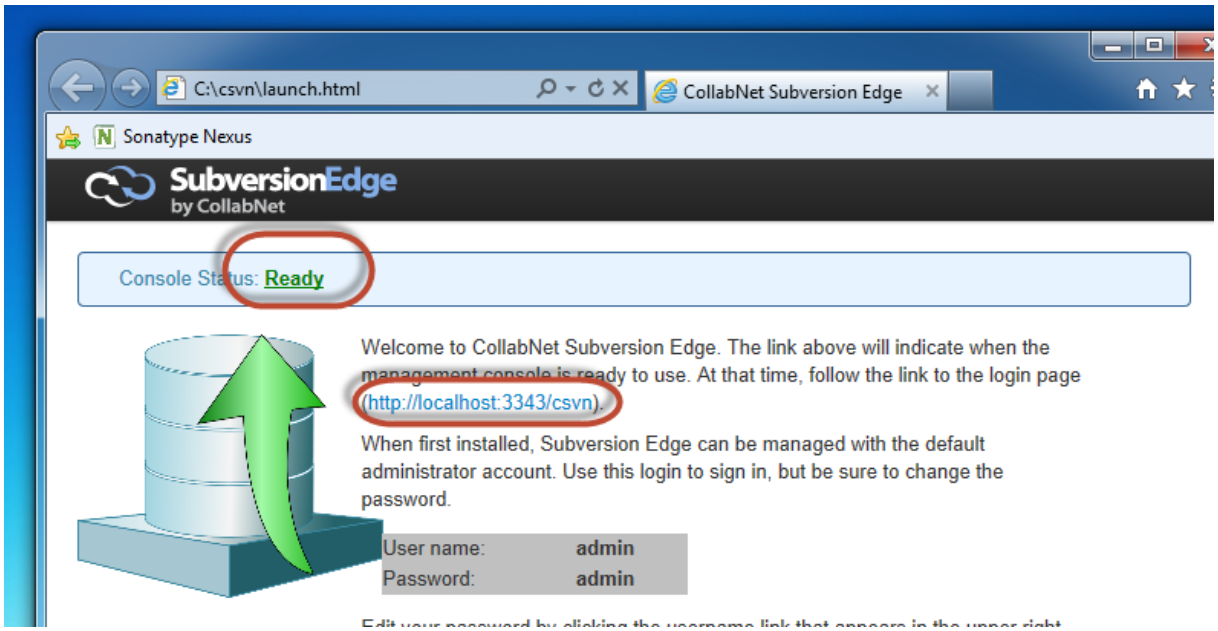
6. Click on **Finish**.



7. The Subversion main page is automatically launched. You may need to **allow the blocked content** for the Internet Explorer.

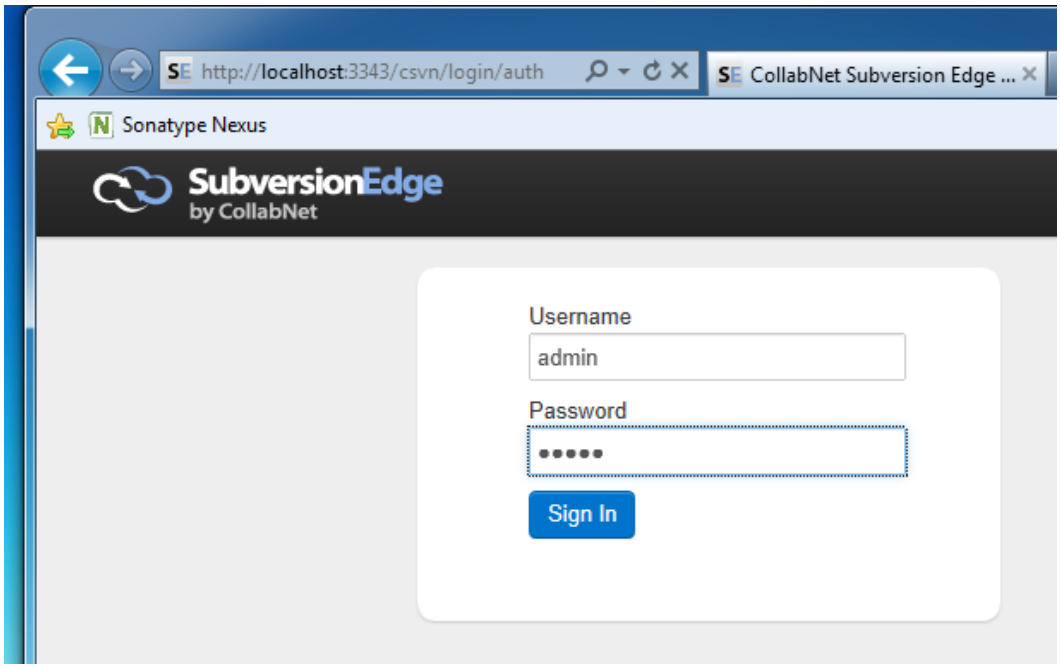


8. After some time, the Console Status is switched to Ready. You can start using Subversion.

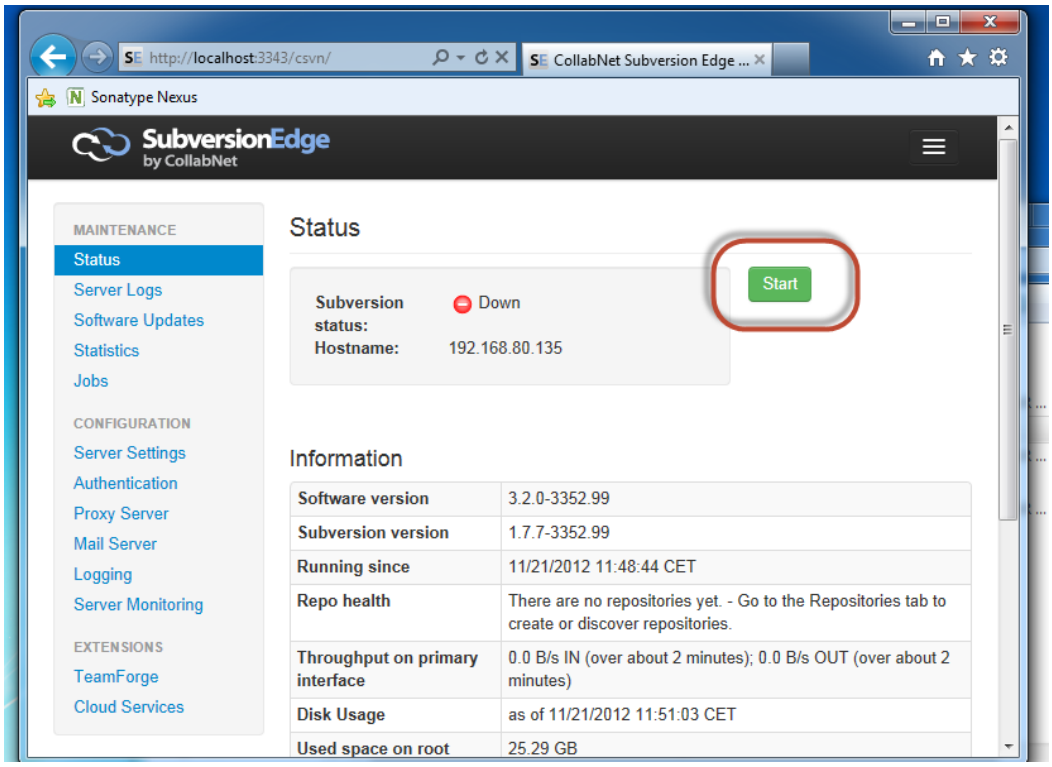


9. Subversion is started and you are automatically prompted to logon to the administration page. The administration page will use the port 3343 of your machine. It can be changed, but we will use the default. In order to access the administration page, you need to provide an administrator account. The default administrator account is:

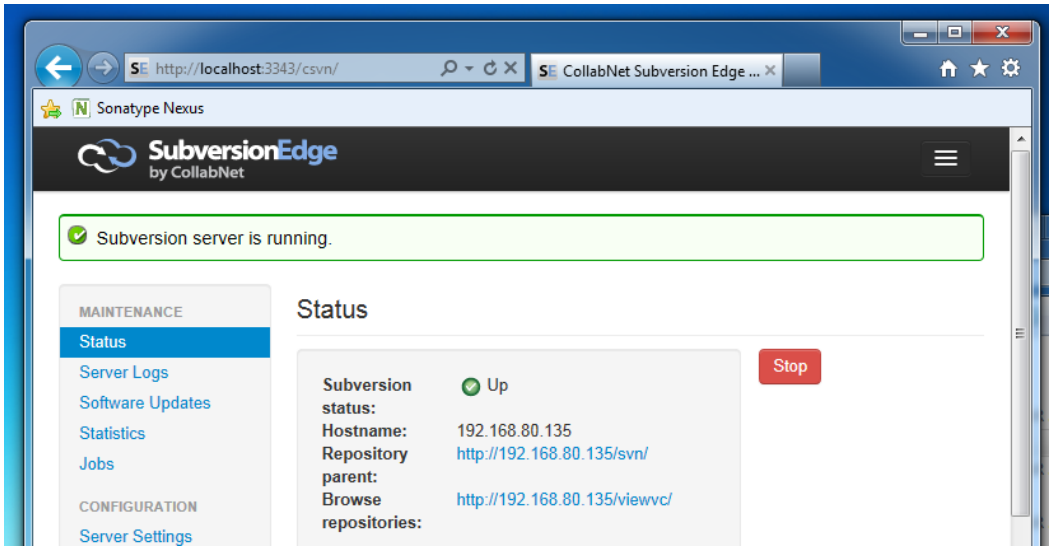
USER	PASSWORD
admin	admin



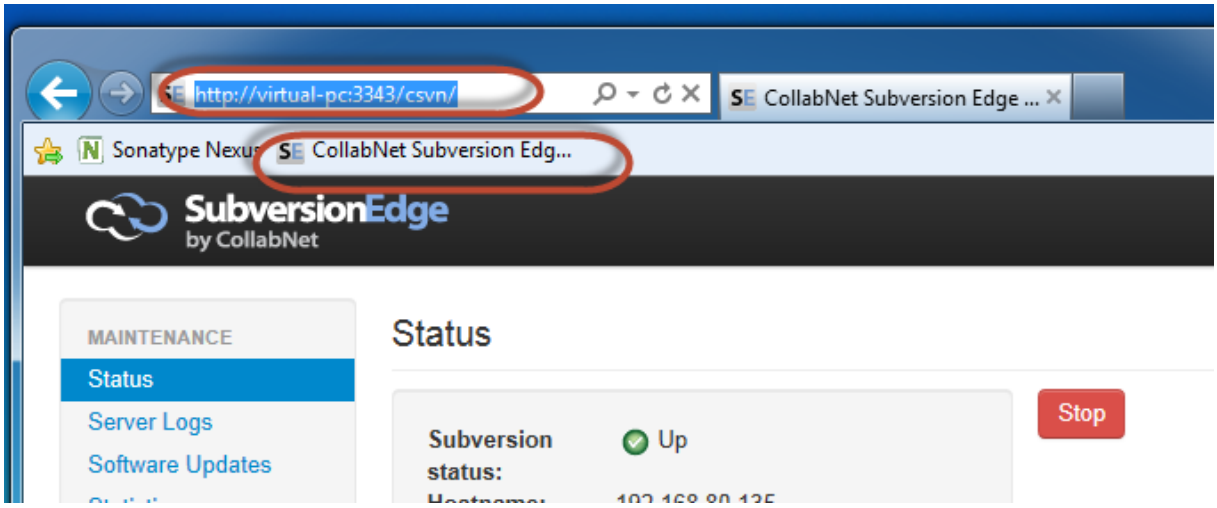
10. Click on the **Start** button to start the server.



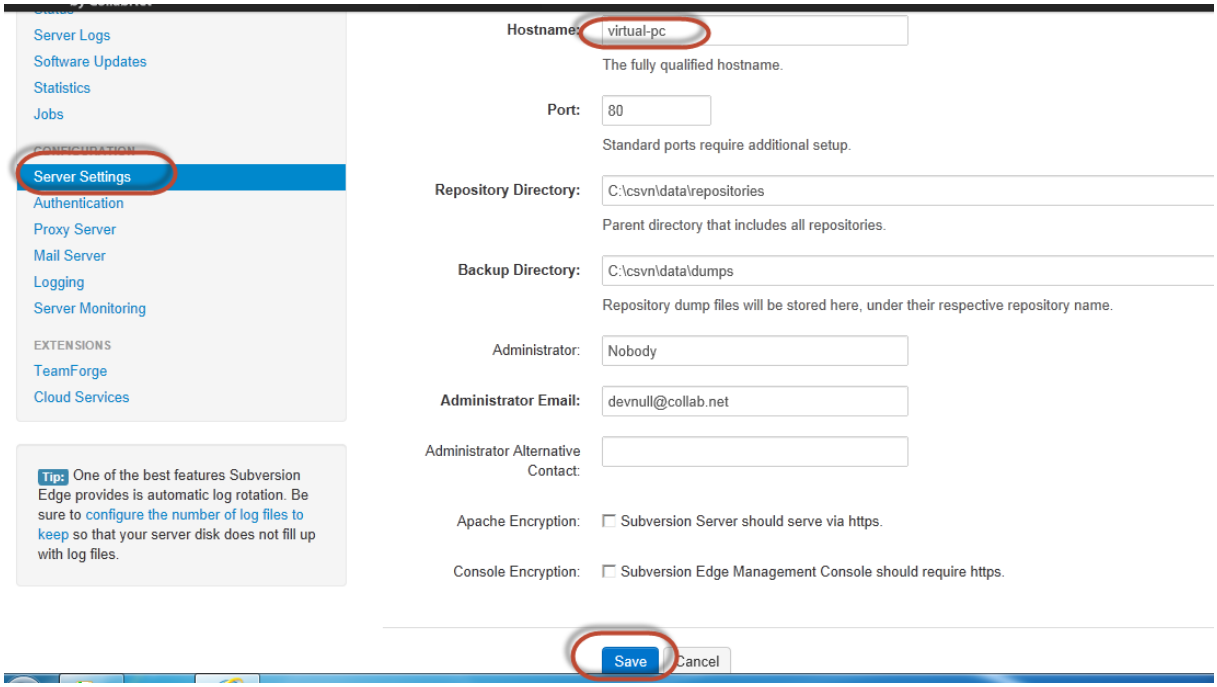
11. Subversion server is now running.



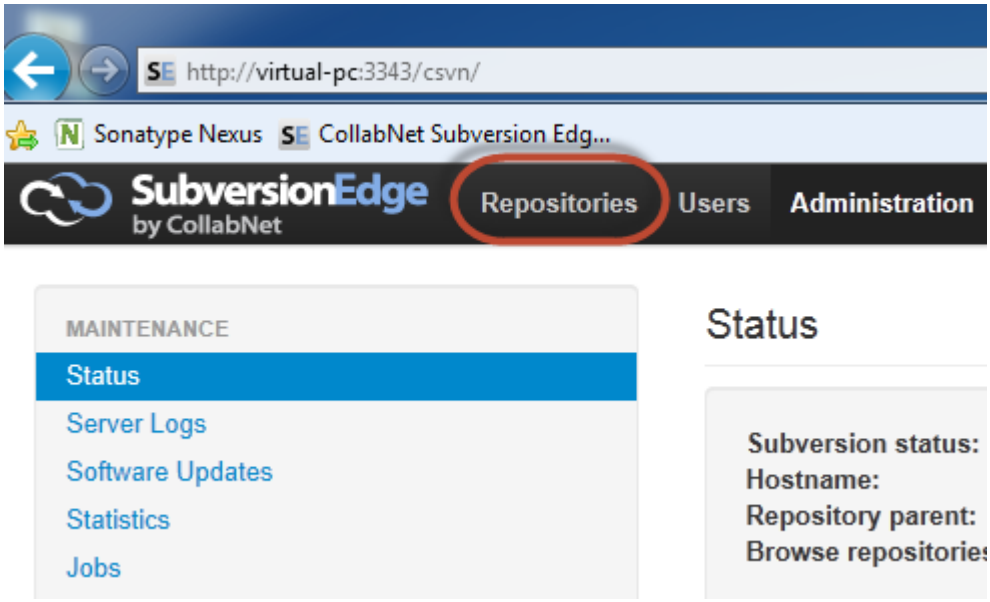
12. In order to avoid any ambiguity, we will use the <server_name> (i.e. in my case, it's "virtual-pc") to logon to Subversion and for our comfort, we can add the link to the Favorites Bar.



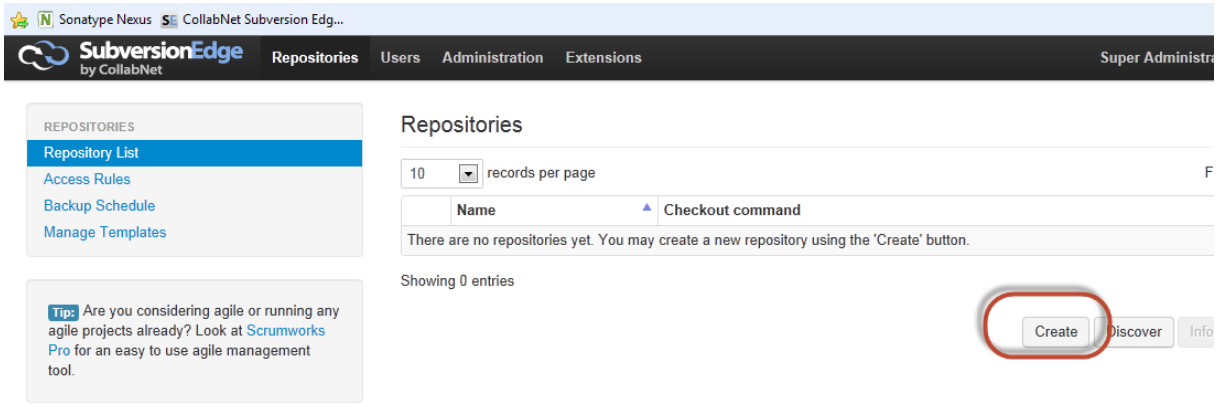
13. Go on **Server Settings** and change the host name to the <server_name>. Then click on **Save**.



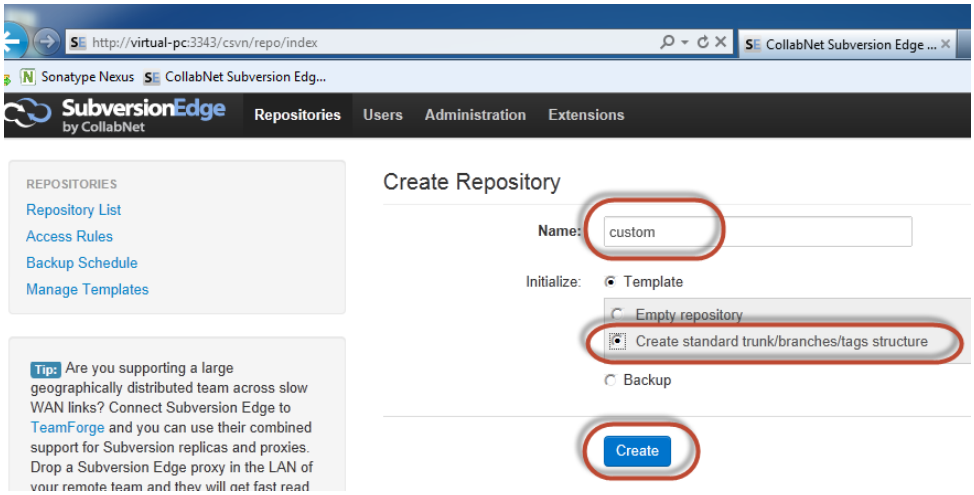
14. Now click on **Repositories**. We need to add a new repository. This repository will be used to host our source code and all the subsequent versions we are going to have.



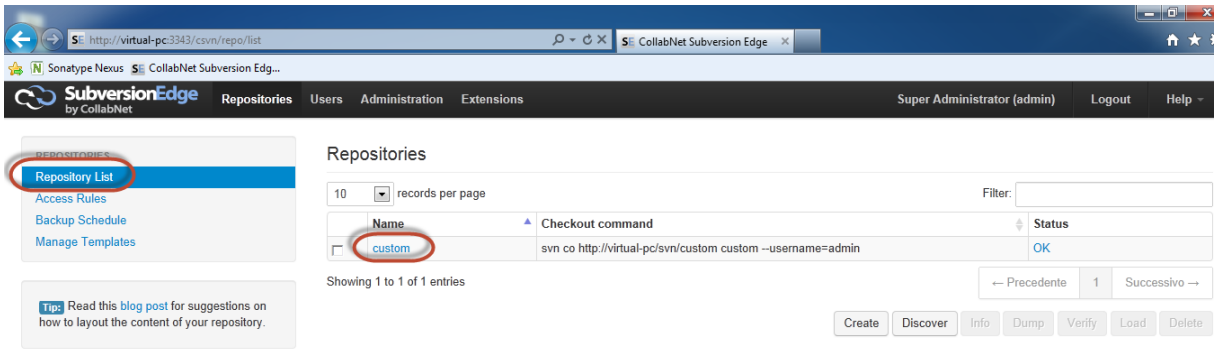
15. Click on **Create**.



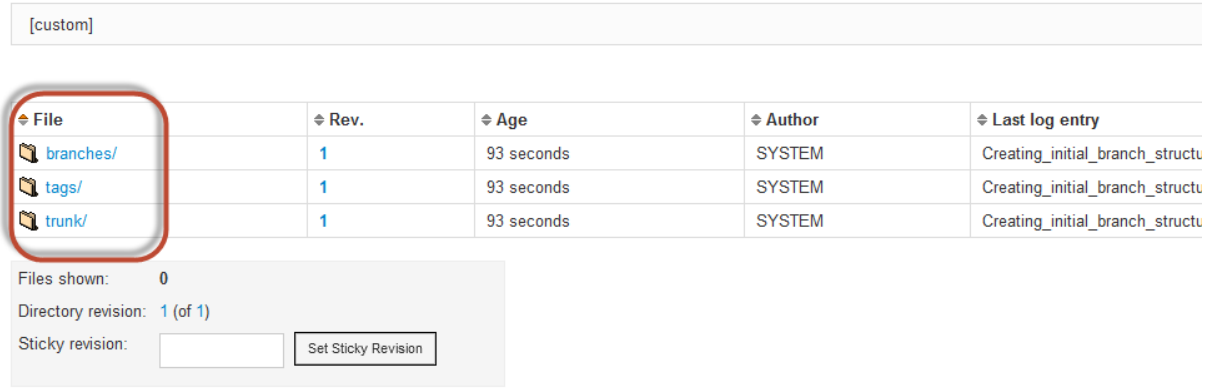
16. Type a repository name (“custom” in my case). Select the option to create a standard trunk/branches/tags structure and then click on **Create**.



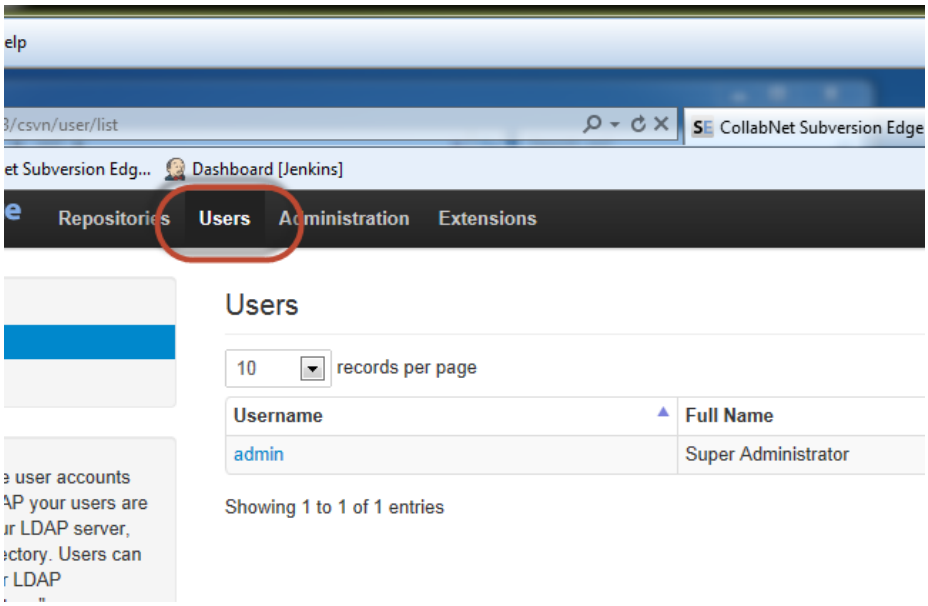
17. If you go back to the repository list, you should be able to see your new repository.



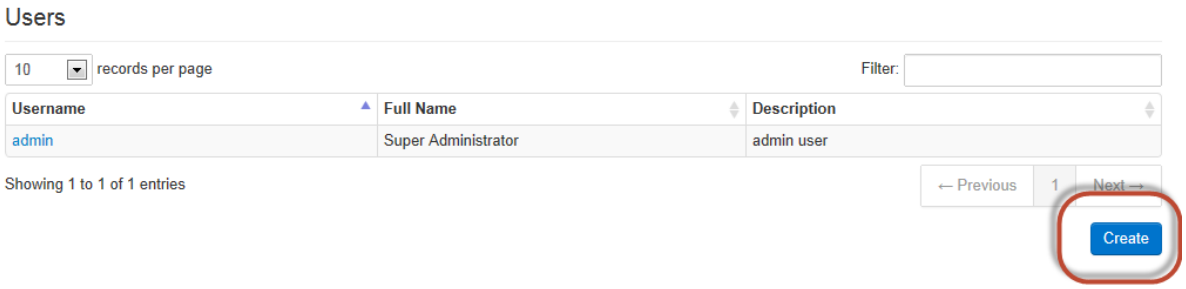
18. The folder “trunk” is where all the main source code is stored, while “branches” is where the users can put their own copies of the “trunk” folder; therefore they do changes to the code without affecting the main code.



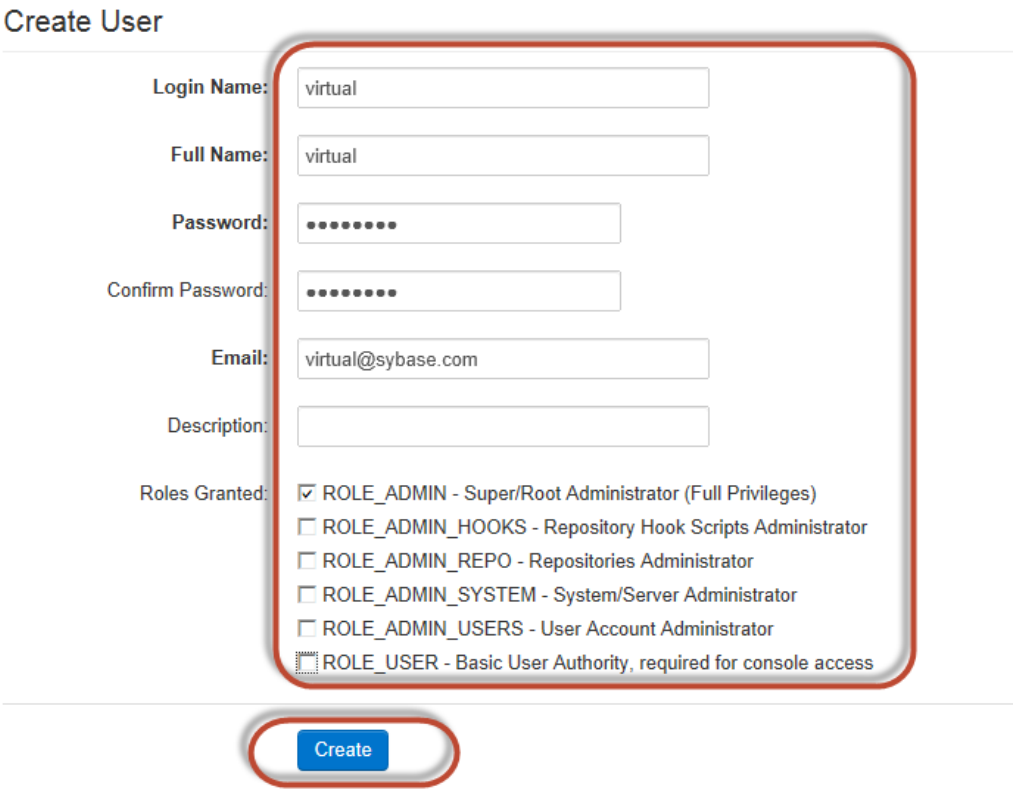
19. Another step at this point is to create a new account inside Subversion with the same user and password as your Windows account. Click on the **Users** tab.



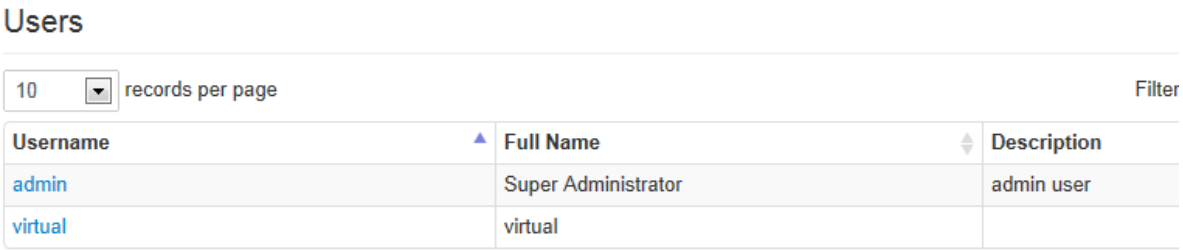
20. Click on **Create**.



21. Fill the page with the required information, assign to this user the ROLE_ADMIN role and click on **Create**.

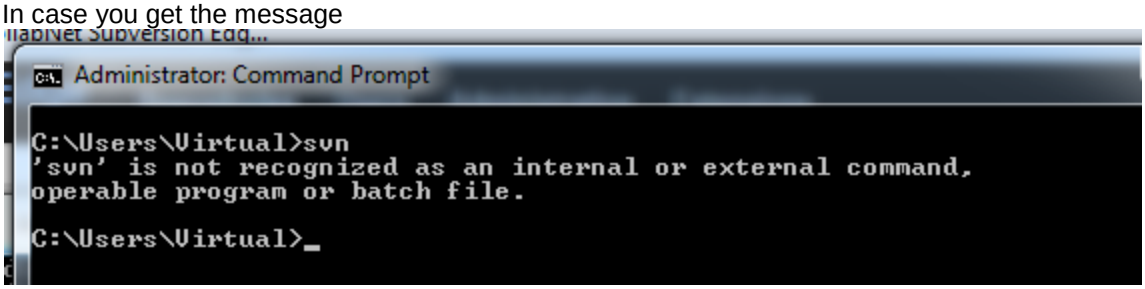


22. You should have two users now in your system.



23. One last optional step is to check that the “svn” command works fine. It may happen that this command is not immediately recognized by the WCMD after the installation. So open the WCMD and simply type the command “svn”.

24. In case you get the message

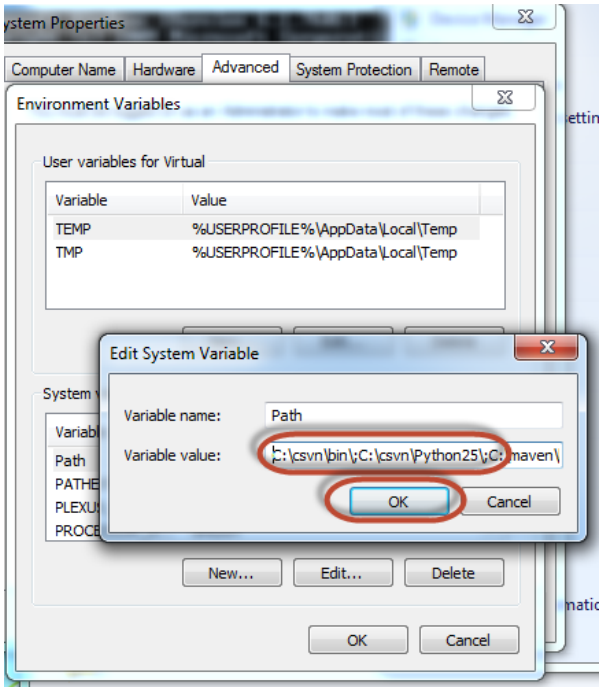


It means that the path variable is not yet fully recognized.

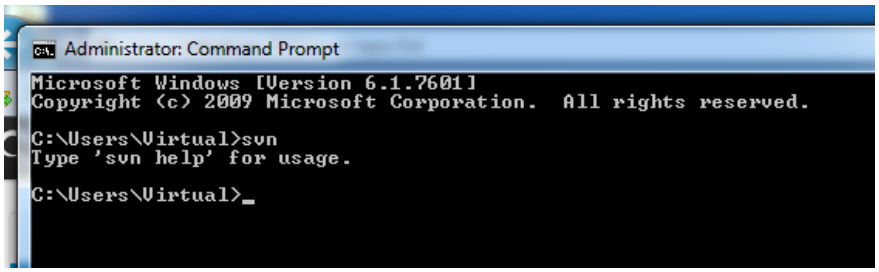
25. Close any open WCMD

26. Open the **Advanced System Settings** again and go to the **Environment Variables**. Locate the PATH variable. Even though you can see the “c:\csvn\bin” folder correctly stored, edit the variable PATH in any case and simply click on **OK**.

    This will force the system to use correctly the new PATH.



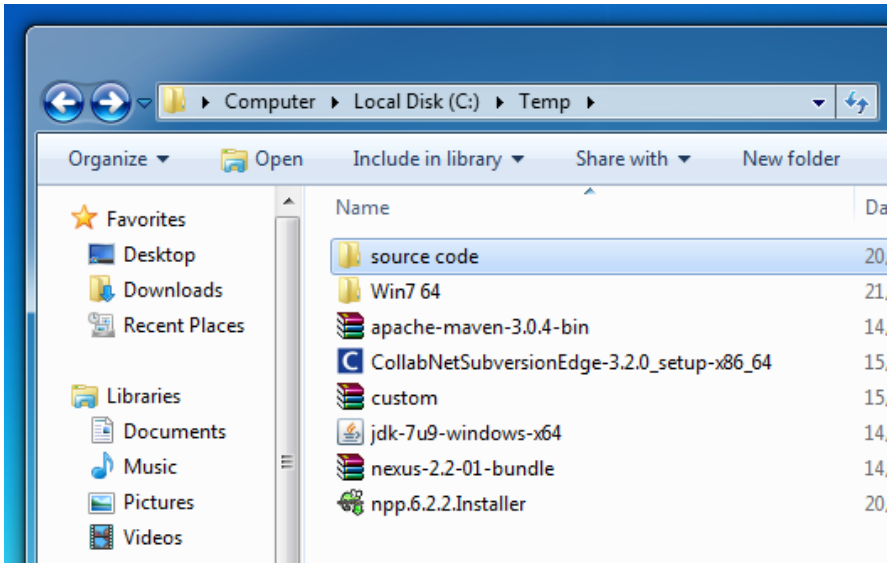
27. Open the WCMD again and type again the command “svn”. You should now get the message “Type ‘svn help’ for usage.



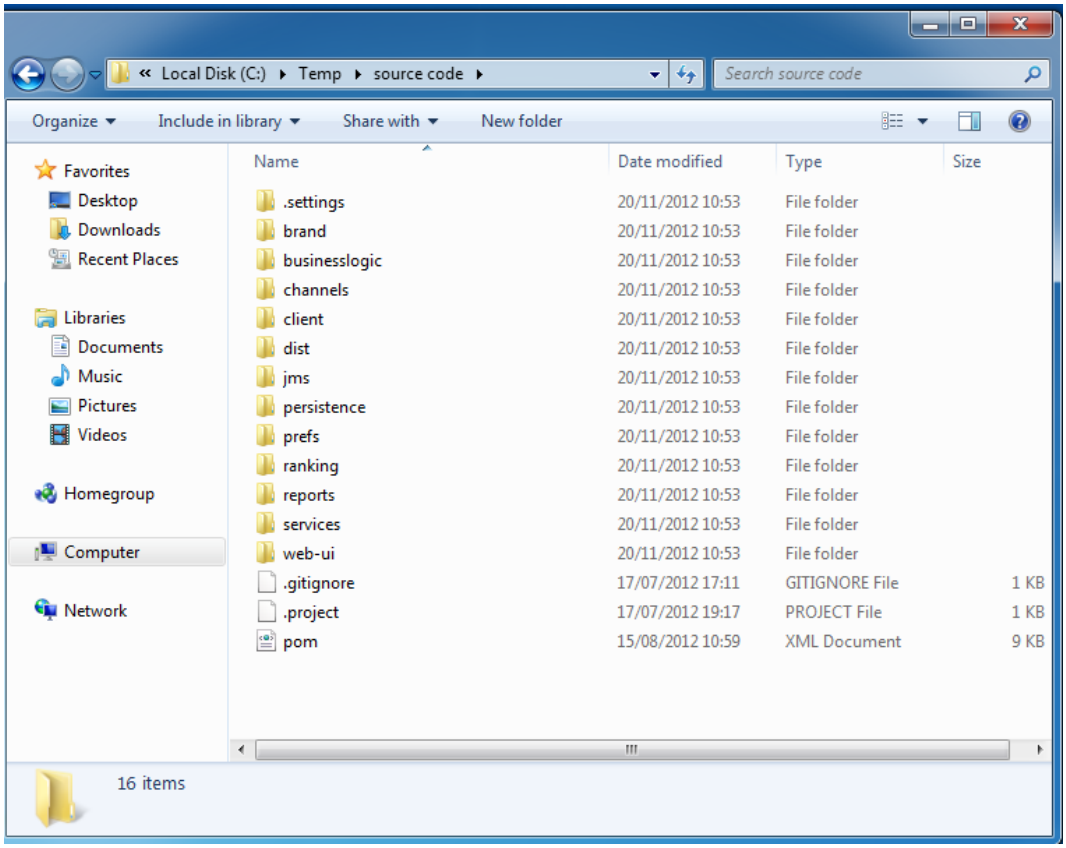
28. This means that your variable PATH is now correctly set up.

4.3 Upload the source code on Subversion

1. Now that Subversion is in place, we need to put inside the source code we have received from SAP Development. The easiest way to do it is to use the Subversion command line operations. The first thing to do is to put a copy of the source code in a temporary folder on the disk. For example, I've copied my source code into the "C:\temp\source code" folder.



2. Inside this folder you can see all the files that are part of our solution. They will be uploaded to the server.



3. As you can see, here there is also the “POM.XML” file that will be the Maven’s starting point for compiling all the source pieces.

   We will use the command line for uploading this source code to the server. Open the WCMD and go to the folder containing the files to upload, using the following command

   CD “c:\temp\source code”

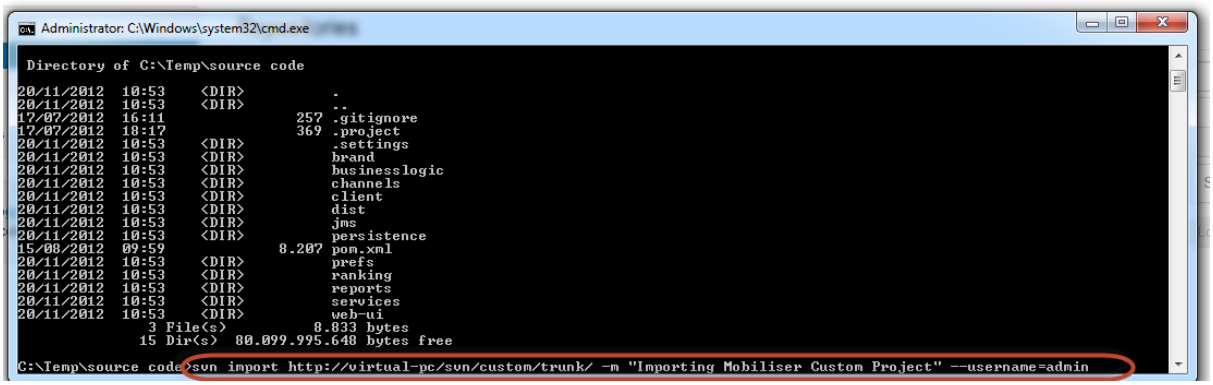
4. Once in the source code folder, run the following command:

   svn import http://<server_name>/svn/<repository_name>/trunk/ -m "Importing Mobiliser Custom Project" --username=admin

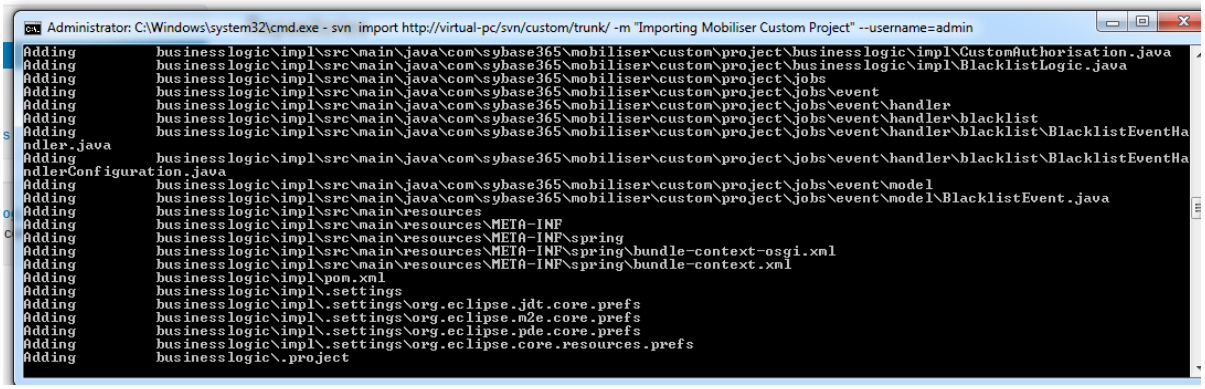
   (in my case it is *svn import http://virtual-pc/svn/custom/trunk/ -m "Importing Mobiliser Custom Project" --username=admin*)

   You will be requested with the admin’s password, which is “admin”.

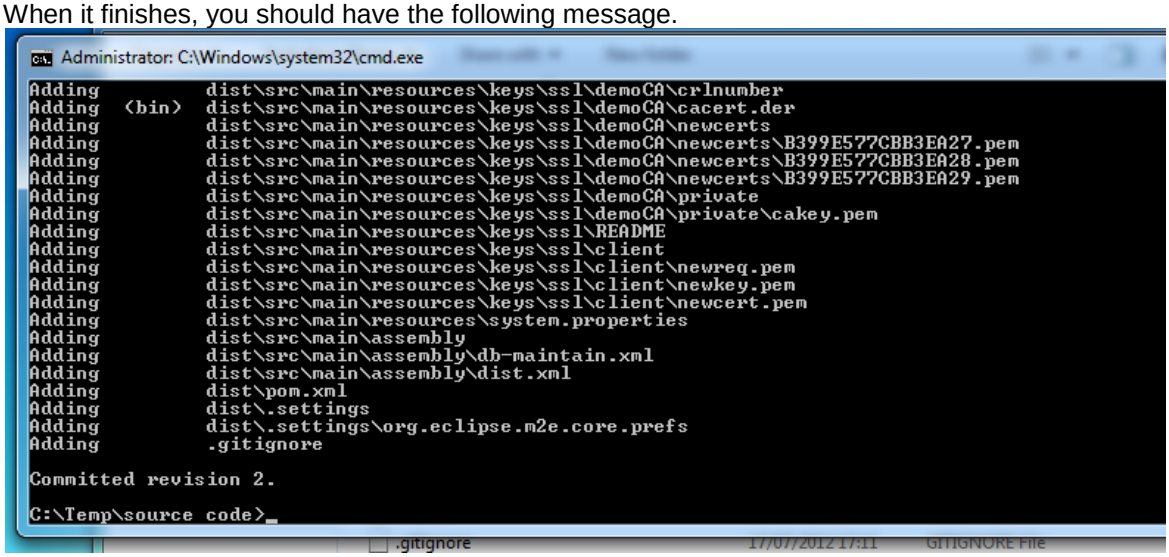
   You can also use the other account you have created for Subversion.



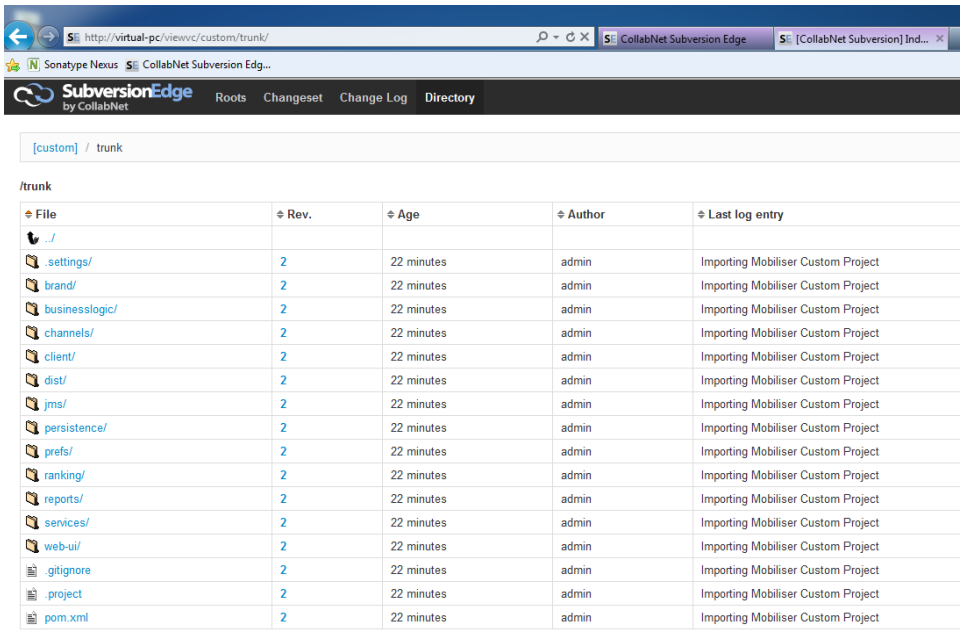
5. The import phase starts, as depicted below.



6. When it finishes, you should have the following message.



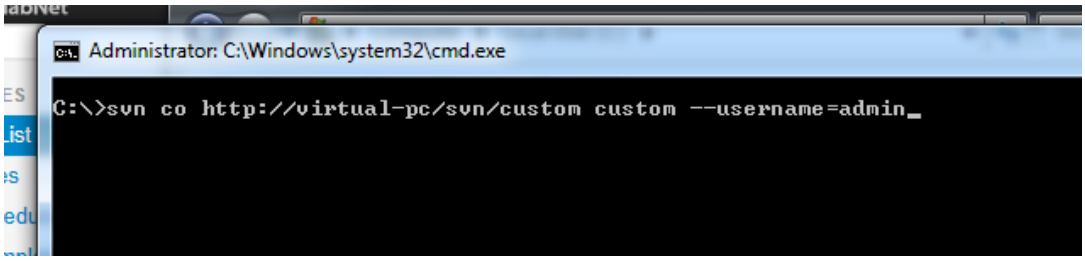
7. If you go back on the Subversion page you can see now the content of your trunk folder.



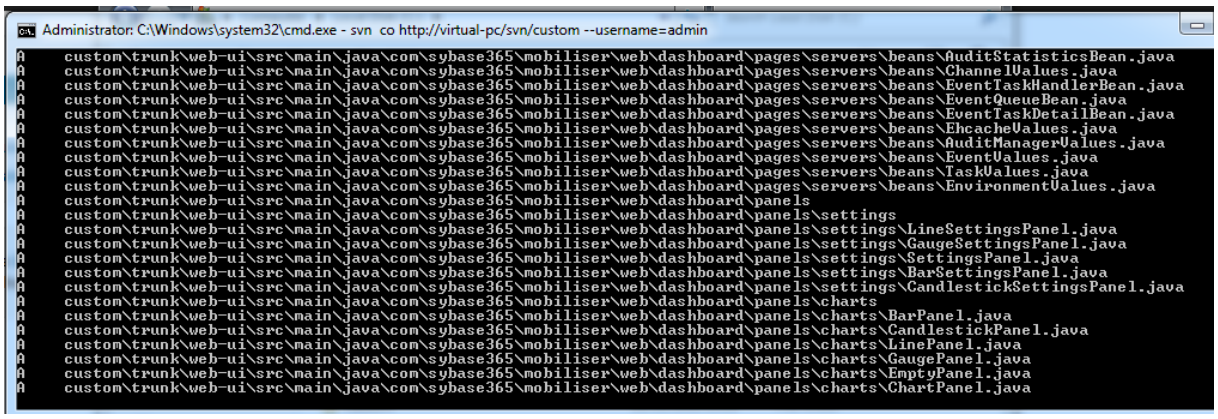
8. You could now remove the source folder located in the temp folder and create your own source folder by checking out the Maven project directly from the source repository. In order to do this, go on the WCMD. Reach the C:\ folder and type the following command:

   svn co http://<server_name>/svn/<repository_name>/trunk <repository_name> --username=admin

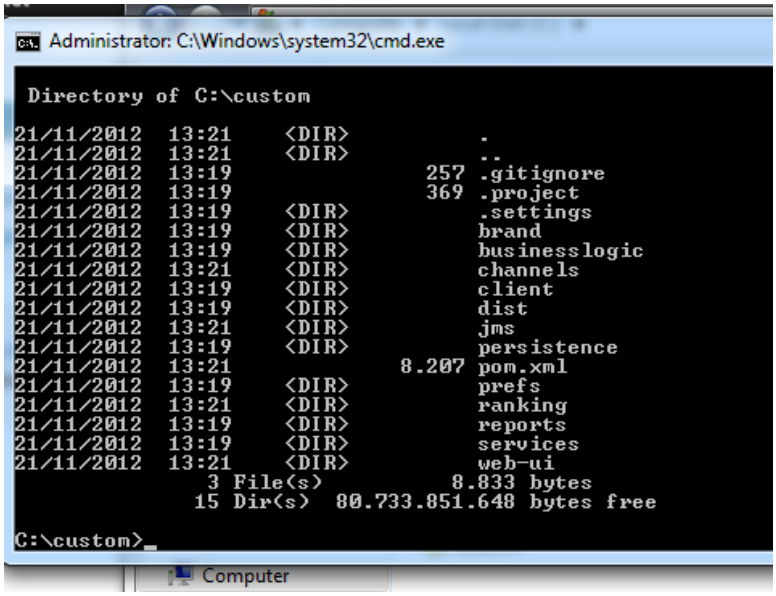
   (in my case it is *svn co http://virtual-pc/svn/custom/trunk custom --username=admin*)



9. The checkout process starts.



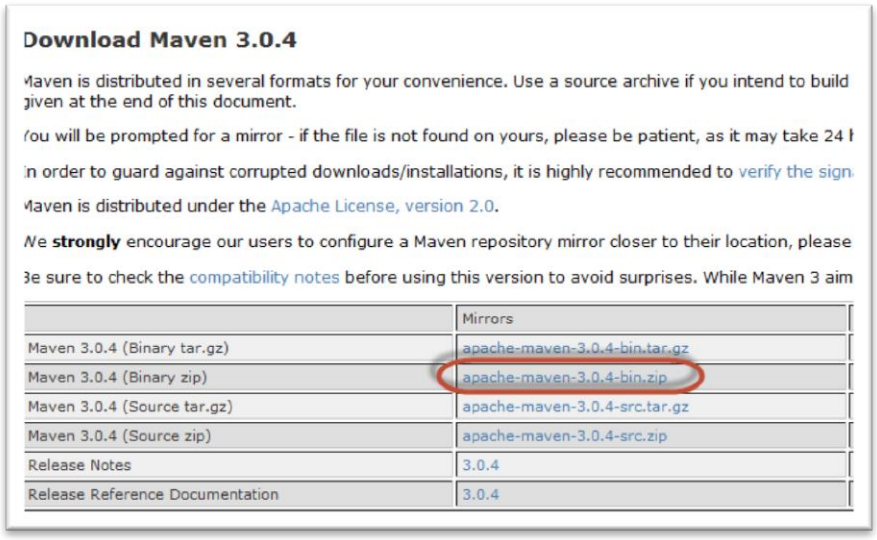
10. At the end you will have a **C:\<repository_name>** folder (in my case *C:\custom*) with the required source code. This will be the folder that we will use for the next chapters.



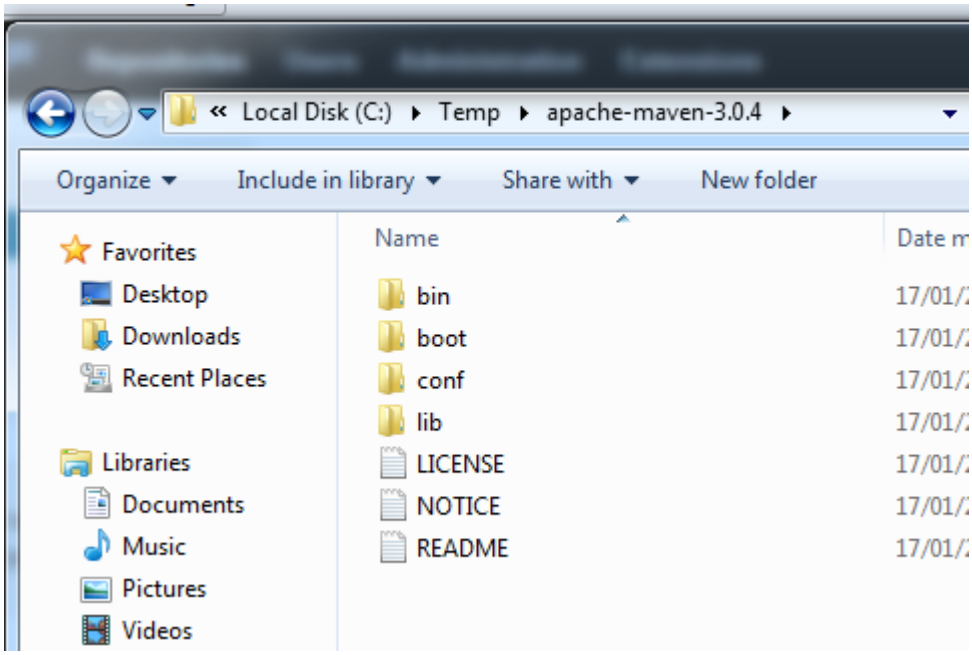
4.4 Install Apache Maven

1. Apache Maven can be downloaded from the following web site:
   http://maven.apache.org/download.html

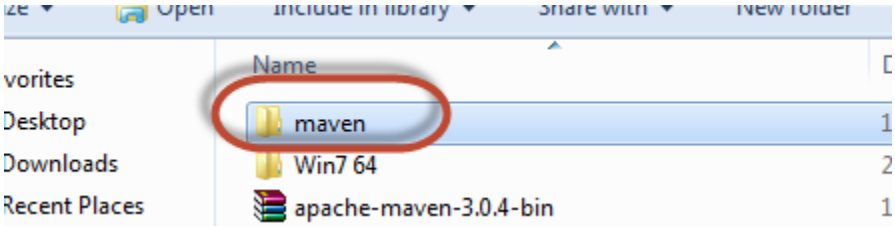
   For the purpose of this guide, we have used version 3.0.4, which is the current latest. We will download and install this version.



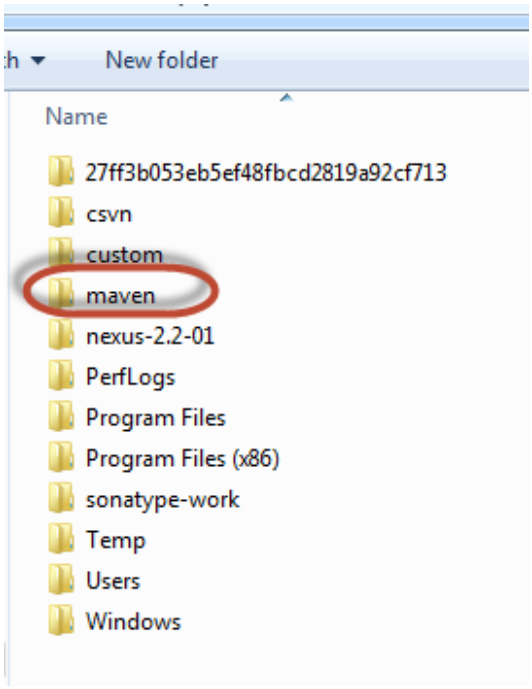
2. Once you have downloaded the package, extract it to a folder on your disk. In my case, I have put it in the folder **c:\temp\apache-maven-3.0.4**



3. Go up to the parent directory and rename this folder simply as **maven**.



4. Just for ease of use, move this folder on the root of your C drive.

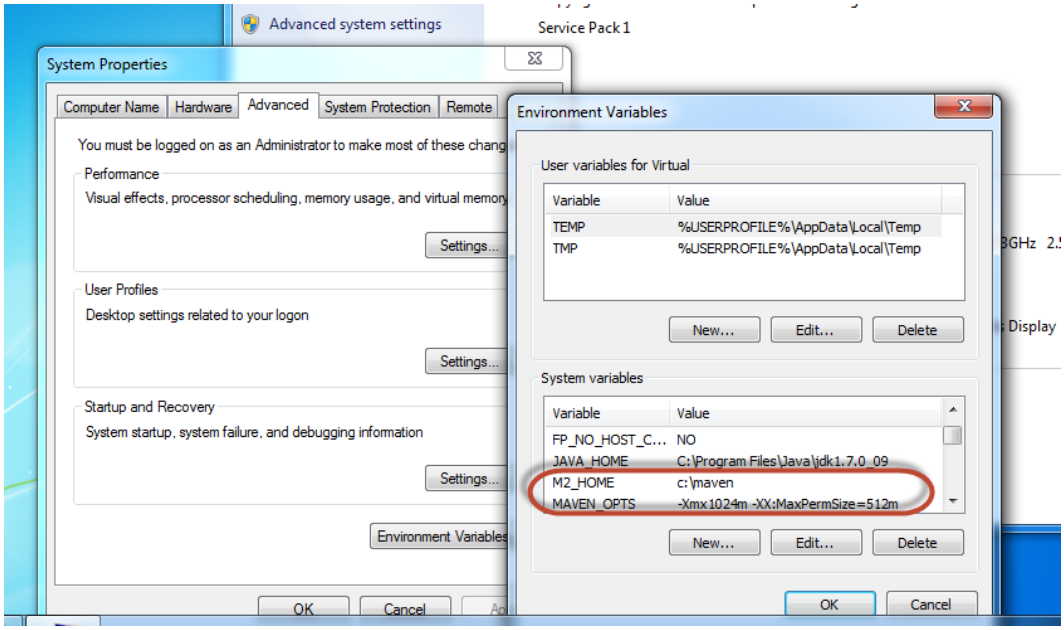


5. Maven is now in place, but we will have to modify and define some variables in order to make it fully working.
   Open the **Advanced System Settings** and click on the **Environment Variables** button.

6. We need to create the following two variables:

   M2_HOME=c:\maven
   MAVEN_OPTS=-Xmx1024m -XX:MaxPermSize=512m

   The first one specifies the base directory for Maven; the second one defines some memory adjustments for making it running fine.

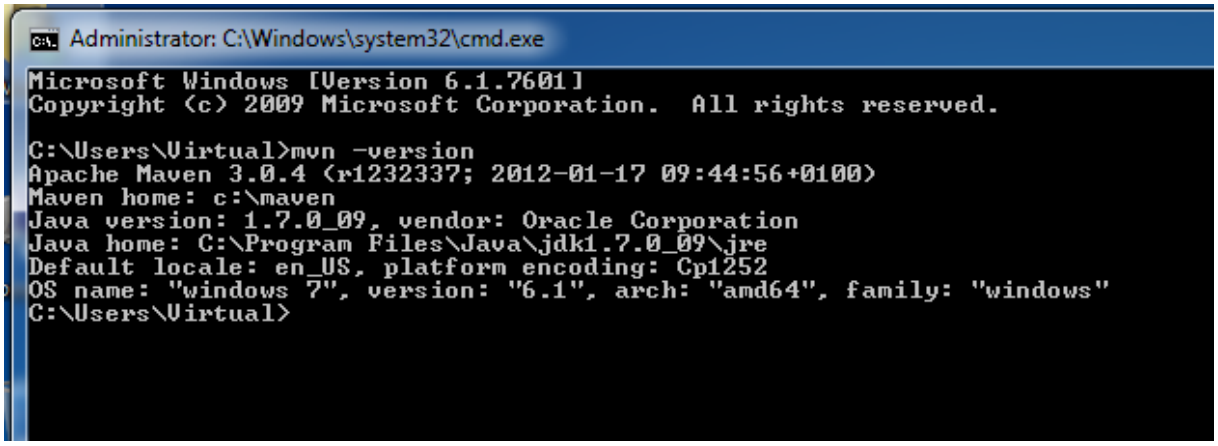


7. Now we need to adjust the variable **PATH**. Locate and edit it, by including the path **c:\maven\bin**.

8. Once you have done your changes, you can test if the command is visible from the WCMD. Open a WCMD and simply type:

   mvn –version

   You should get the following result

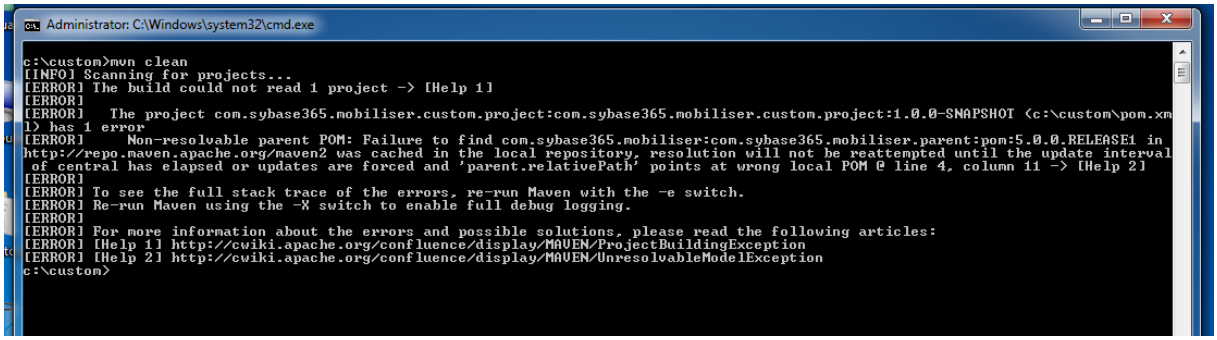


9. One last step is to define some settings for Maven. Even though this may appear as a complex operation, it will be explained here as simple as possible. The standard configuration file is stored in the **c:\maven\conf** folder and it is named **settings.xml**. This file, normally, contains all the generic settings for Maven, but after you have just installed Maven, it doesn't contain any setting: all the lines inside are commented out.

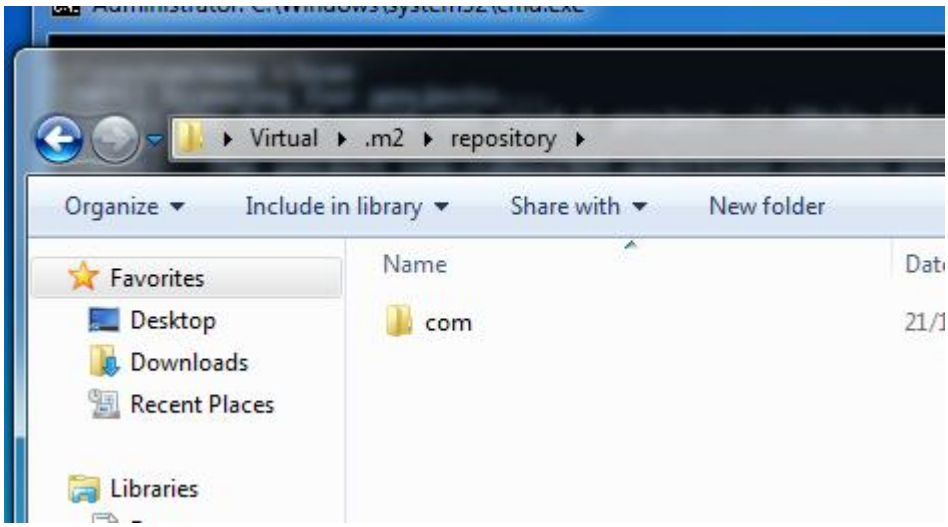
   For exemplification purposes, we can now see what happens if we want to start compiling our source code we have checked out in the **C:\custom** folder using the **settings.xml** file that comes out with the installation package. Open a WCMD and go into the **c:\custom** folder.

10. Run the command **mvn clean**. The scope of this command is to prepare the building of the project by erasing the output folder.

    You should most probably get an error like this:



11. It happens because Maven is not able to download all the artifacts needed for the construction of this project, as defined by the POM.XML file located in the current folder. This means that we need to specify somewhere, for Maven, the location of the artifacts' repository. Since we have already Nexus in place, we have to tell Maven to use our Nexus system. The **mvn** command will read **the settings.xml** and will download the required artifacts by caching them locally in a folder on the disk. The standard folder for this is **c:\Users\<windows_user_name>\.m2**. In this folder you should find another folder called "repository" which will host all the required artifacts downloaded from Nexus. At the moment it's empty since Maven doesn't know yet where to get the artifacts from.



12. We need to specify this information for Maven. We could edit the original **settings.xml** file located in the **c:\maven\conf** folder, but we can also create a new file into another location: the **c:\Users\<windows_user_name>\.m2**, where the repository is located. The settings file in the **c:\maven\conf** folder is common to all the Maven users, the other, instead, located under your Windows account, is just used by your user.

    For this reason we can delete or simply ignore the settings file under the **c:\maven\conf** folder and place another file under the **c:\Users\<windows_user_name>\.m2** folder.

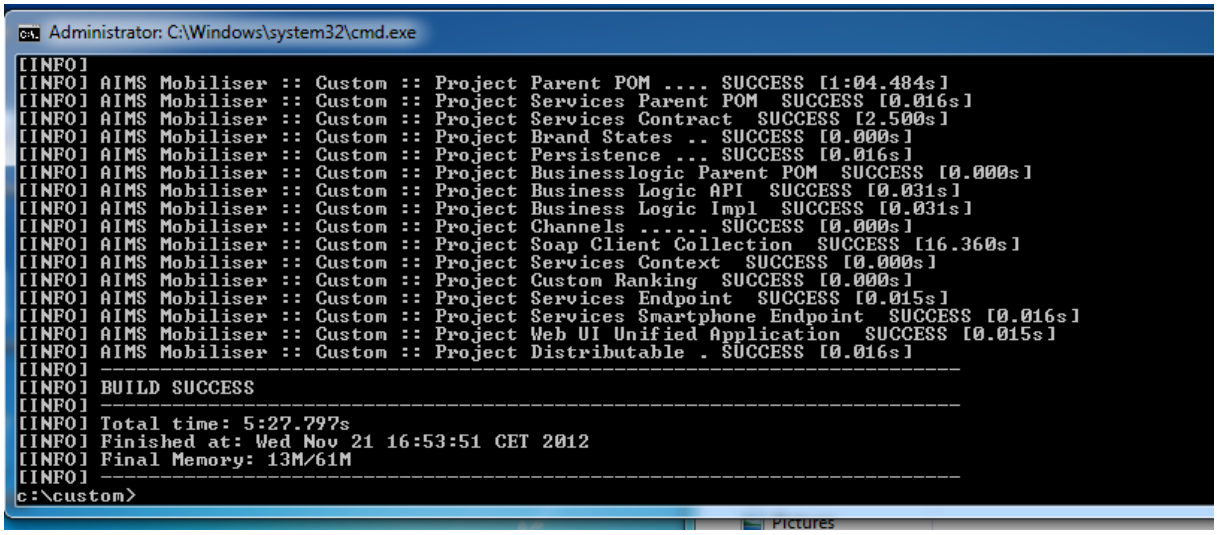
    This file should look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
       http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <mirrors>
    <mirror>
      <!--This sends everything else to /public -->
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>http://virtual-pc:8081/nexus/content/groups/public</url>
    </mirror>
  </mirrors>
  <profiles>
    <profile>
      <id>nexus</id>
      <!--Enable snapshots for the built in central repo to direct -->
      <!--all requests to nexus via the mirror -->
      <repositories>
        <repository>
          <id>central</id>
          <url>http://central</url>
          <releases><enabled>true</enabled></releases>
          <snapshots><enabled>true</enabled></snapshots>
        </repository>
      </repositories>
     <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>http://central</url>
          <releases><enabled>true</enabled></releases>
          <snapshots><enabled>true</enabled></snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
  <activeProfiles>
    <!--make the profile active all the time -->
    <activeProfile>nexus</activeProfile>
  </activeProfiles>
</settings>
```

Note

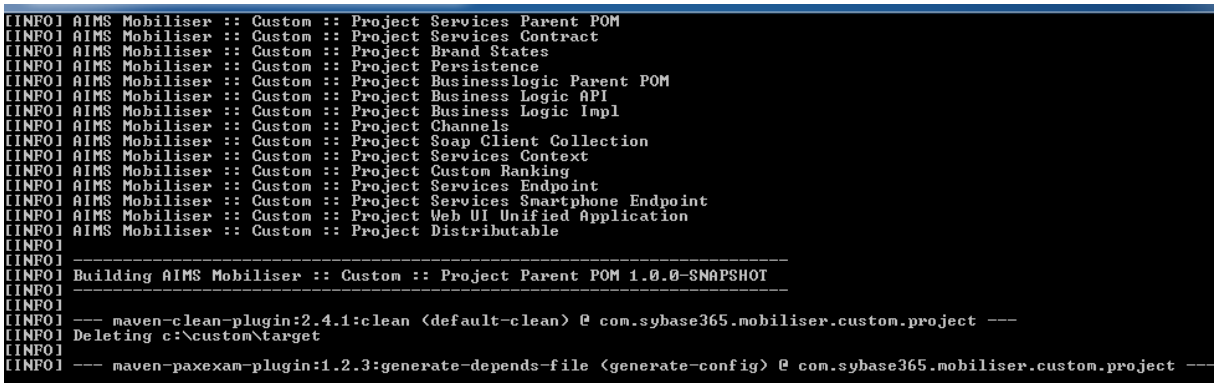
Pay attention to the string marked in yellow. That string is the link to the Nexus repository.

13. After putting this code inside the settings.xml file, retry the above **mvn clean** command. You should have the following result:



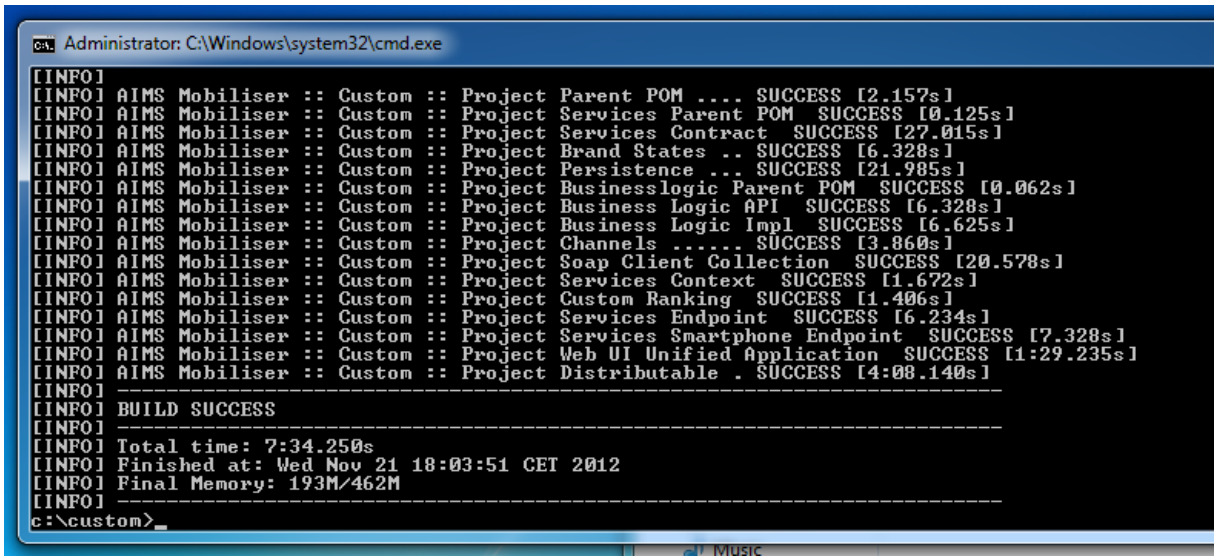
14. If that was successful, you can use also another command which will completely rebuild your solution and will generate the executable programs. The command is the following:

mvn clean install



15. Please note that this is the first time you are running this command and since it has to download all the artifacts from the repository and to cache them locally, it may require a lot of time. Depending on the speed of your network it could require even more than an hour. The total size of the final local repository folder will be approx. 650MB for this particular build.

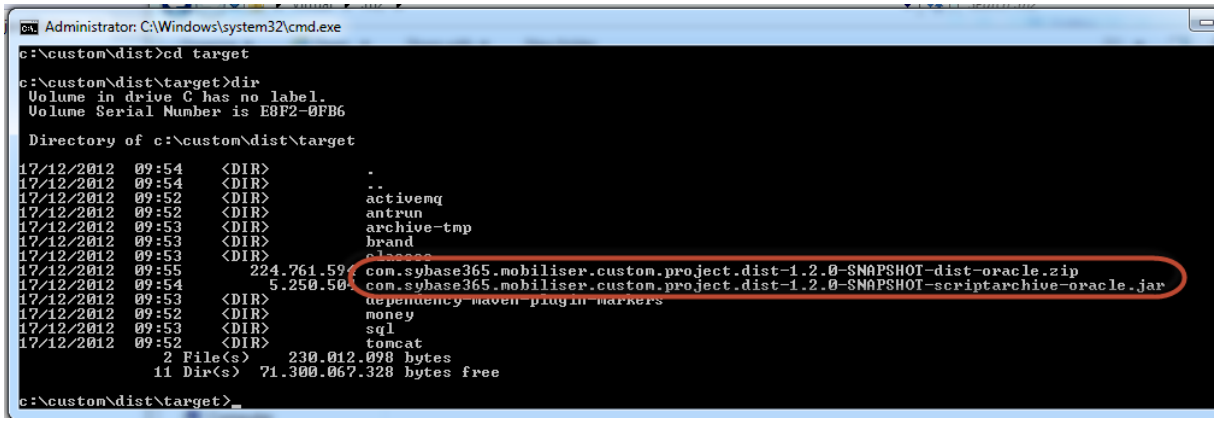
    When the build finishes, you should receive the BUILD SUCCESS message:



16. If you go in the folder **c:\custom\dist\target** you will find two main big files:

    com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-dist-oracle.zip
    com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-scriptarchive-oracle.jar

    These two files are the ones that we will use in the next chapters for preparing the database and running our Mobiliser instance.

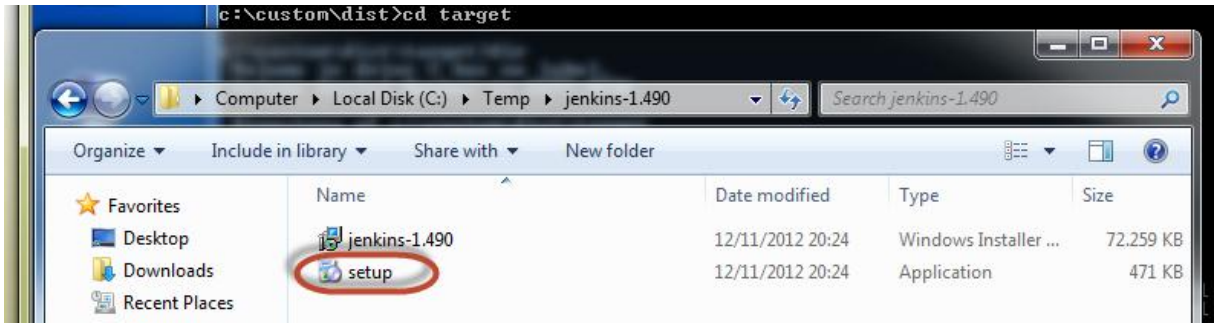


4.5 Install Jenkins

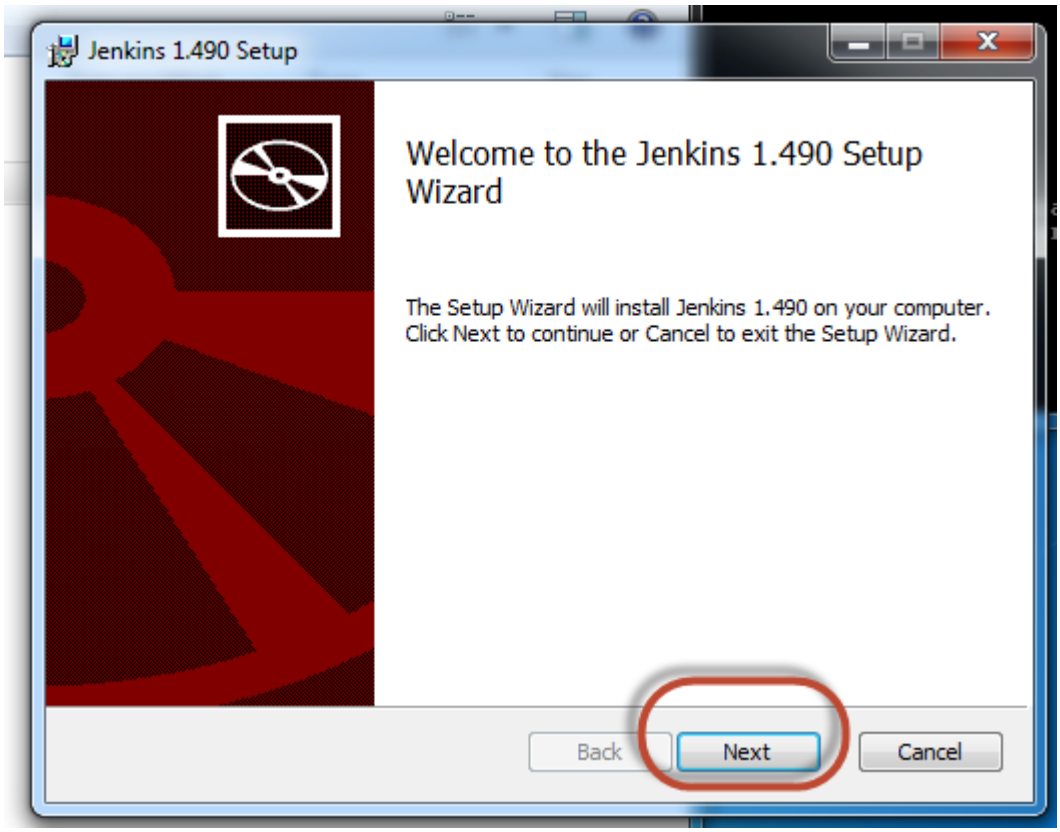
1. Jenkins can be downloaded from the following web site:

   http://jenkins-ci.org/

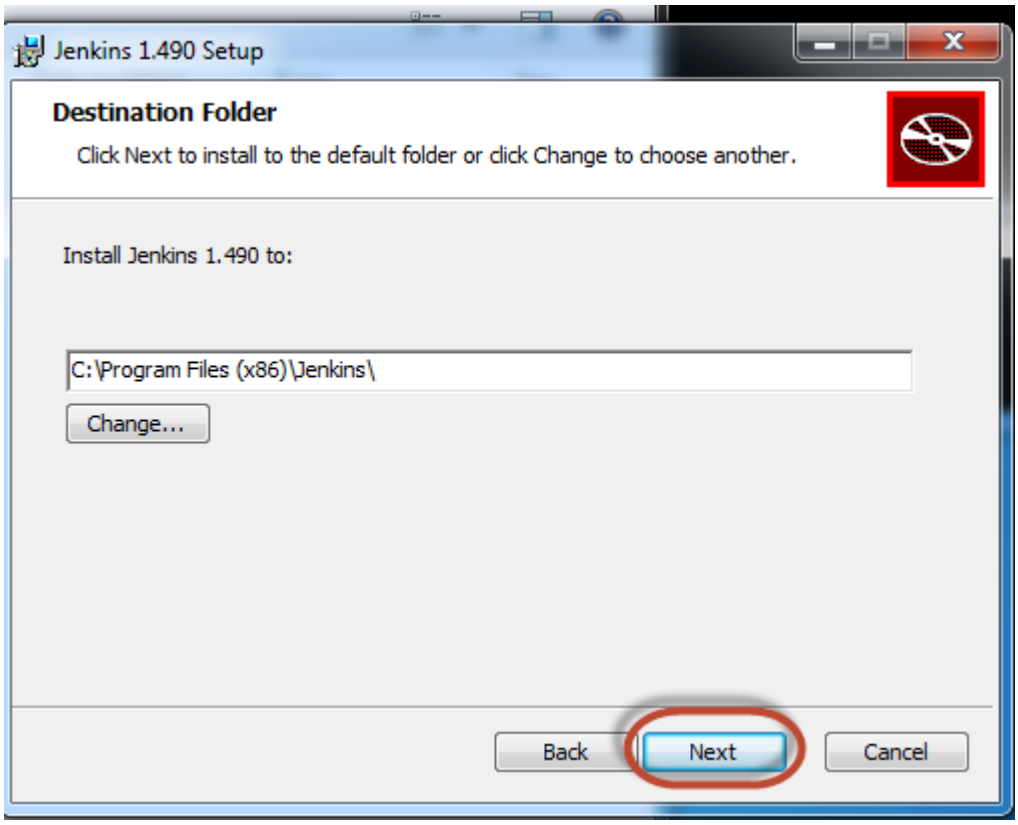
   Unpack the downloaded package in a folder on the disk and you will find two files. Double click on the **setup** file in order to start the installation:



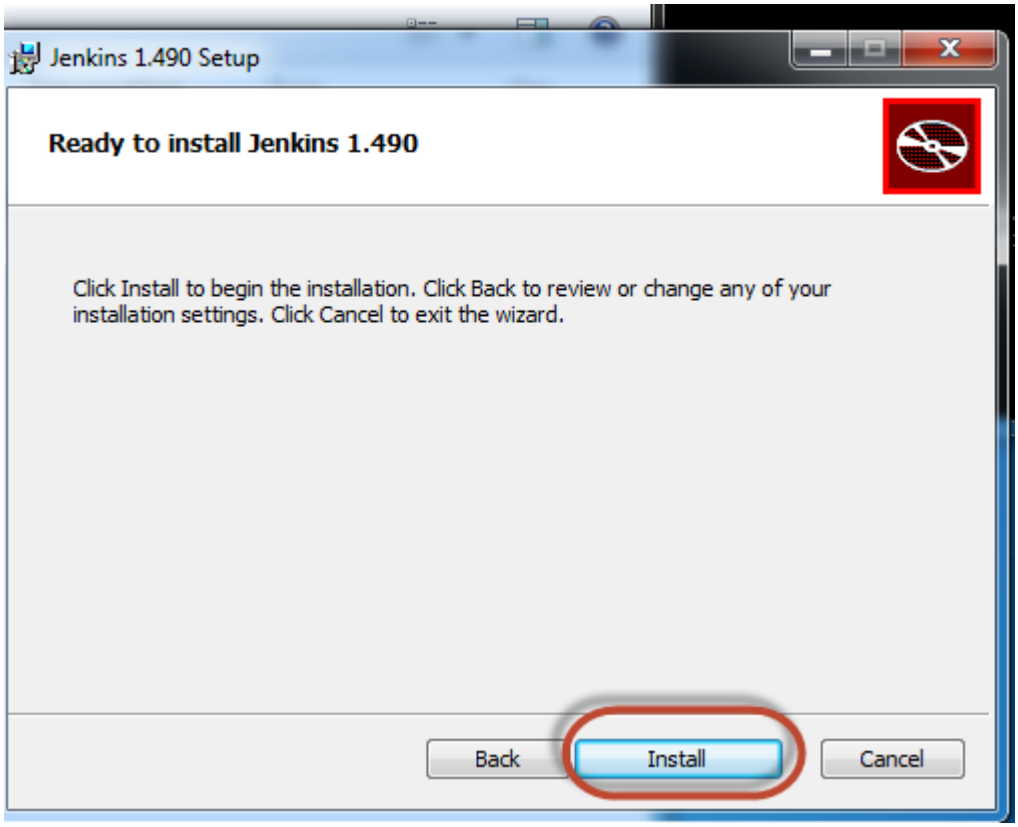
2. Click on **Next**.



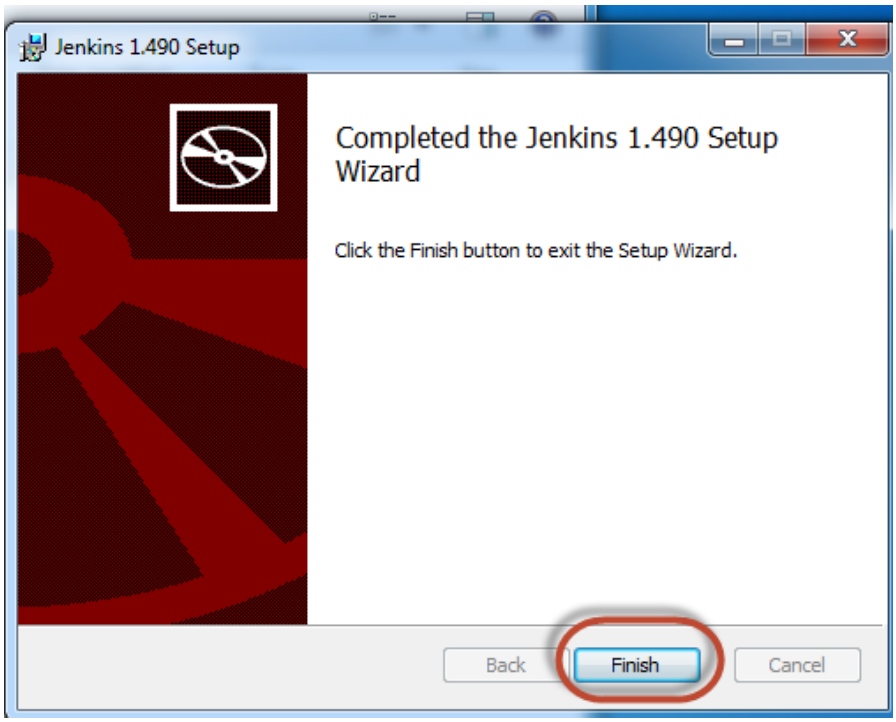
3. Click on **Next**.



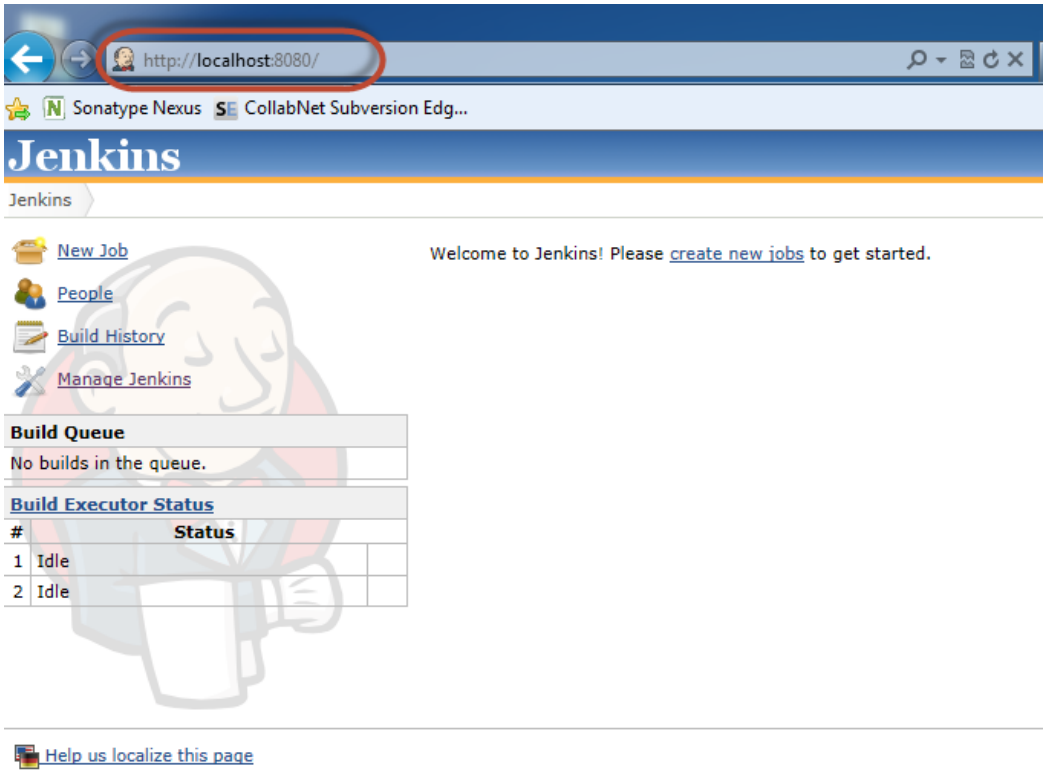
4. Click on **Next**.



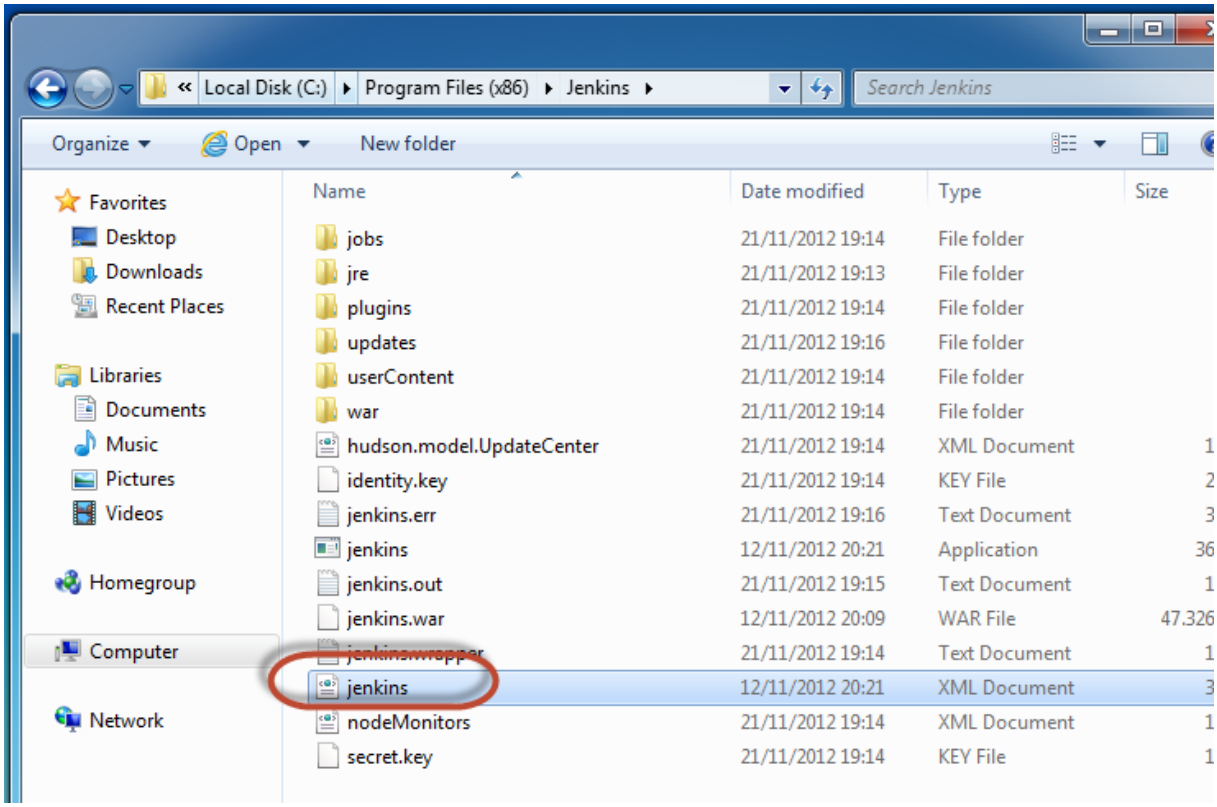
5. Click on **Finish**.



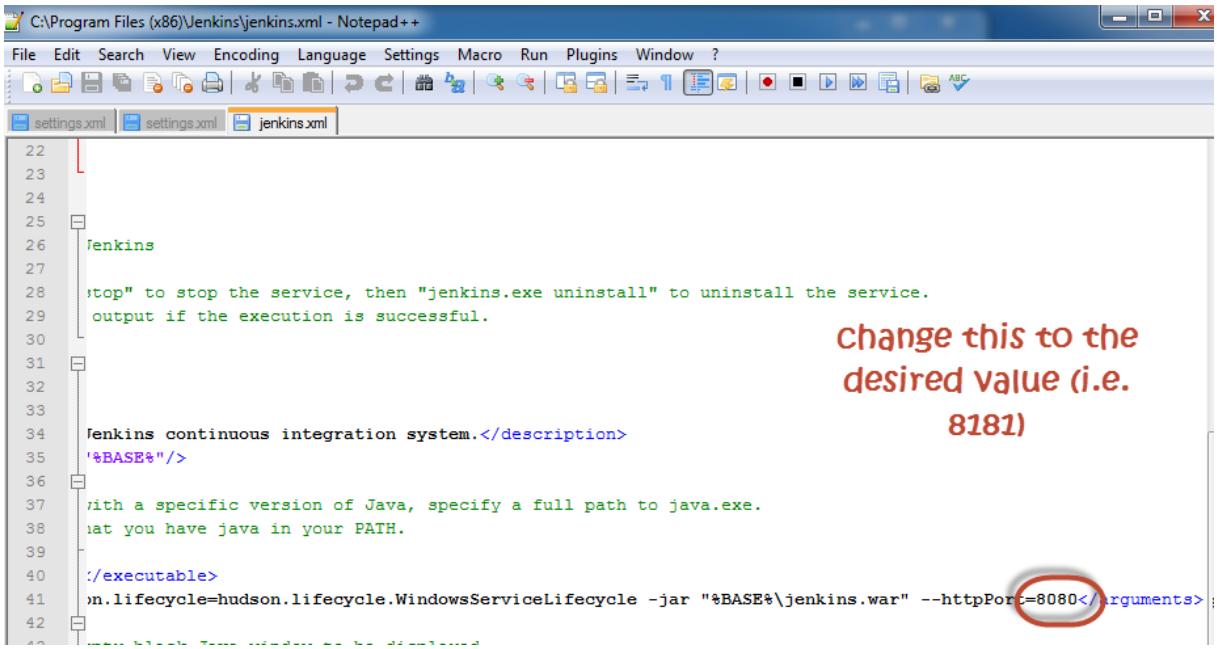
6. Once Jenkins is installed, the Jenkins Dashboard is automatically opened. As you can see here, by default, the Jenkins service is installed on the port 8080. Since this port will be used by Mobiliser as well, it's easier and nicer to change the port for Jenkins.



7. In order to do this, go in the Jenkins installation directory and open, for editing, the file **jenkins.xml**.

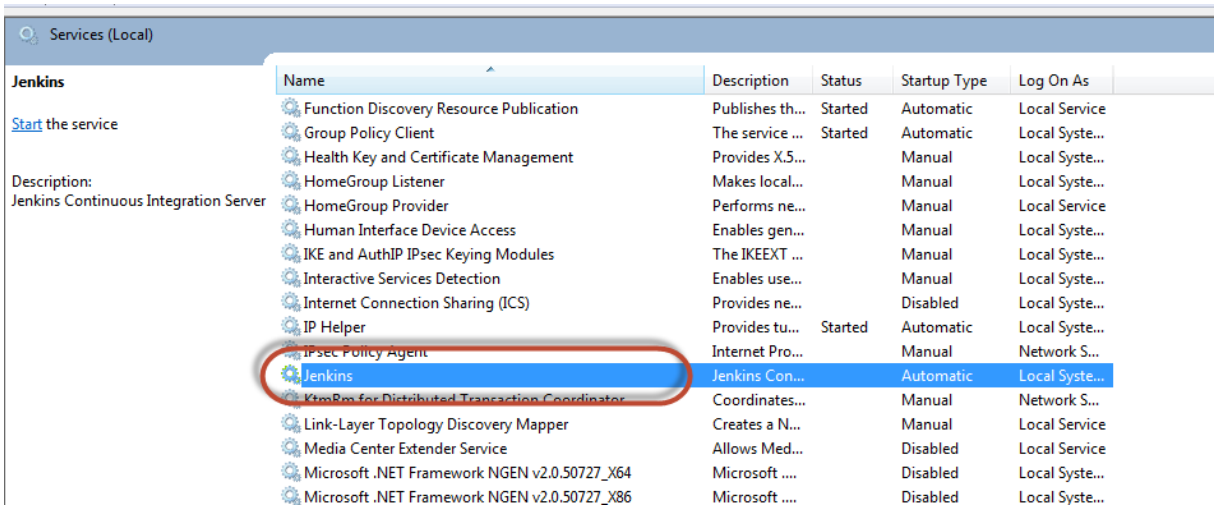


8. Change the **–httpPort** value from 8080 to your desired value. In my case we will use 8181.

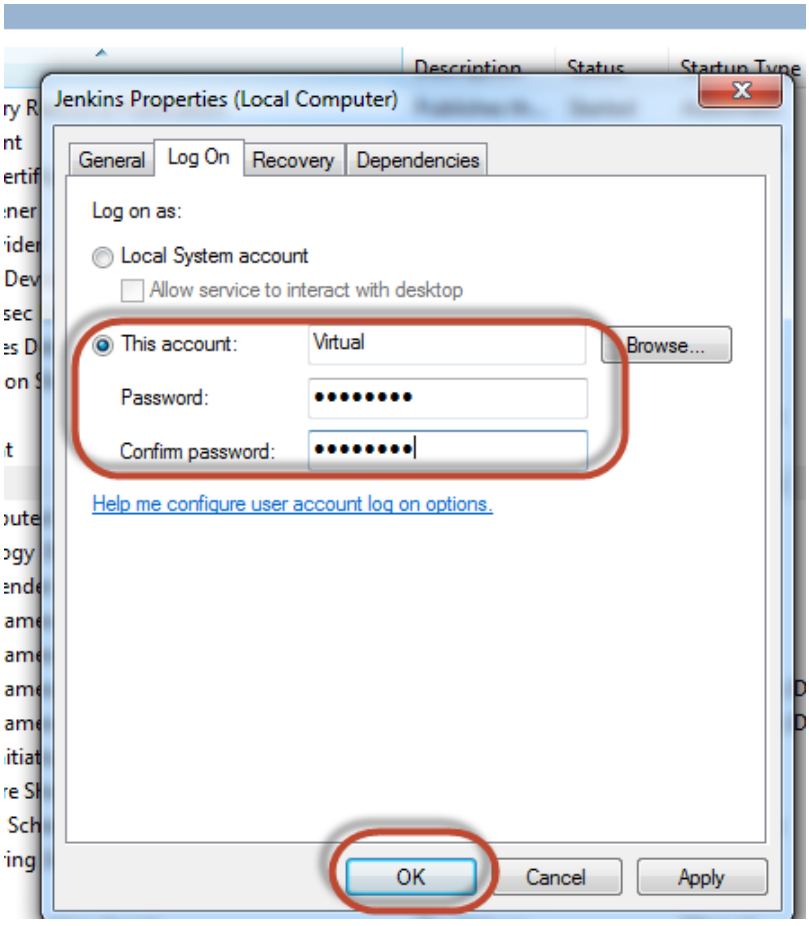


9. Now open Windows Services. We need to change the user who runs the service. By default, this user is set to the Local System Account, but this can be a problem because in this way Jenkins is not able to locate correctly the Maven repository folder. Indeed, we have it in the **c:\Users\<windows_user_name>\.m2** folder.

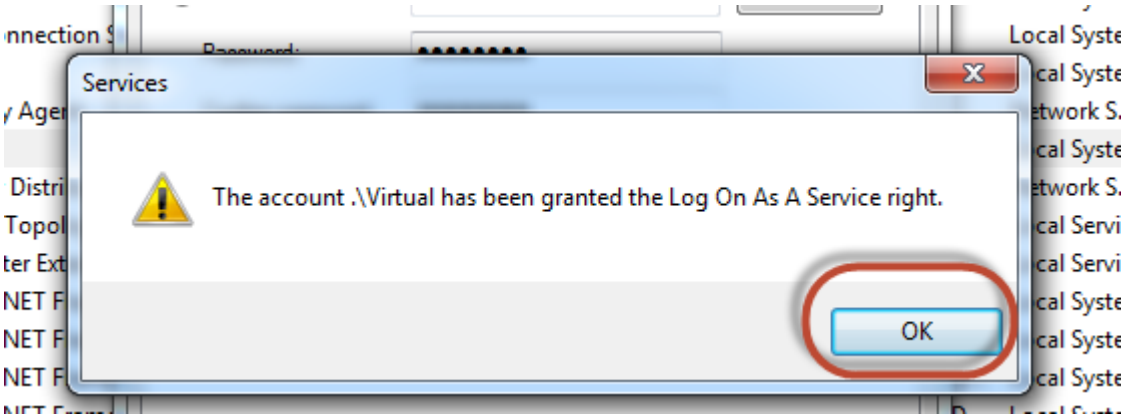
   Double click on the Jenkins service.



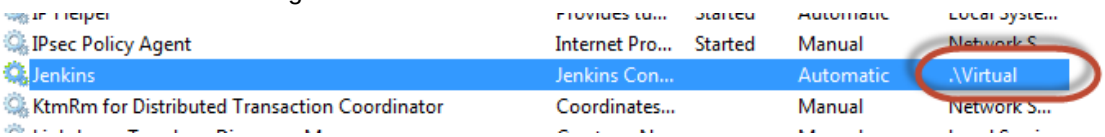
10. Change the Log On account to your **<windows_user_name>/<windows_user_password>** and click on **OK**.



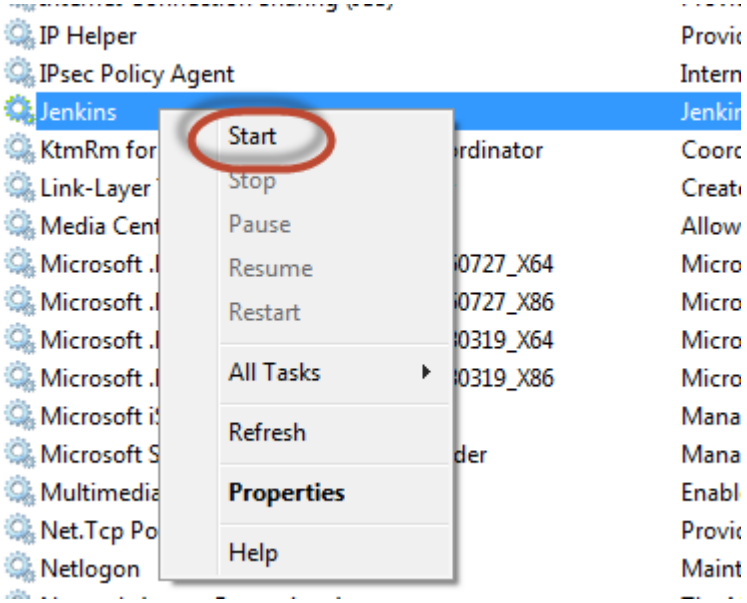
11. Click on **OK**.



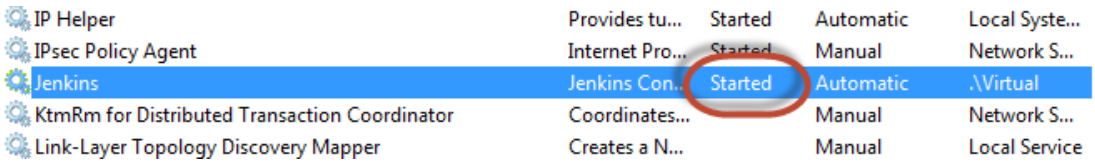
12. Jenkins has now a new Log On account.



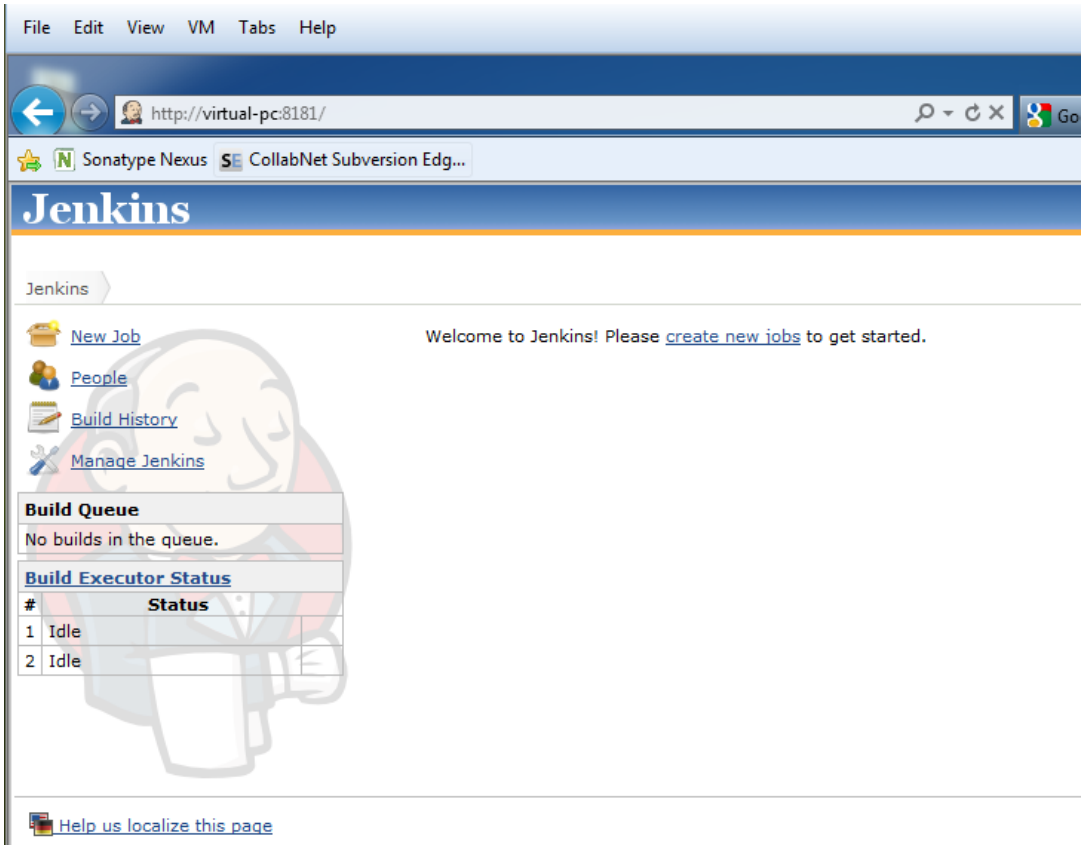
13. Finally we need to restart the service. Right click on Jenkins and choose **Start**.



14. In a few seconds the system will be ready again.



15. Go again on Jenkins main page using the link http://<server_name>:8181
(in my case it is http://virtual-pc:8181)



16. Click on **Manage Jenkins**.

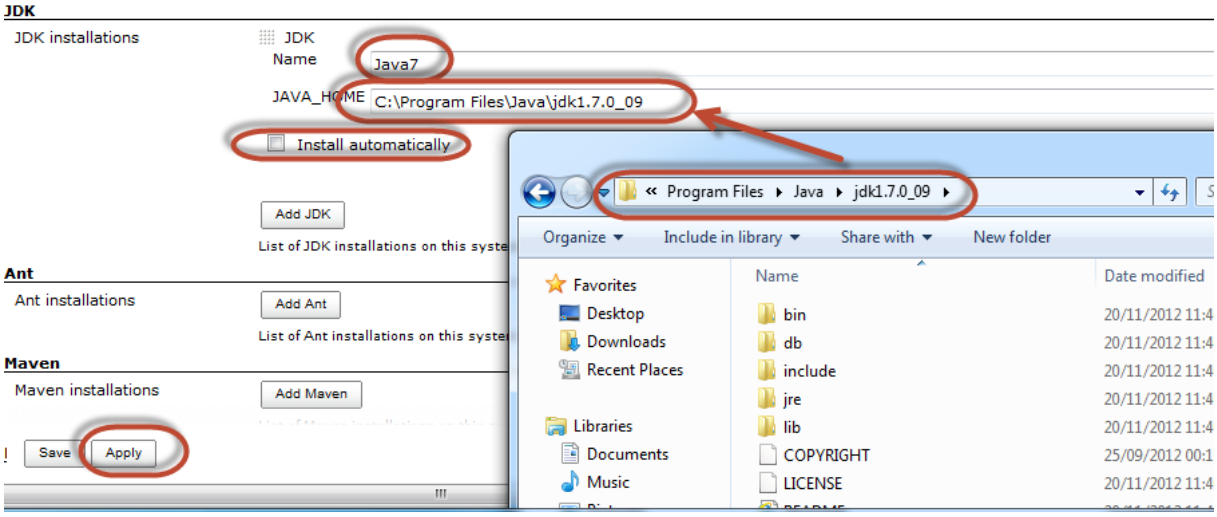
17. Click on **Configure System**.

18. Click on **Add JDK**.

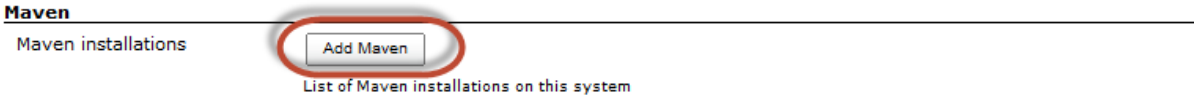
19. Uncheck first the **Install automatically** option and then provide the following information:

Field	Value
Name	Java7
JAVA_HOME	C:\Program Files\Java\jdk1.7.0_09

Click on **Apply**.



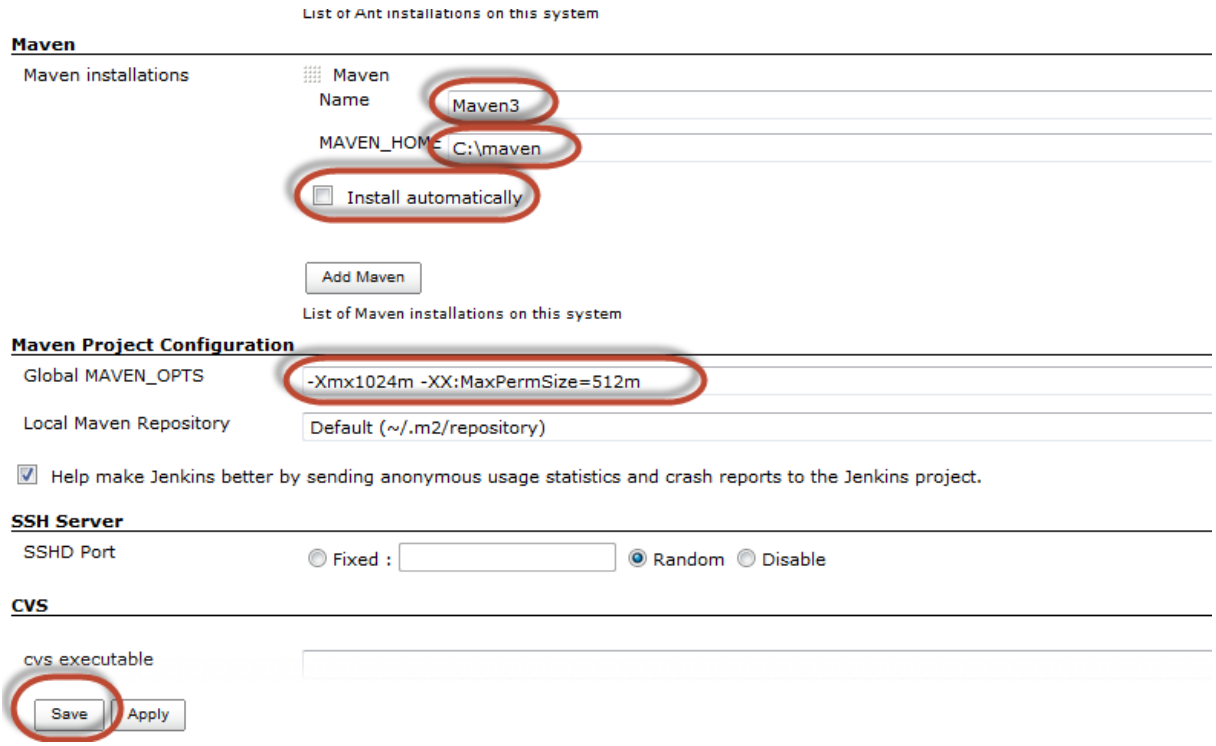
20. Click on **Add Maven**.



21. Uncheck first the Install automatically option and then provide the following information:

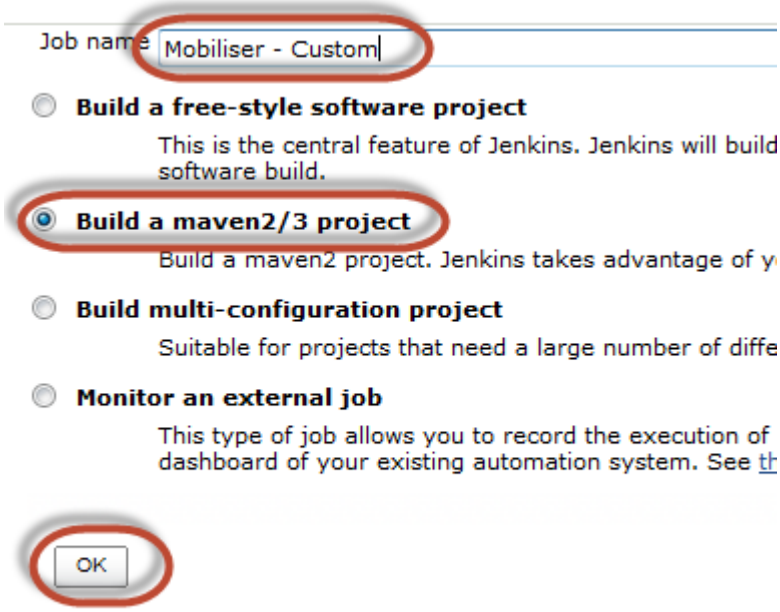
Field	Value
Name	Maven3
MAVEN_HOME	C:\maven
Global MAVEN_OPTS	-Xmx1024m -XX:MaxPermSize=512m
Local Maven Repository	Default

Click on **Save**.

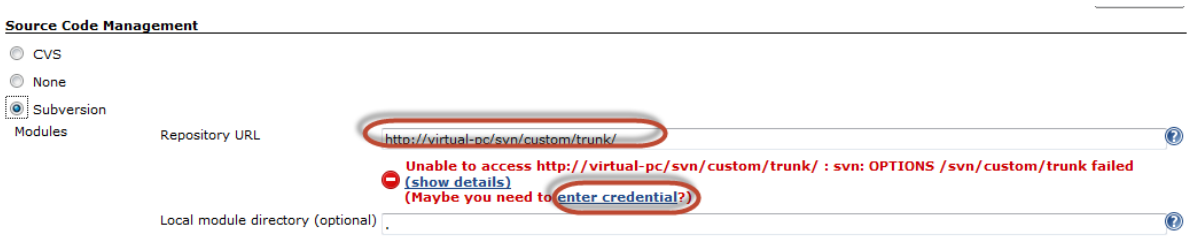


22. Now that Jenkins is configured, we are ready to create our first job. Click on **New Job**.

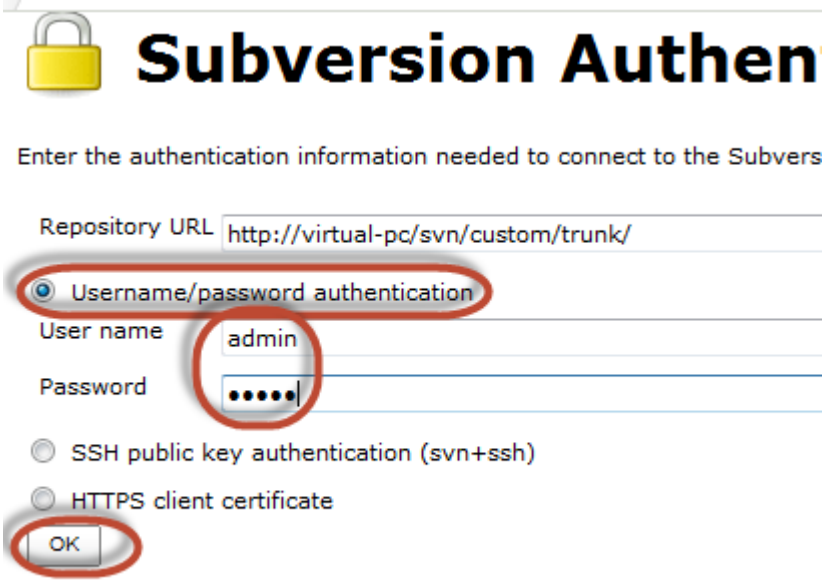
23. Assign a name to your new job (i.e. **Mobiliser – Custom**), select the option **Build a maven 2/3 project** and click on **OK**.



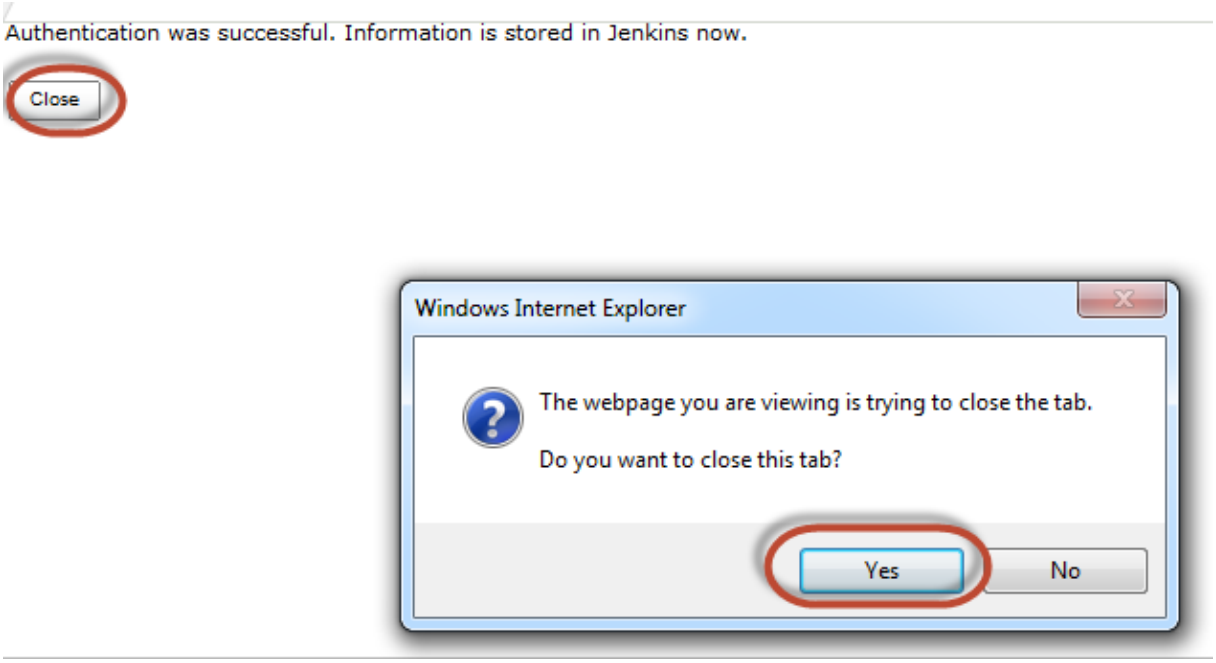
24. Select the Subversion source code management tool; provide the path to your Subversion repository http://<server_name>/svn/<repository_name>/trunk in the Repository URL field. Click on **enter credential.**



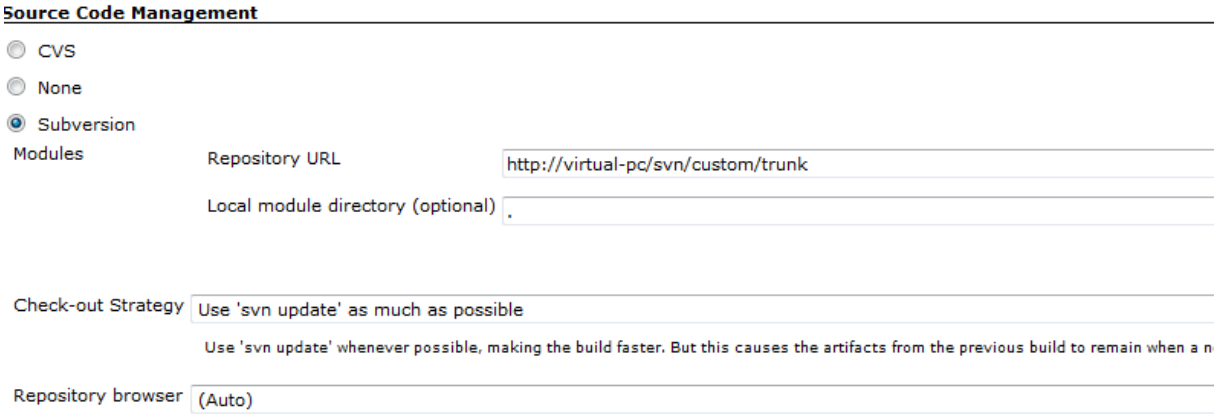
25. Provide the right credentials for accessing this repository and click on **OK**.



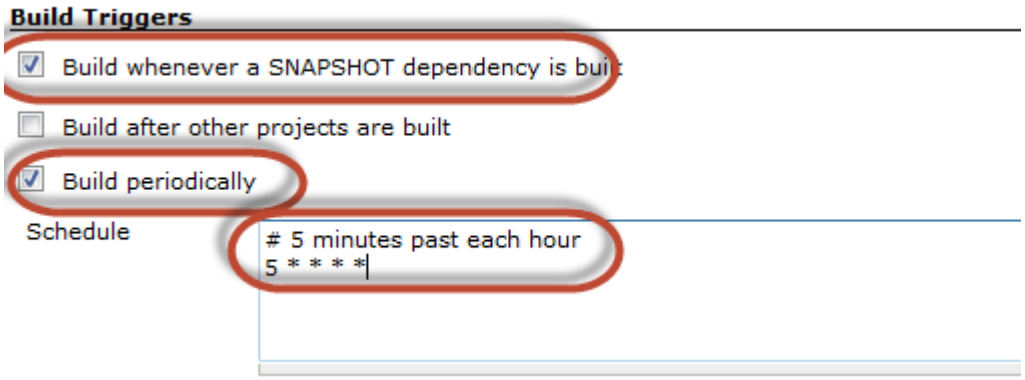
26. Click on **Close** and then on **Yes**.



27. Go back on the previous page and check that all is fine.



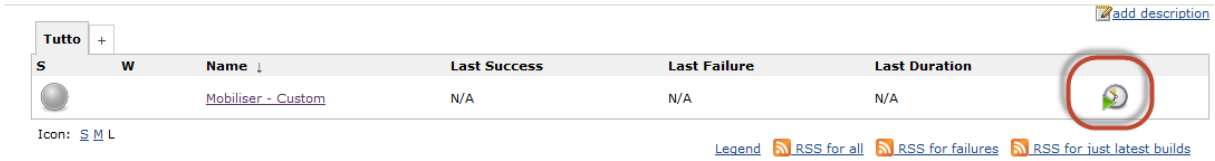
28. Now we need to enable some build triggers. Just for this example we are enabling the first option and the ‘Build periodically’ option. This last option requires also some schedule settings which follow the “cron” standards. Please refer to the “cron” guide to know more about the commands to use. In my case, I’m scheduling the build to start every hour, at 5 minutes past the hour.



29. When you have finished, click on **Save**.

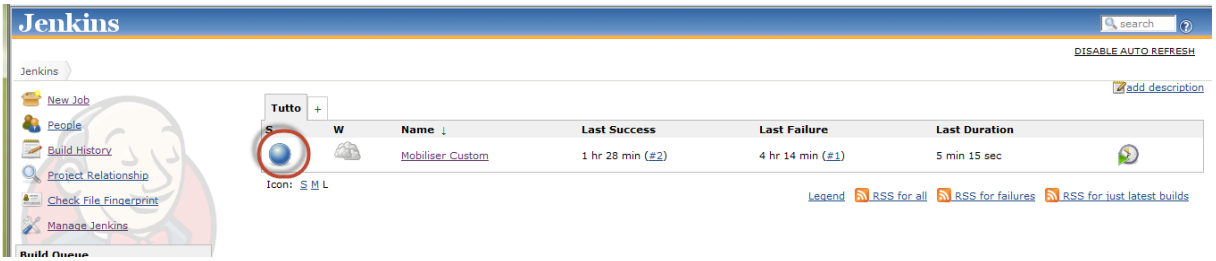
30. Go back to the dashboard.

31. Just to test the job, you can run it manually by clicking on the button with the green arrow.

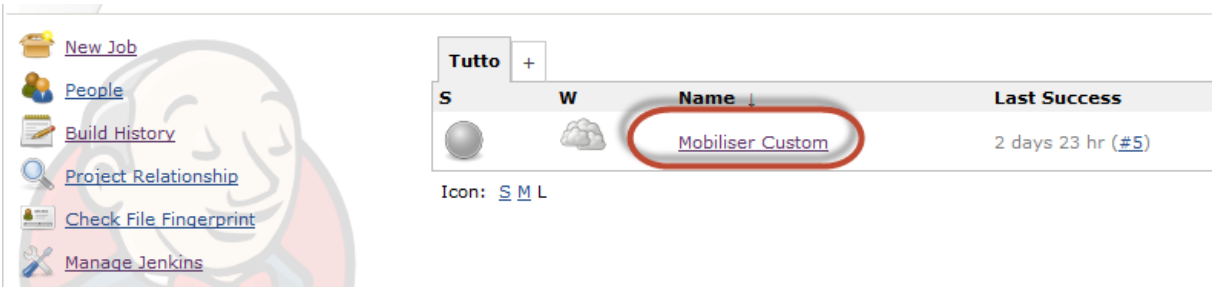


32. During the process, you will see a blinking icon and a progress bar on the left.

33. If the build finishes successfully, you should see a blue icon, with some other details.



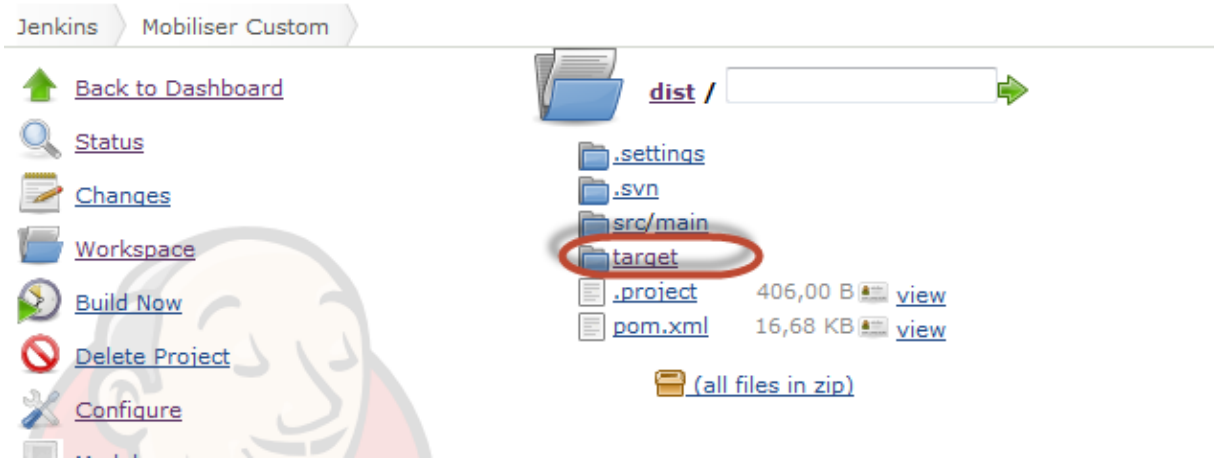
34. Once the build is finished successfully, you can check the output by clicking on the job name.



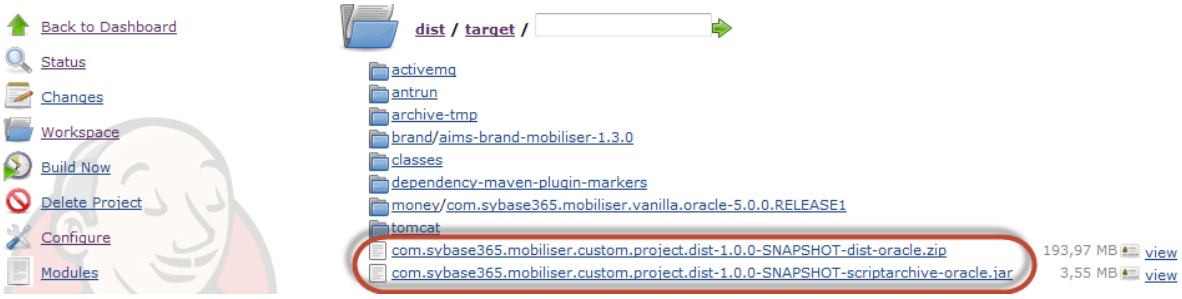
35. Then click on **Workspace**.

36. Select **Dist**.

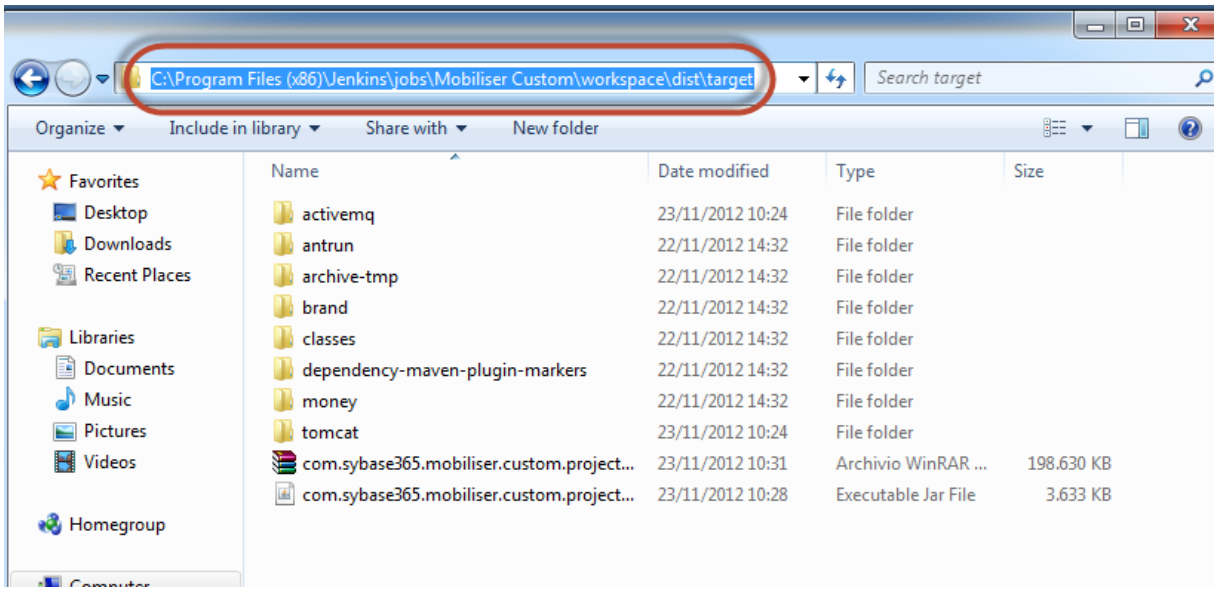
37. Select **Target**.



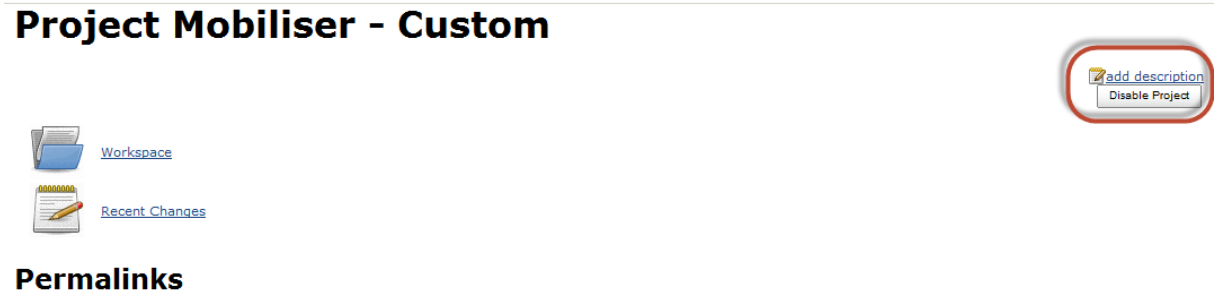
38. Here you can find the two file that Jenkins built for you.



39. Physically the files are located in the following Jenkins folder.



40. If a particular job needs to be disabled for some reason that can be done by clicking on the **Disable Project** button.



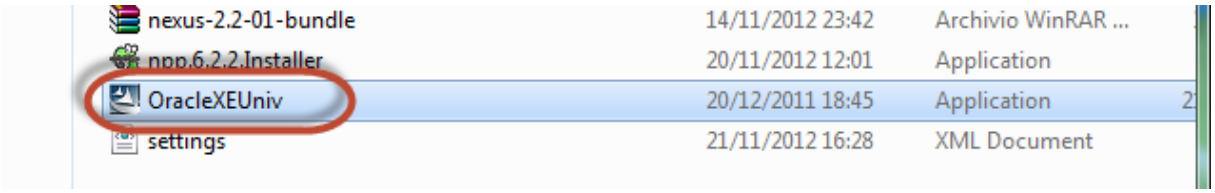
4.6 Install Oracle Express XE

1. The XE version of Oracle Database can be downloaded from the following web site:

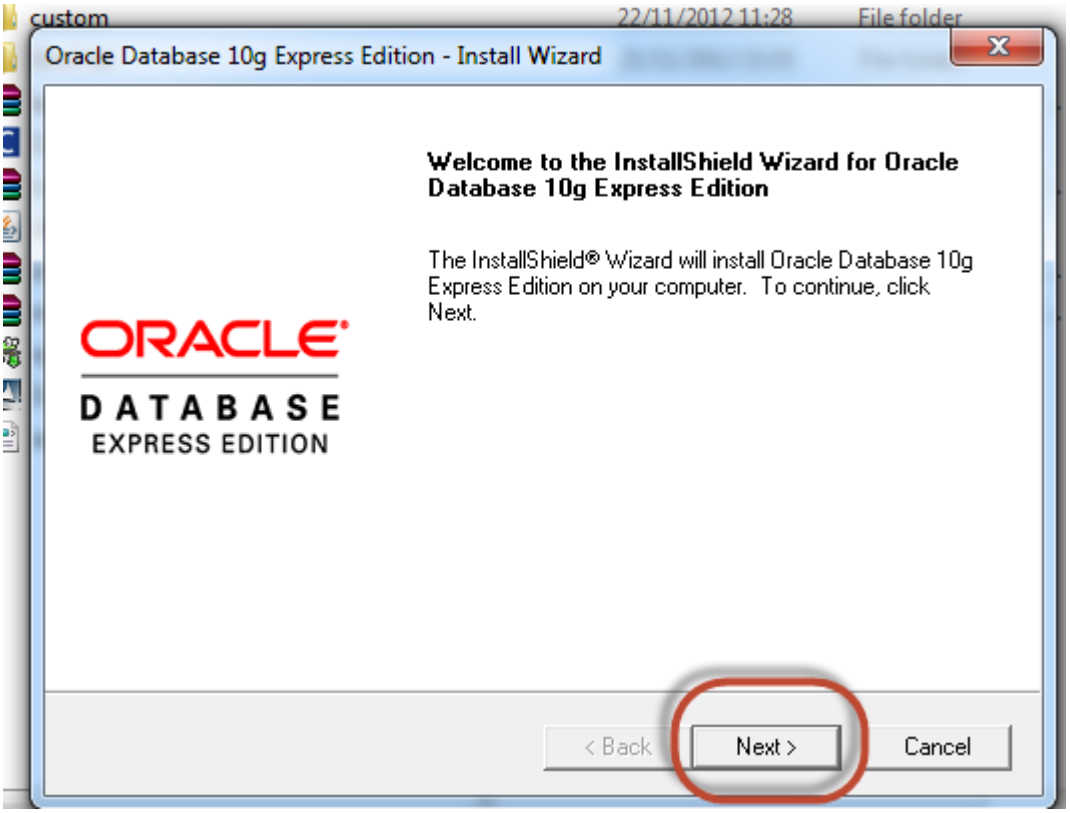
   http://www.oracle.com/technology/software/products/database/xe/index.html

2. For our current document, we are using the Oracle Database 10g Express Edition. You can find a link to a copy of the installation package at the end of this document. Probably on the web site you can find a more recent release, but it should be fine as well.

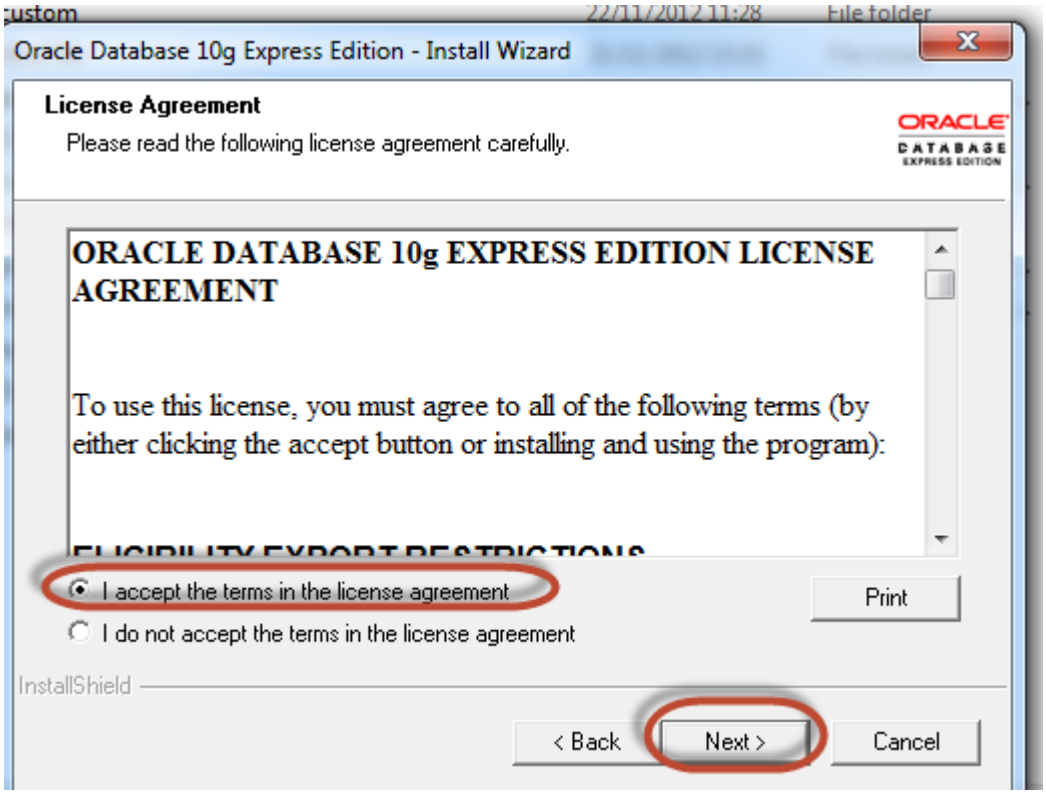
   Double click on the OracleXEUniv.exe file to start the installation.



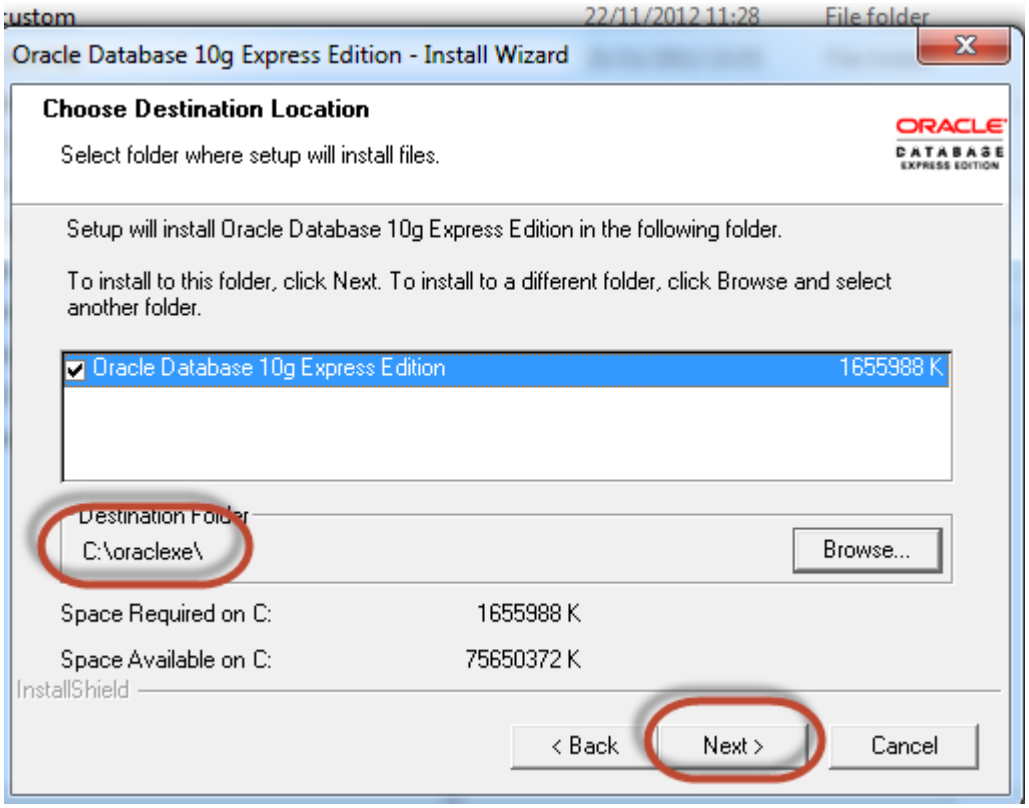
3. Click on **Next**.



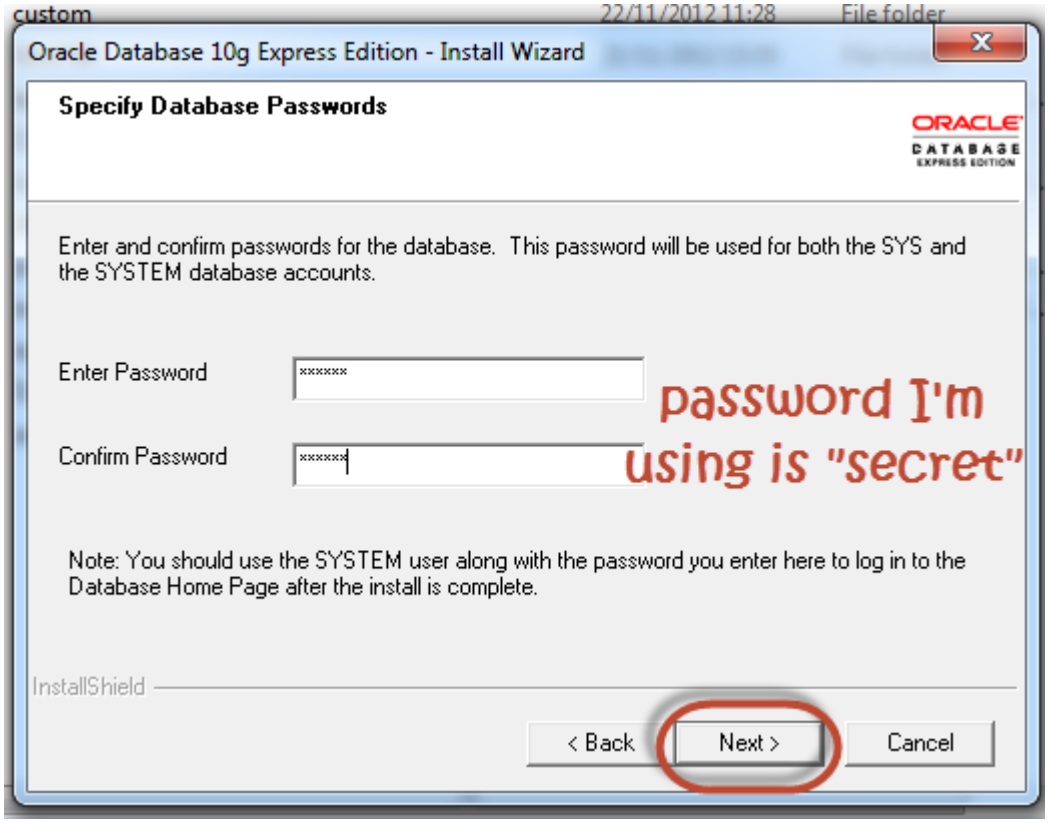
4. Accept the license agreement and click on **Next**.



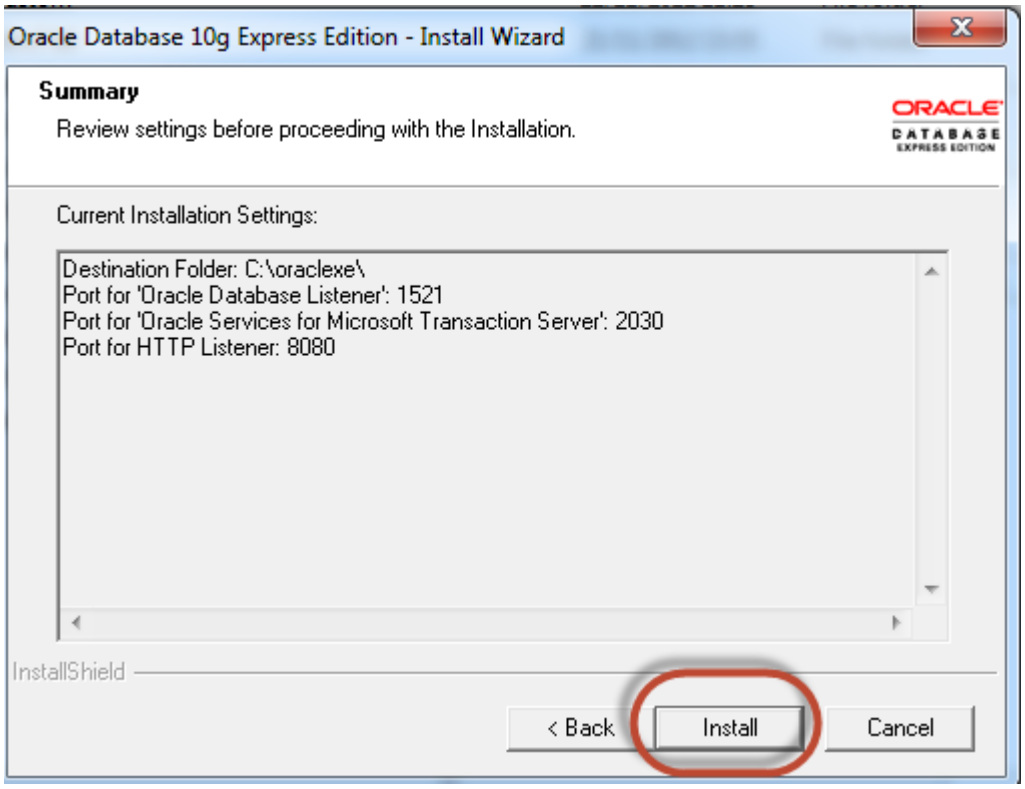
5. Provide a valid installation path or simply leave it as by default and click on **Next**.



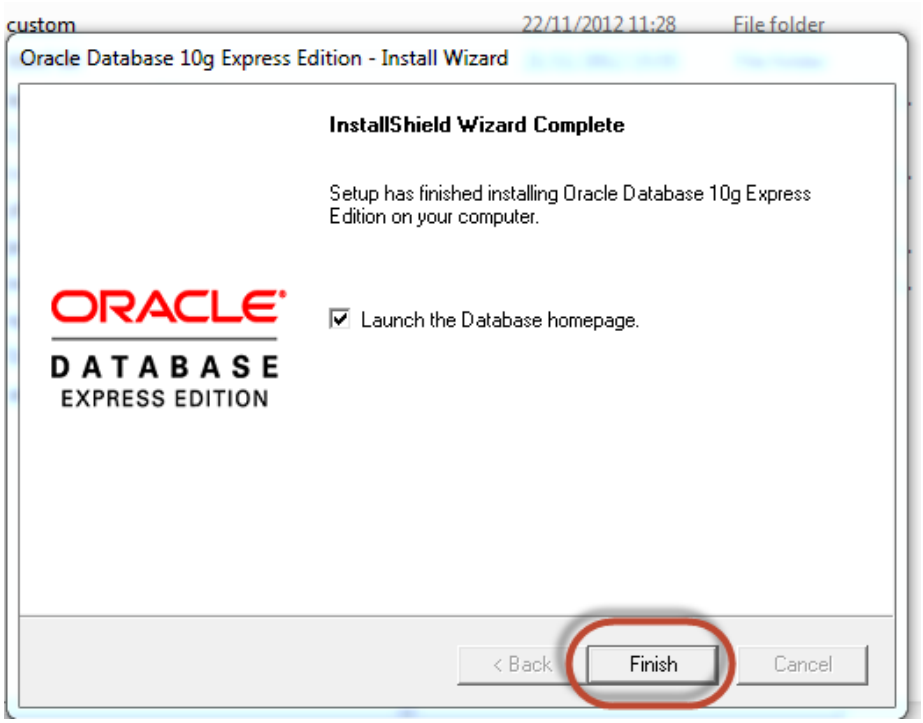
6. Enter the password that will be used for the SYS and the SYSTEM accounts in the Oracle database. We are using the password “secret”. Click on **Next**.



7. Click on **Install**.



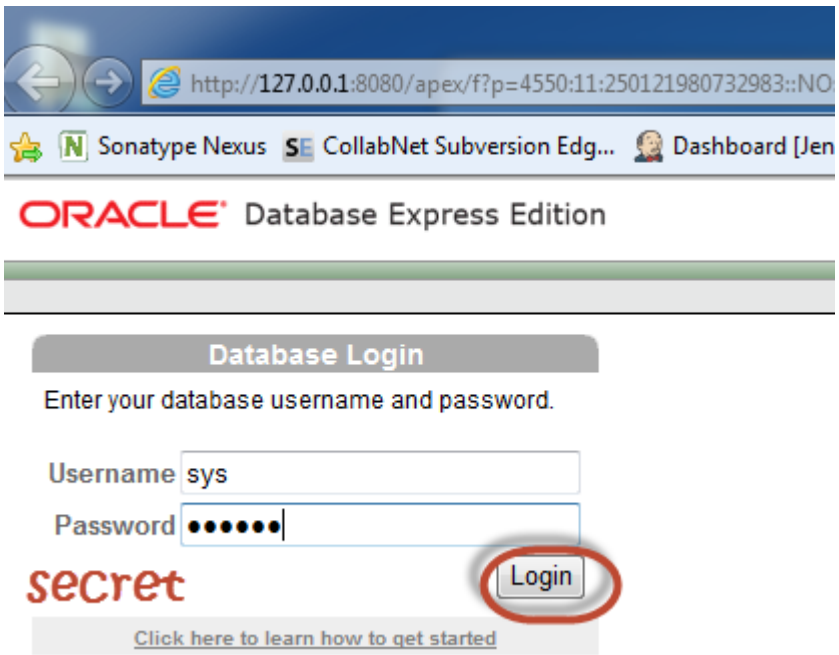
8. Click on **Finish**.



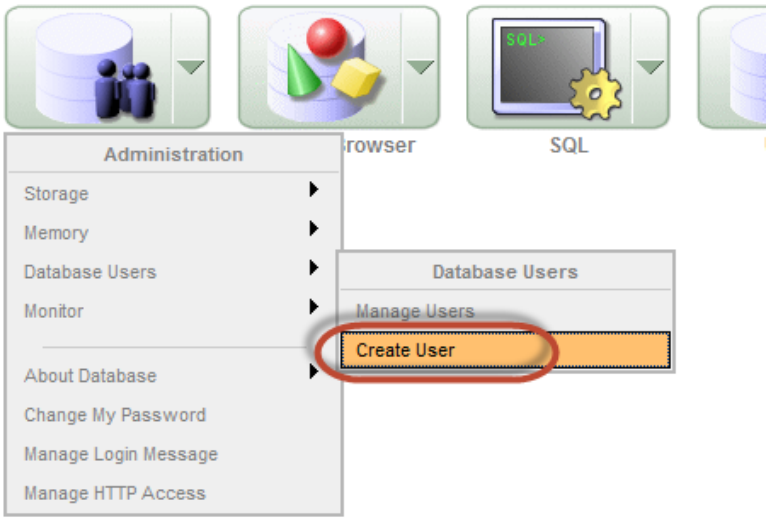
9. Enter the account

USER	PASSWORD
sys	secret

we set before and click on **Login**.



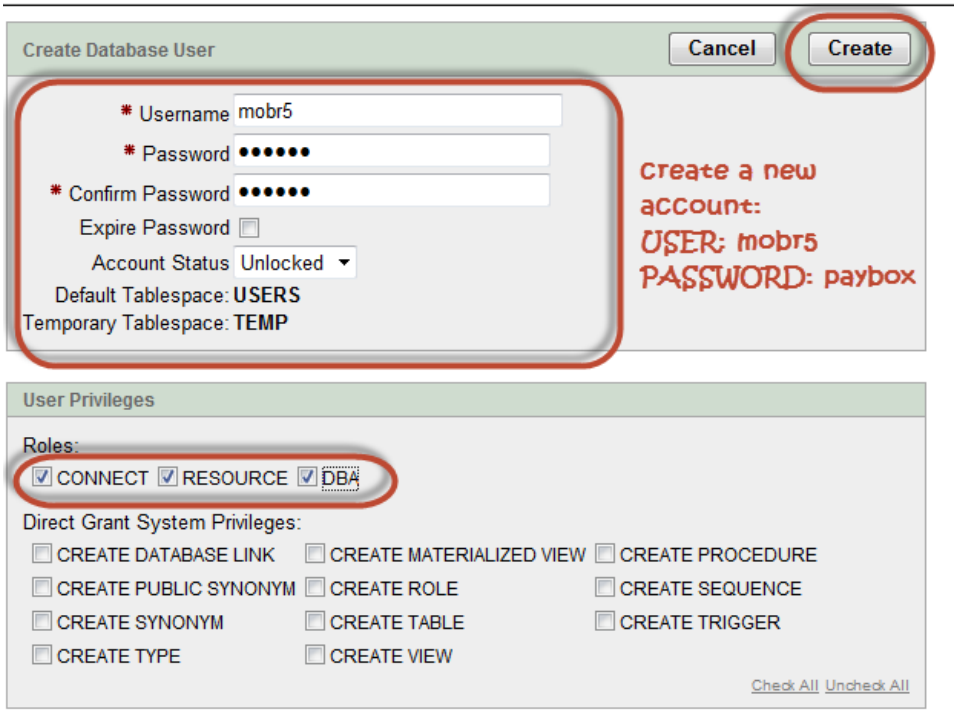
10. The administration home page is automatically launched. Click on **Administration → Database Users → Create User**.



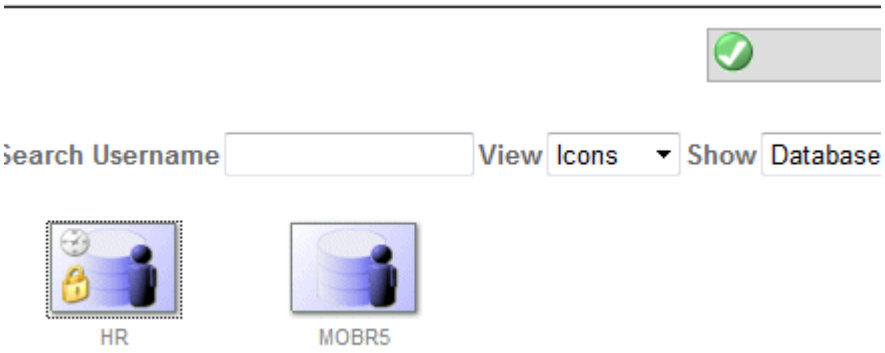
11. Enter the following user:

USER	PASSWORD
mobr5	paybox

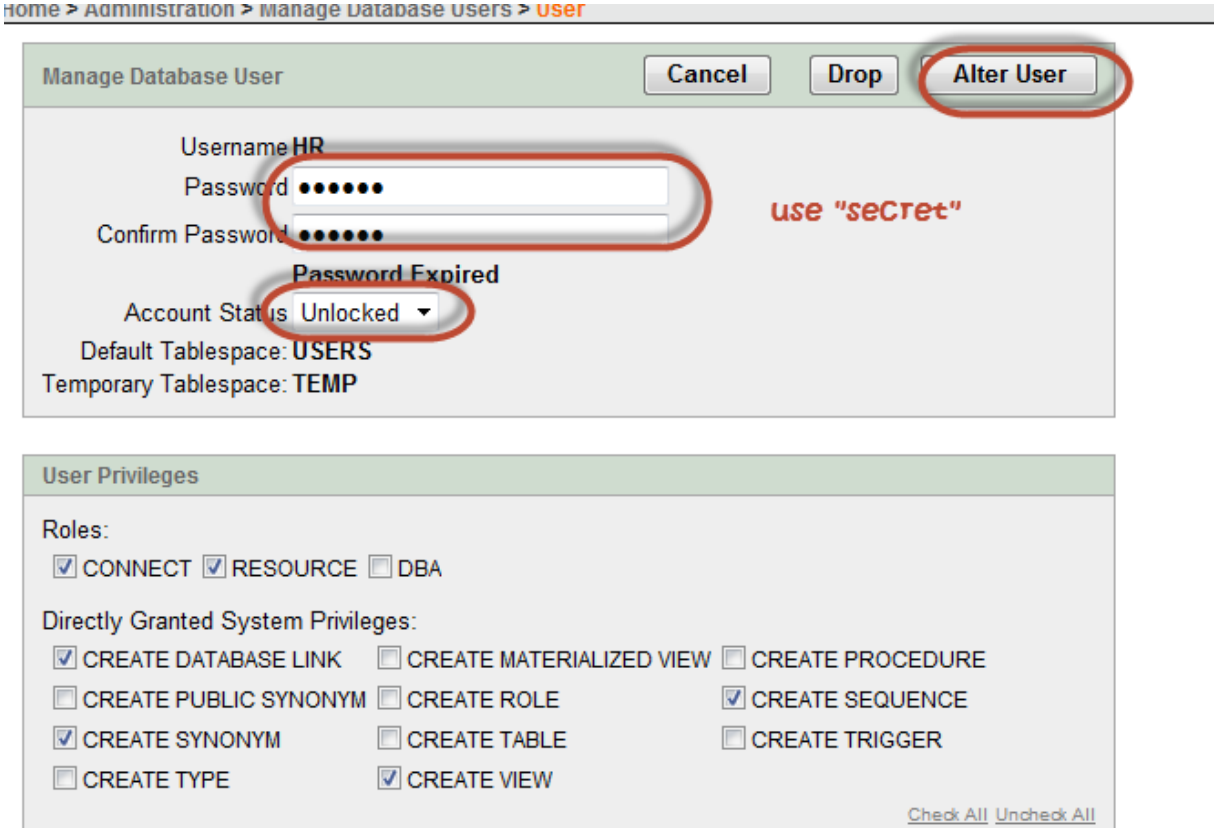
Uncheck the **Expire Password** option, set the **Account Status** to **Unlocked**, check the **DBA** role and click on **Create**. We are using this user because, in this way, we won't have to change it in the **dbmaintain.properties** file and in the Mobiliser database itself.



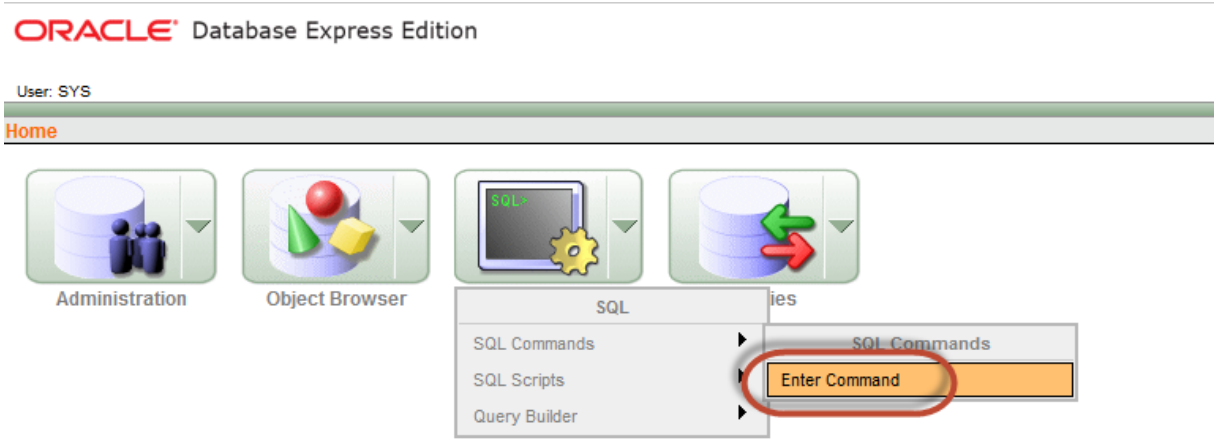
12. You may want also to unlock the **HR** user for the sample database. This user is used for creating other databases. For security reasons, it is by default locked. We can just unlock it and set a new password, so click on the **HR** user.



13. Set the new password (we are using “secret” here), set the user as **Unlocked** and click on **Alter User**.



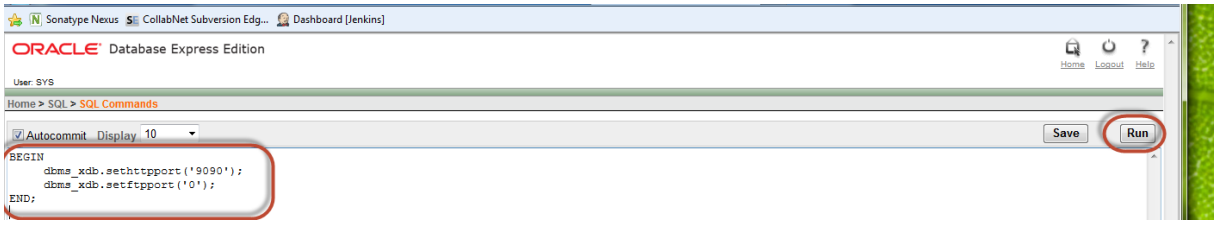
14. Oracle database has been installed on port 8080. As you know, this port will also be used by Mobiliser, so we will change this port for Oracle. In order to do this, you need to click, from the home page, on **SQL → SQL Commands → Enter Command**.



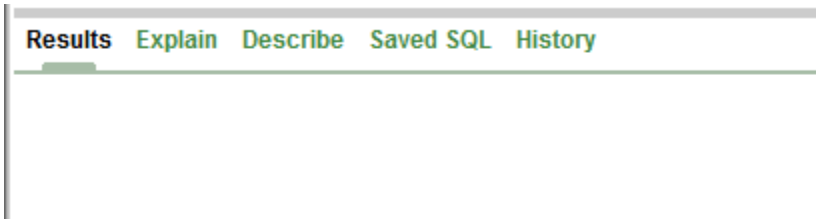
15. Enter the following command in the form and click on **Run**.

```
BEGIN
    dbms_xdb.sethttpport('9090');
    dbms_xdb.setftpport('0');
END;
```

This will change the port from 8080 to 9090 inside the database. You can choose any other port you want to use.



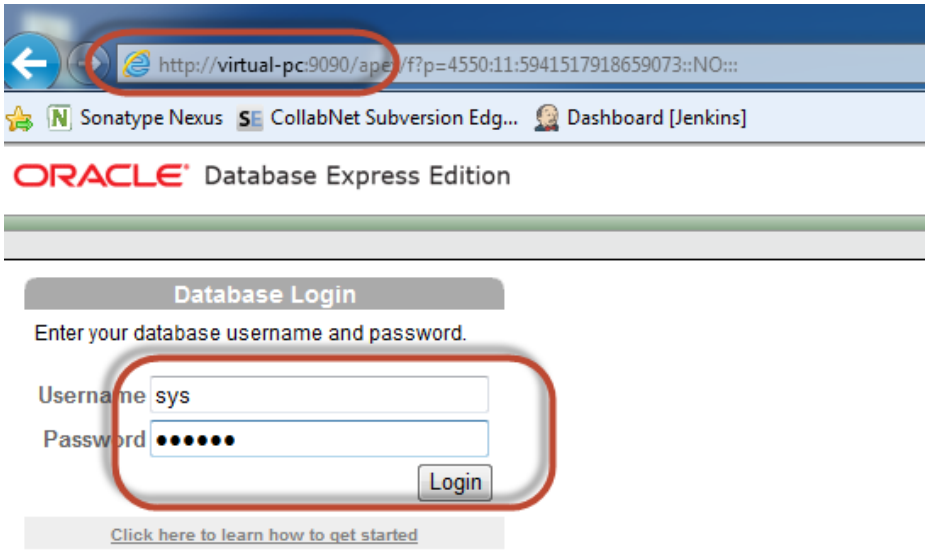
16. The command starts and it will be finished when the **Results** window will be blank.



17. The port is now changed. You can close you browser and connect again to the following web site: http://<server_name>:9090/apex
(in my case it will be: http://virtual-pc:9090/apex)

18. Type again the credentials for the **SYS** account and click on **Login**.

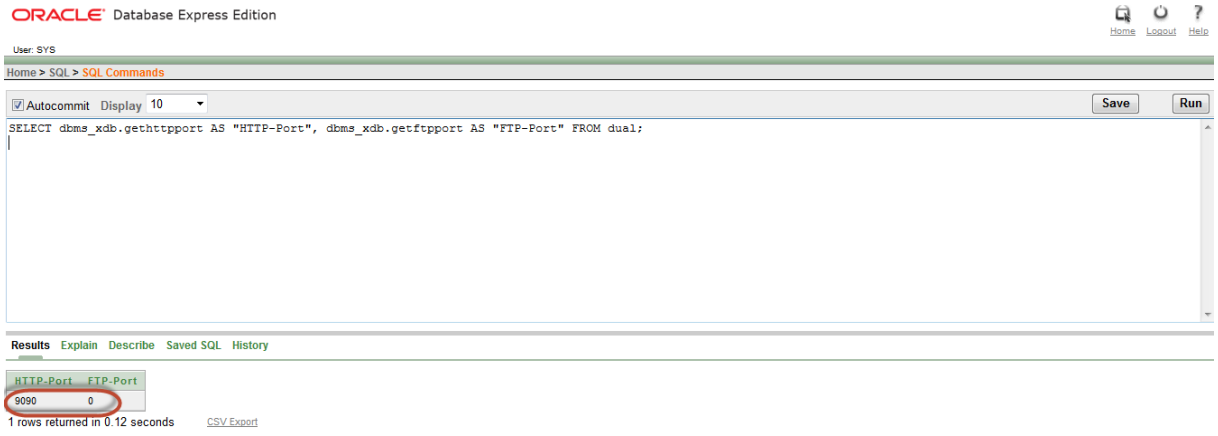
USER	PASSWORD
sys	secret



19. The following command can be used to check that the port is correctly changed for the entire database.

SELECT dbms_xdb.gethttpport AS "HTTP-Port", dbms_xdb.getftpport AS "FTP-Port" FROM dual;

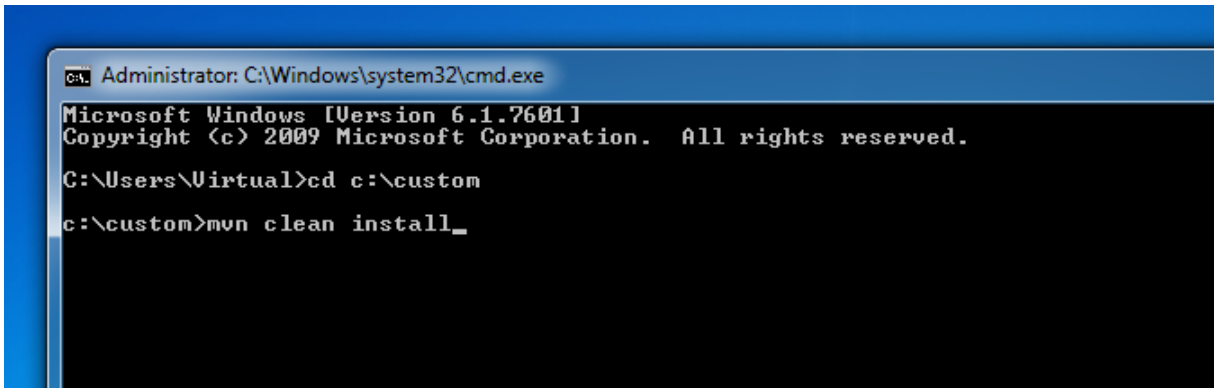
You will get the following result.



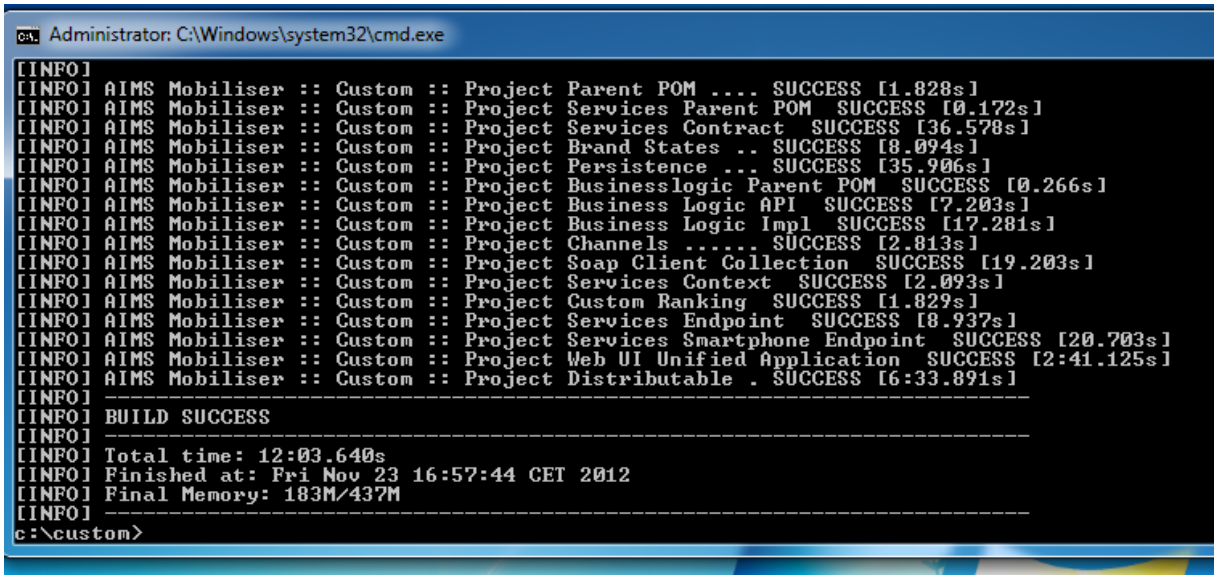
20. Your Oracle Database has been successfully installed.

4.7 Install the Mobiliser database

1. In order to run Mobiliser we need to install its database. The instructions to create the entire database, that is the complete schema, is automatically generated by the Maven “install” command and it’s put in a .jar file that you can execute in order to recreate the database with all the objects and the required data records. This means that if you have not executed yet the “mvn clean install” command, it’s time to do it now. Open a WCMD, move to the folder where you have your source project and run the command “mvn clean install”.



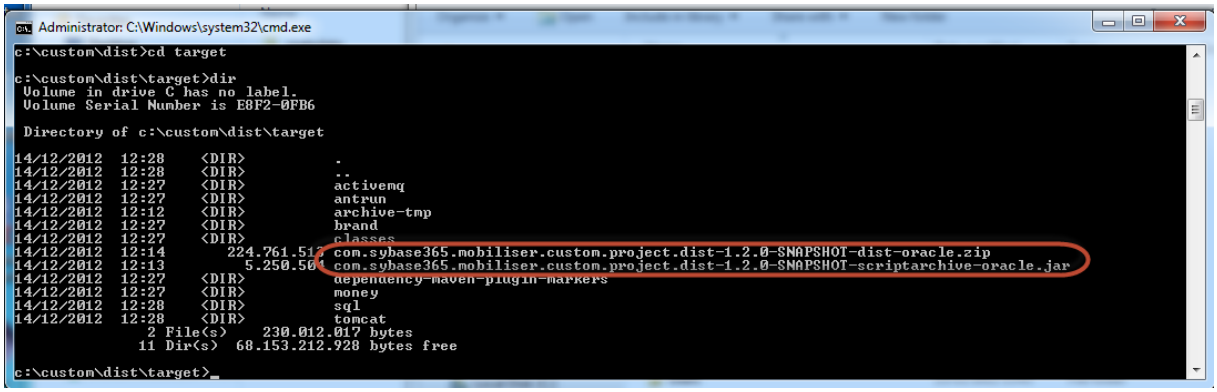
2. The process finished successfully.



3. At the end of the process, if you go inside the **c:\custom\dist\target** folder you will see two main big files:

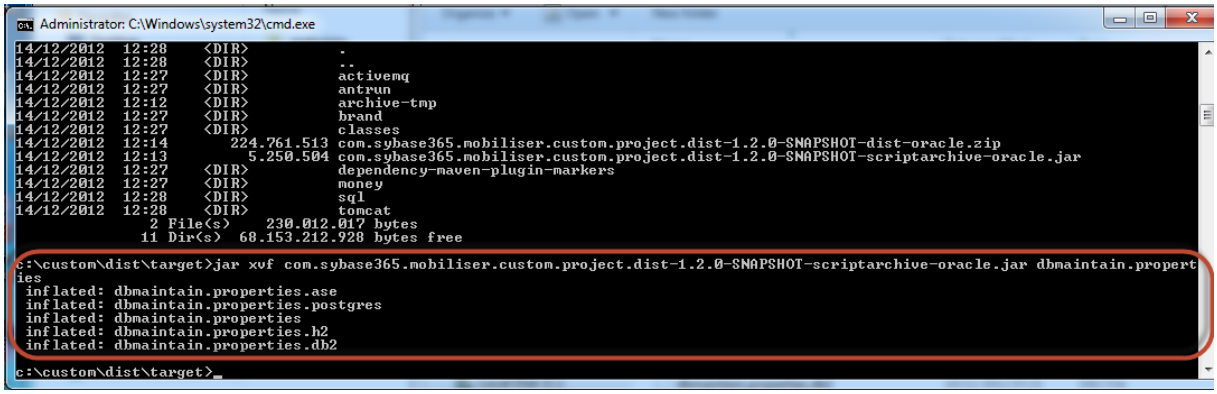
   com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-dist-oracle.zip
   com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-scriptarchive-oracle.jar

   The second file is the one that we will use to generate the database.



4. Before running the creation of the database we need to tell to the DB generator where it has to create the Mobiliser database and also some other settings. These settings are located in a file called **dbmaintain.properties**. This file is automatically checked by the DB generator when it starts. This file is contained in the .jar file we mentioned above, so we need just to extract it and use it. In order to extract this file you need to run the following command from the WCMD in the **c:\custom\dist\target** folder:

   jar xvf com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-scriptarchive-oracle.jar dbmaintain.properties



5. After the extraction you will find in the current folder some other files. You can delete all the dbmaintain.properties files with the extensions ".ase, .h2, .db2, .postgres" because we don't need them. We just need the file **dbmaintain.properties** located in the current folder. For ease of use, we can leave the file in the current position.

6. If you look at the content of this file, you see there are a lot of commands which will be launched against the database. In particular, on top, you may notice that it contains the name of the user and the password we have previously created on Oracle.

```
8
9   ### Database connections ###
10
11  database.dialect=oracle
12
13  database.driverClassName=oracle.jdbc.driver.OracleDriver
14  database.url=jdbc:oracle:thin:@localhost:1521:xe
15
16  database.userName=mobr5
17  database.schemaNames=mobr5
18  database.password=paybox
19
```

7. One of the required changes to this file is the specification of the driver for the database you want to use. This can be made by changing the property database.driverLocation in the following way:

   database.driverLocation=<path to the>/ojdbc14.jar

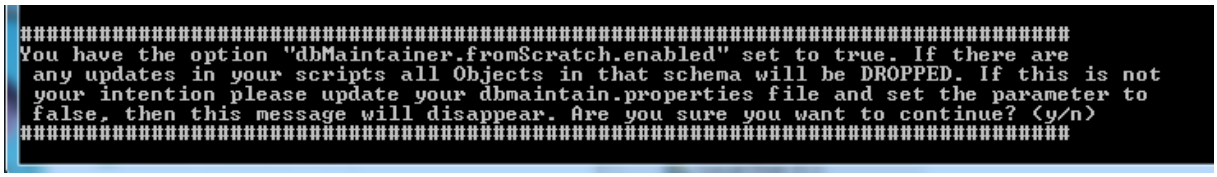
   (in my case it is database.driverLocation=C:/oraclexe/app/oracle/product/10.2.0/server/jdbc/lib/ojdbc14.jar)
   The file should look as in this screen shot.

```
1   ########################################
2   # Default configuration of DbMaintain #
3   ########################################
4
5   # This file contains default configuration values for dbmaintain. This file should not be edited.
6   # All properties in this file can be overridden, either in the project specific properties file
7   # (dbmaintain.properties)
8
9   ### Database connections ###
10
11  database.dialect=oracle
12
13  database.driverClassName=oracle.jdbc.driver.OracleDriver
14  database.url=jdbc:oracle:thin:@localhost:1521:xe
15
16  database.userName=mobr5
17  database.schemaNames=mobr5
18  database.password=paybox
19
20  #Must be set if the driver is not packaged inside the scriptarchive or is present on the classpath
21  #e.g. /path/to/ojdbc14.jar
22  database.driverLocation=C:/oraclexe/app/oracle/product/10.2.0/server/jdbc/lib/ojdbc14.jar
23
24  # Comma separated list of locations where database scripts can be found. This list may contain folders as
25  dbMaintainer.script.locations=scripts
26
```

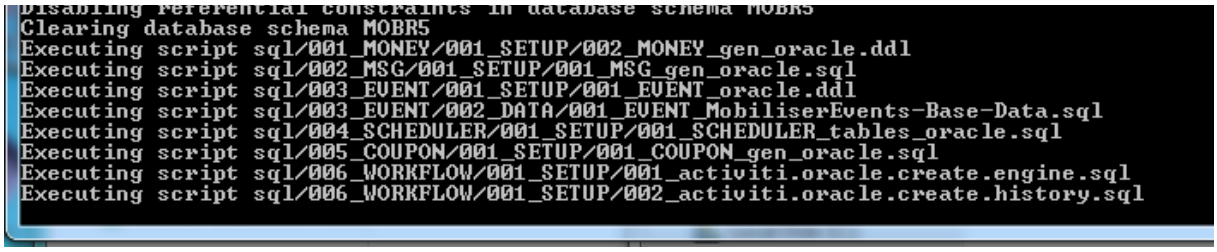
8. Once saved the file, you can start creating the Mobiliser database. Go back on the WCMD and, from the folder **c:\custom\dist\target**, run the following command:

   java -jar com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-scriptarchive-oracle.jar

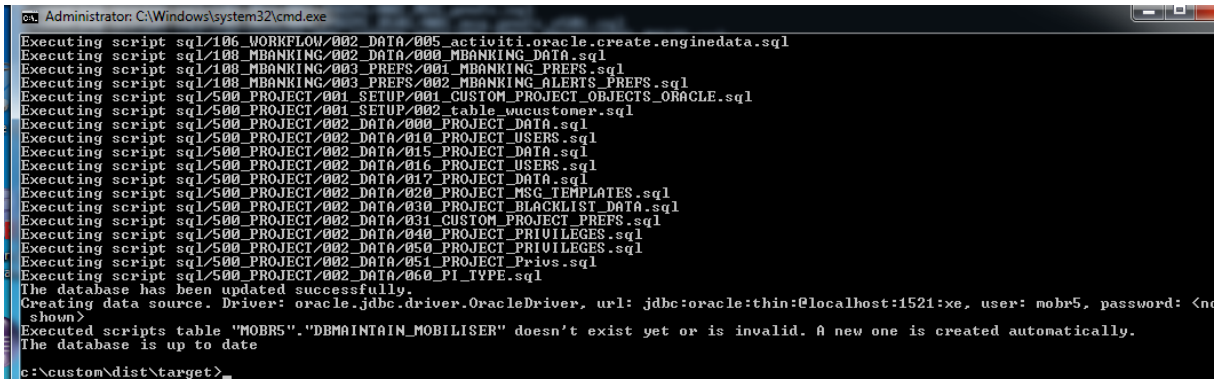
9. You are prompted to confirm if you want to continue since, if a database already exists, it will be overwritten. For our scope, answer "**y**".



10. The DB creation starts.

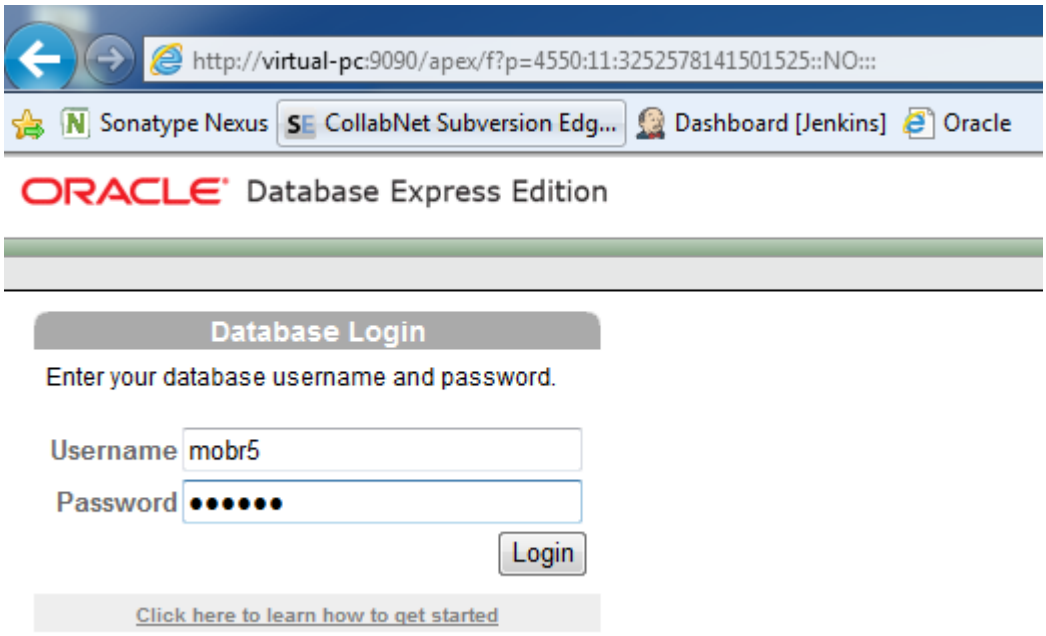


11. After a few seconds you should get the following messages. The database has been created successfully.

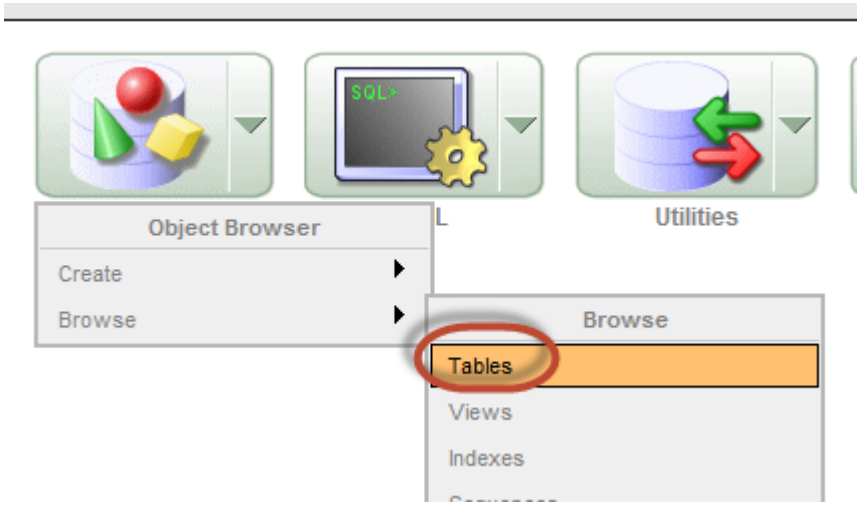


12. You can also browse the DB by opening the Oracle administration page and logging in with the **mobr5** user.

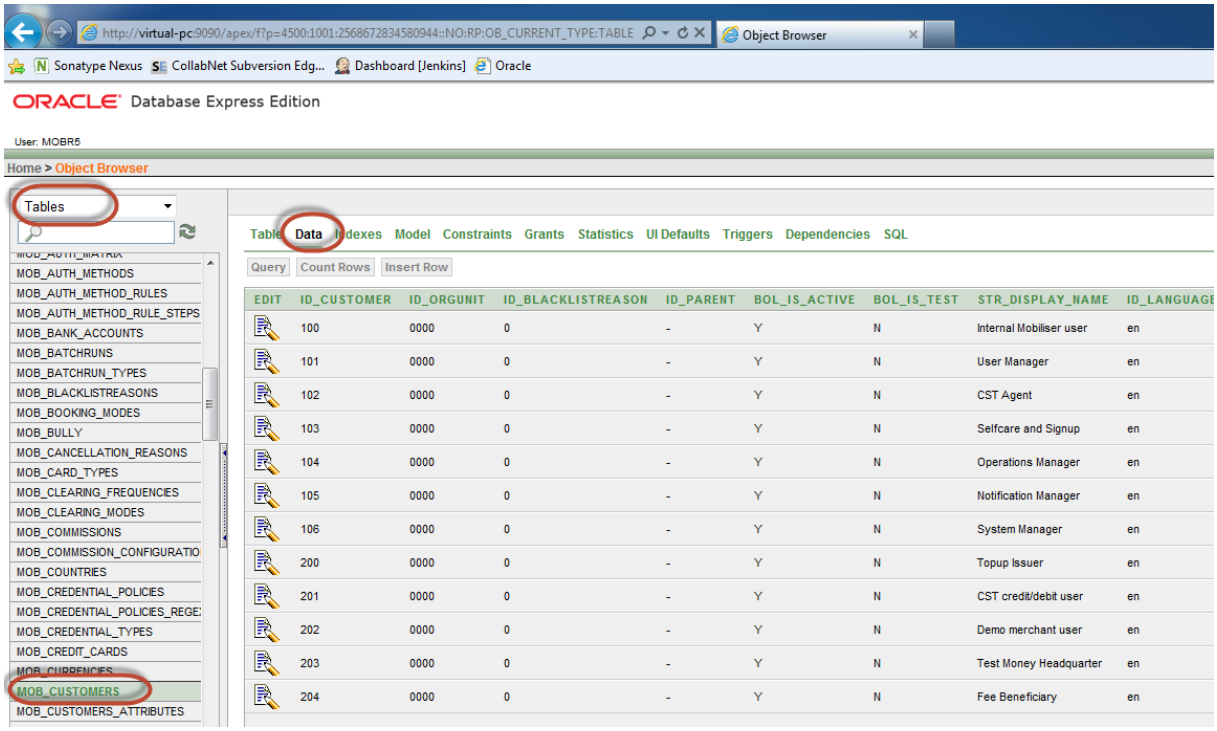
USER	PASSWORD
mobr5	paybox



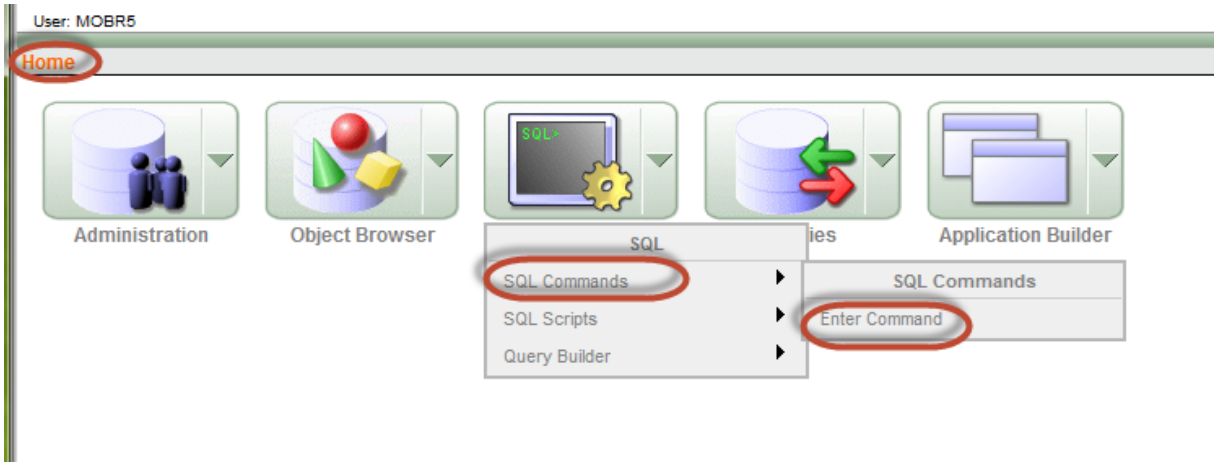
13. Select **Object Browser → Browse → Tables**.



14. Browse for the table **MOB_CUSTOMERS** and select **Data** on the right side. You should see all the already recorded customers for the database.

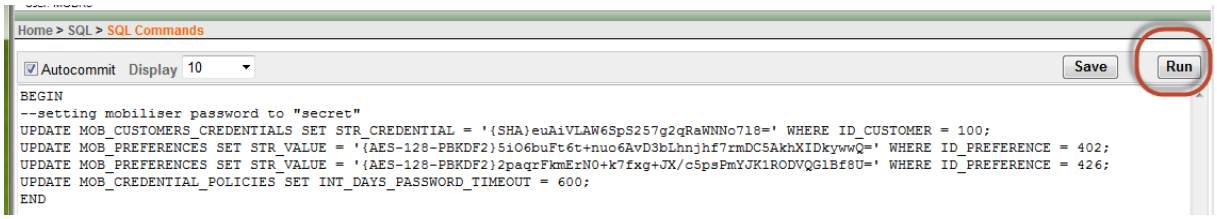


15. There is another change required in the database. We need to set the initial password for the “sysmgr” user to “secret”. In order to do this go on **Home →SQL → SQL Commands**

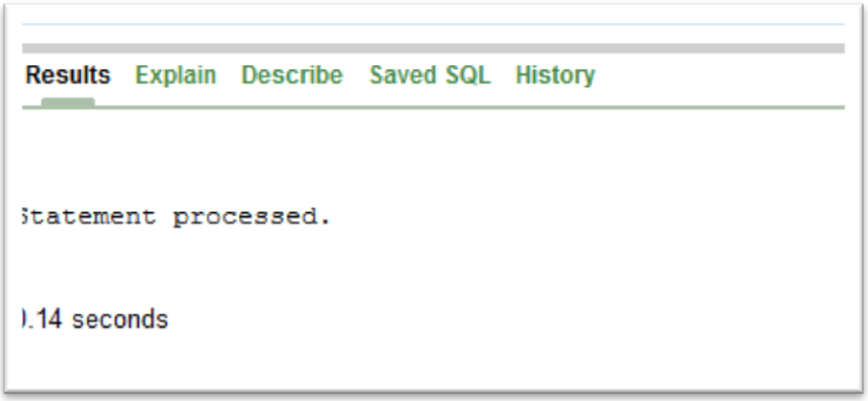


16. In the next screen, type the following code and click on **Run**:

```
BEGIN
UPDATE MOB_CUSTOMERS_CREDENTIALS SET STR_CREDENTIAL = '{SHA}euAiVLAW6SpS257g2qRaWNNo718='
WHERE ID_CUSTOMER = 100;
UPDATE MOB_PREFERENCES SET STR_VALUE = '{AES-128-
PBKDF2}5iO6buFt6t+nuo6AvD3bLhnjhf7rmDC5AkhXIDkywwQ=' WHERE ID_PREFERENCE = 402;
UPDATE MOB_PREFERENCES SET STR_VALUE = '{AES-128-
PBKDF2}2paqrFkmErN0+k7fxg+JX/c5psPmYJK1RODVQGlBf8U=' WHERE ID_PREFERENCE = 426;
UPDATE MOB_CREDENTIAL_POLICIES SET INT_DAYS_PASSWORD_TIMEOUT = 600;
END
```



17. If all worked fine, you should get a result like this:



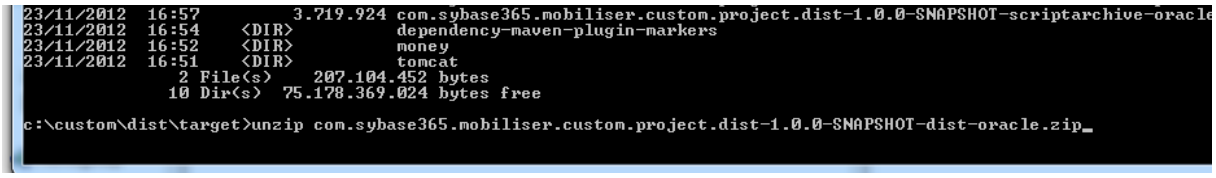
The password is successfully set.

18. Your Mobiliser database creation is finished.

4.8 Run Mobiliser

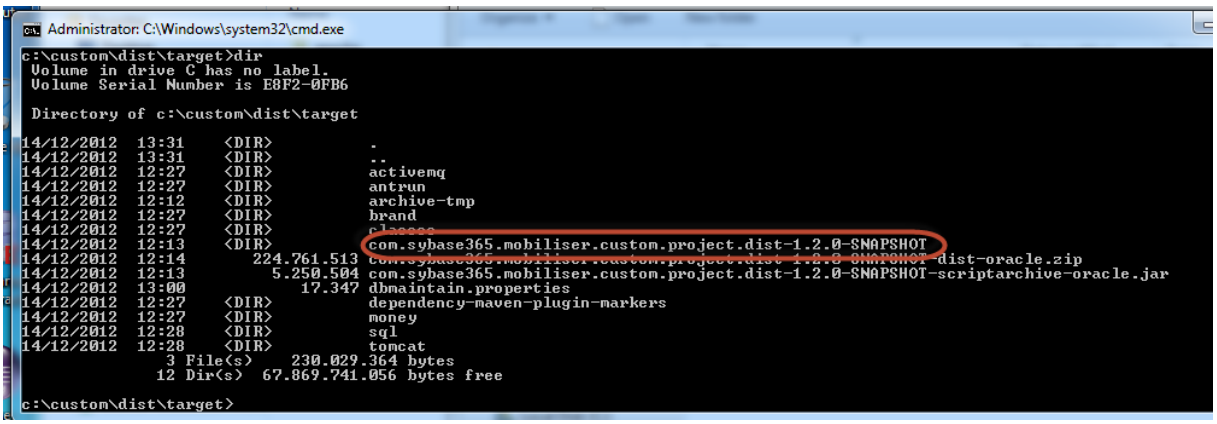
1. We are going to start Mobiliser. Open a WCMD, go to the folder **c:\custom\dist\target** and run the following command:

   unzip com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT-dist-oracle.zip

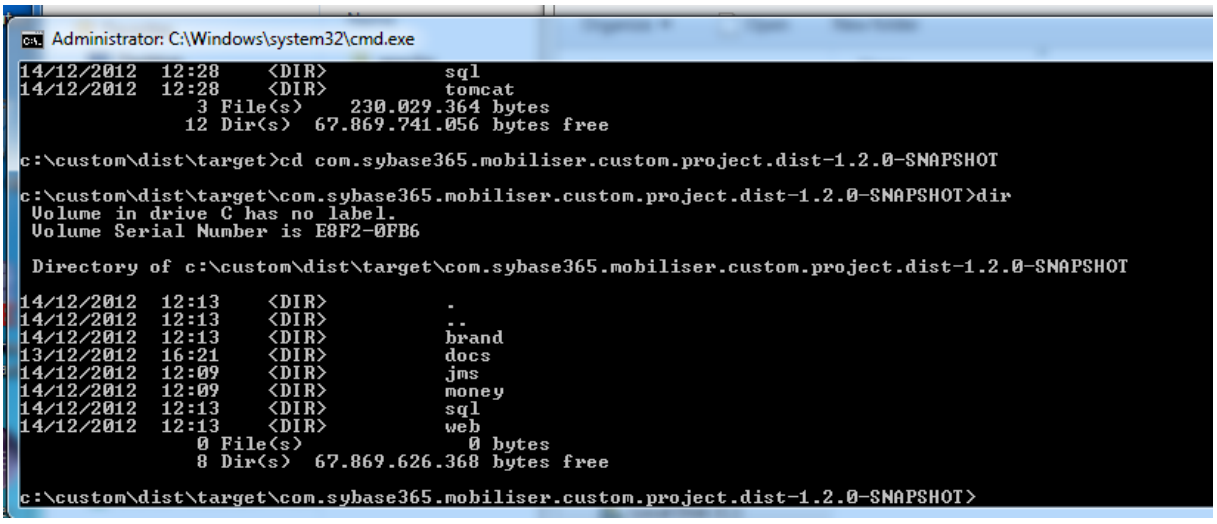


2. We are going to unzip the entire content of the package that Maven prepared for us in the current folder. After unzipping you should find a new folder called

   com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT

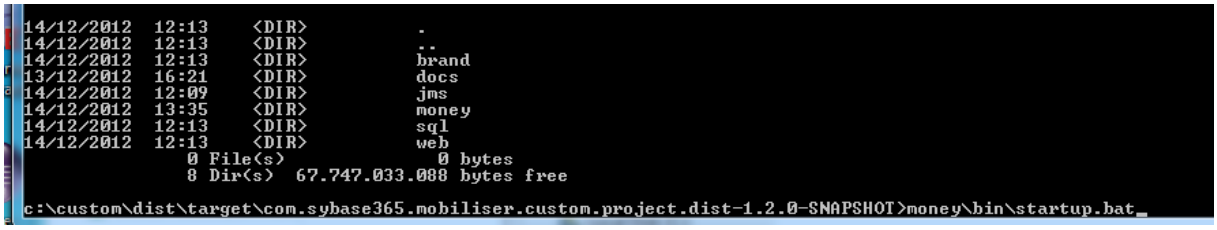


3. Go inside this folder and type **dir**. This is the list of the applications that are part of the Mobiliser package. The main package is “Money Mobiliser”.

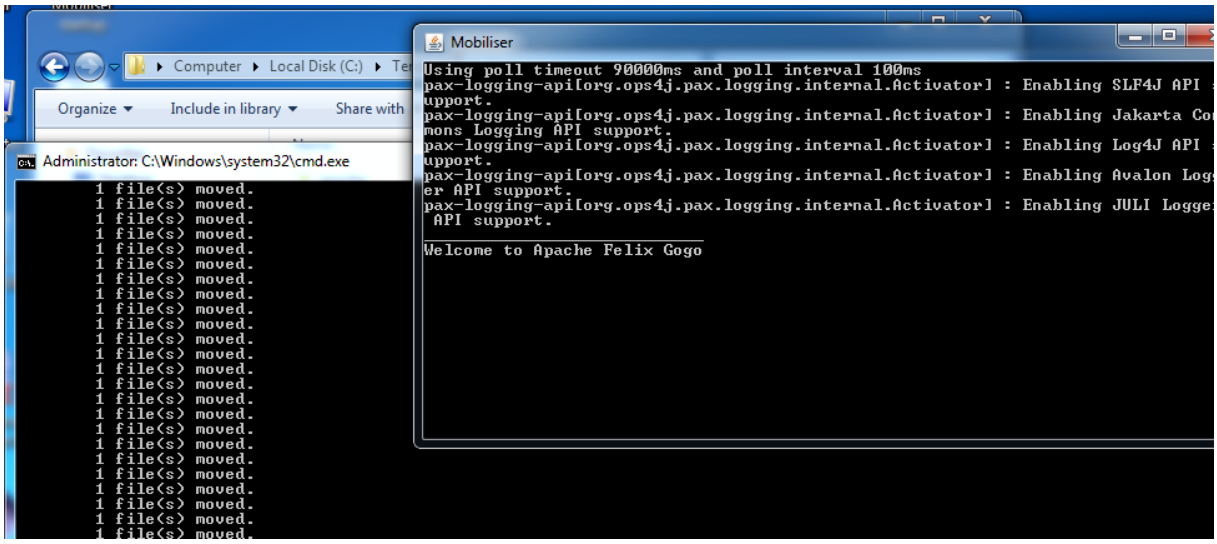


4. We are going to start it first. So run the following command:

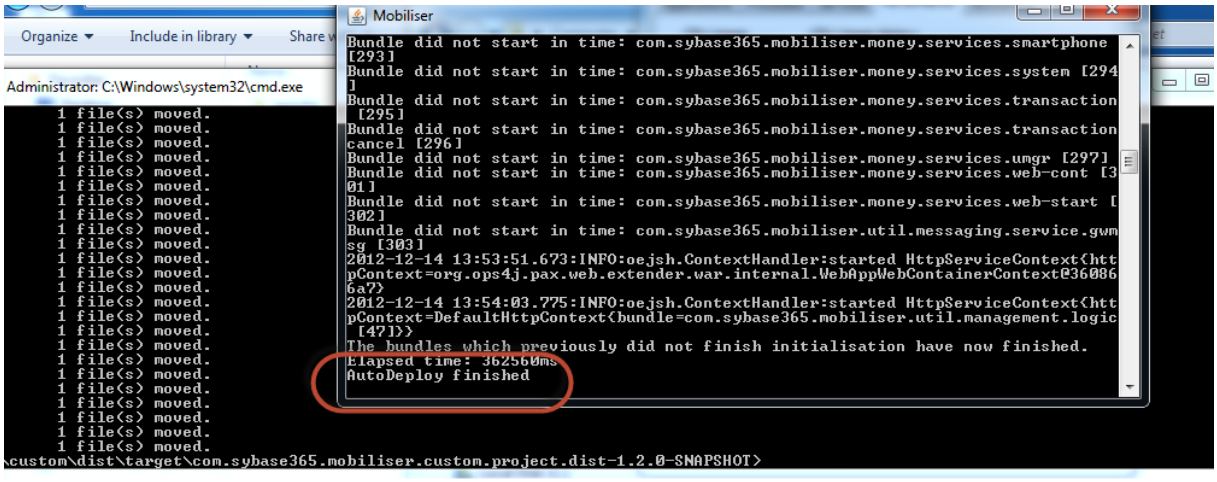
   money\bin\startup.bat



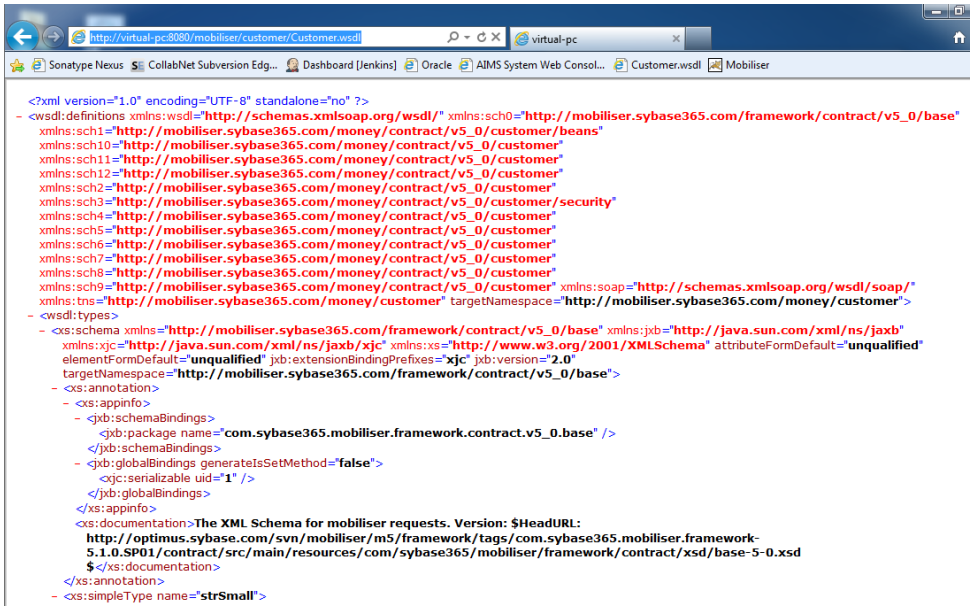
5. A new console window appears. In case you are requested to allow access to the Java Platform through the firewall, click on **Allow access**.



6. A lot of messages will be displayed in the new console window. As soon as you will see the message “AutoDeploy finished” in the new console window, you will understand that Mobiliser is now started and you can play with it.



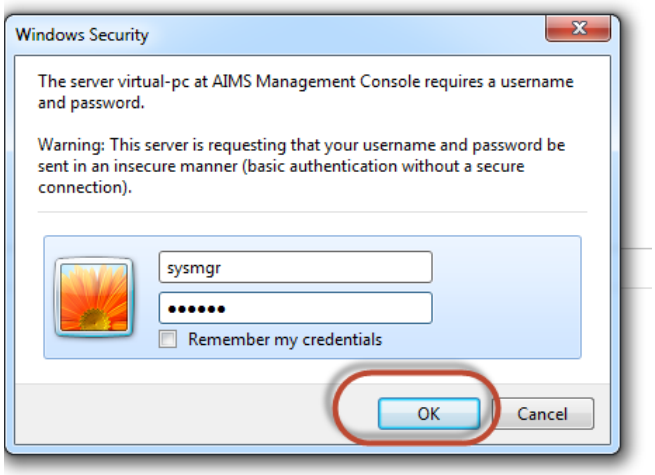
7. Another valid check to do is to open the internet browser and run the following URL:
   http://<server_name>:8080/mobiliser/customer/Customer.wsdl
   (in my case it is http://virtual-pc:8080/mobiliser/customer/Customer.wsdl)
   If Mobiliser is successfully started, you should get the Customer's schema:



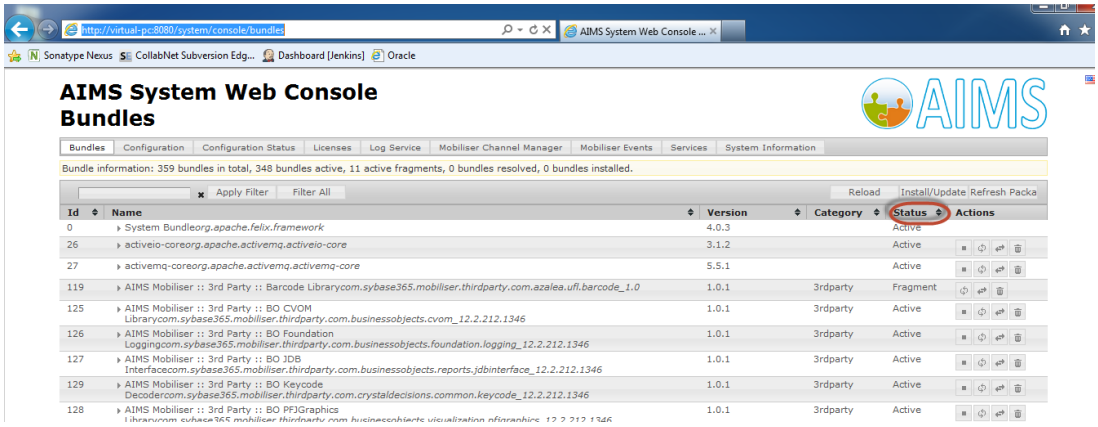
8. Now that Mobiliser is up, from the internet browser go to the following link:
   http://<server_name>:8080/system/console
   (in my case it is http://virtual-pc:8080/system/console)

9. Enter the following credentials:

USER	PASSWORD
sysmgr	secret



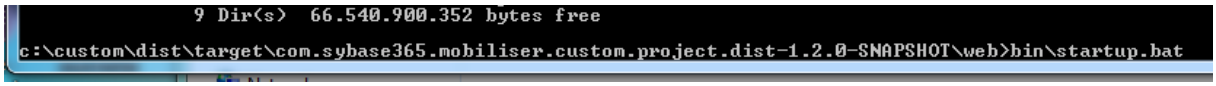
10. The AIMS System Web Console will be opened on the Bundles page. Click on the header of the Status column on the right, in order to sort all the bundles by status.



11. Go down in the list. As soon as you don't see any bundle in the **Installed** status, it means the Money Mobiliser is fully loaded.

12. In order to reach the main page of Mobiliser portal, you need to start the web interface. It's located under the folder "**web**". You can start it by going into this folder and running the command:

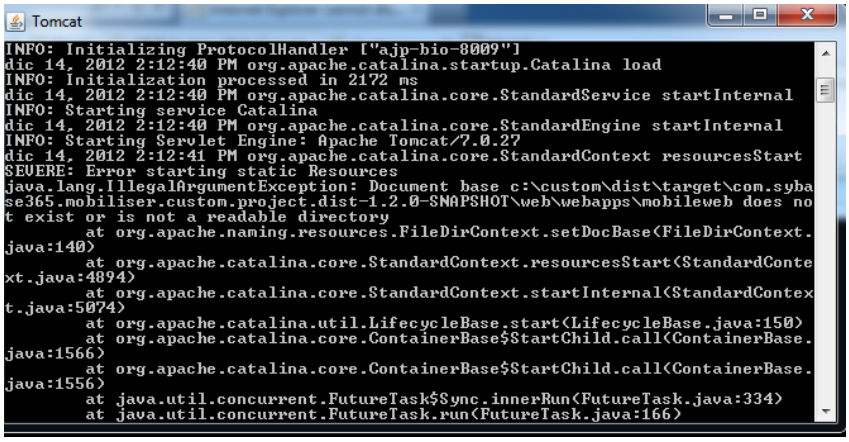
bin\startup.bat



Note

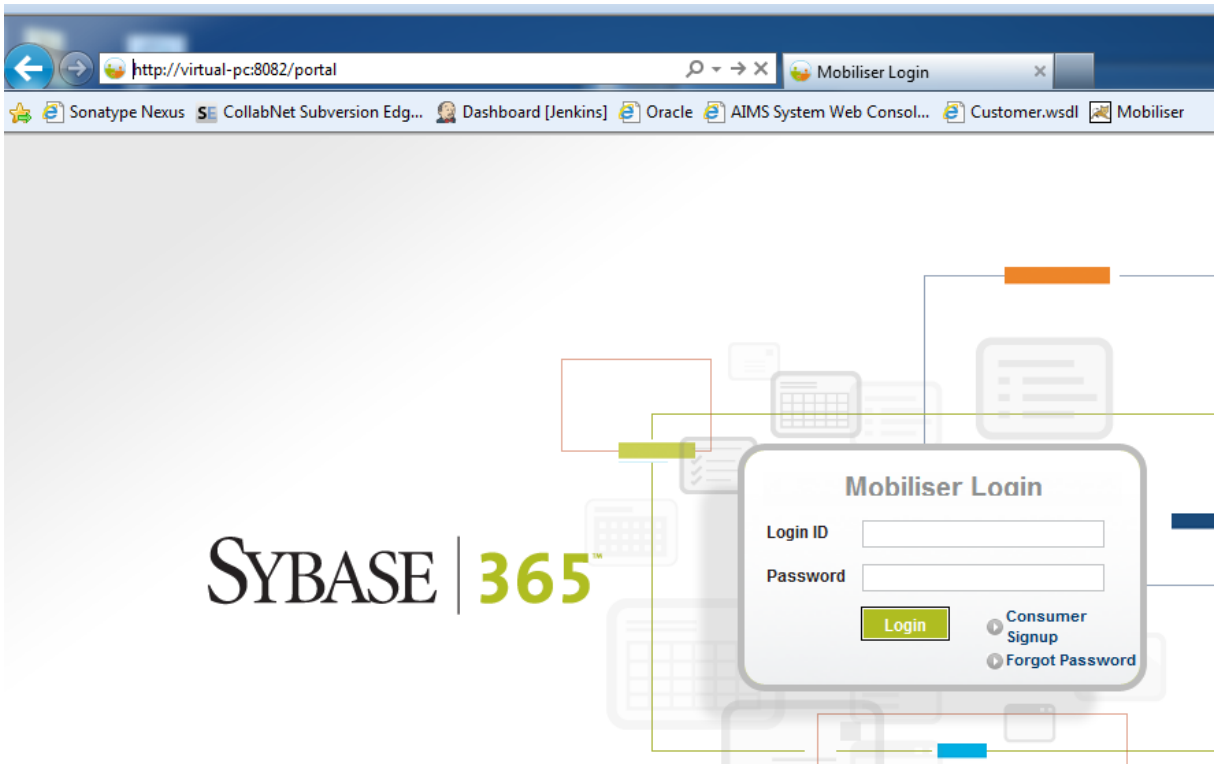
As you can see here we are inside the folder "web" when running the above command. It's important that you run this command by accessing first that folder, otherwise you may get some errors related to the CATALINA path and the Mobiliser portal won't start.

13. You may also encounter the following error when running it. Don't care about it; it's just because there is a missing folder named "mobileweb", but this won't affect the Mobiliser portal launch:

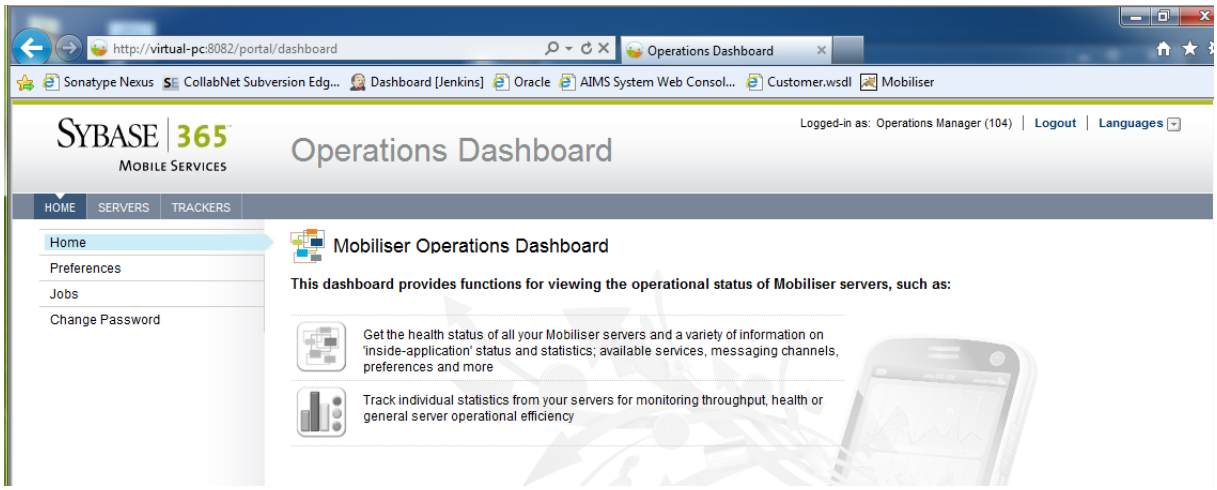


14. From the internet browser type http://<server_name>:8082/portal
(in my case it is http://virtual-pc:8082/portal) and enter the following credentials:

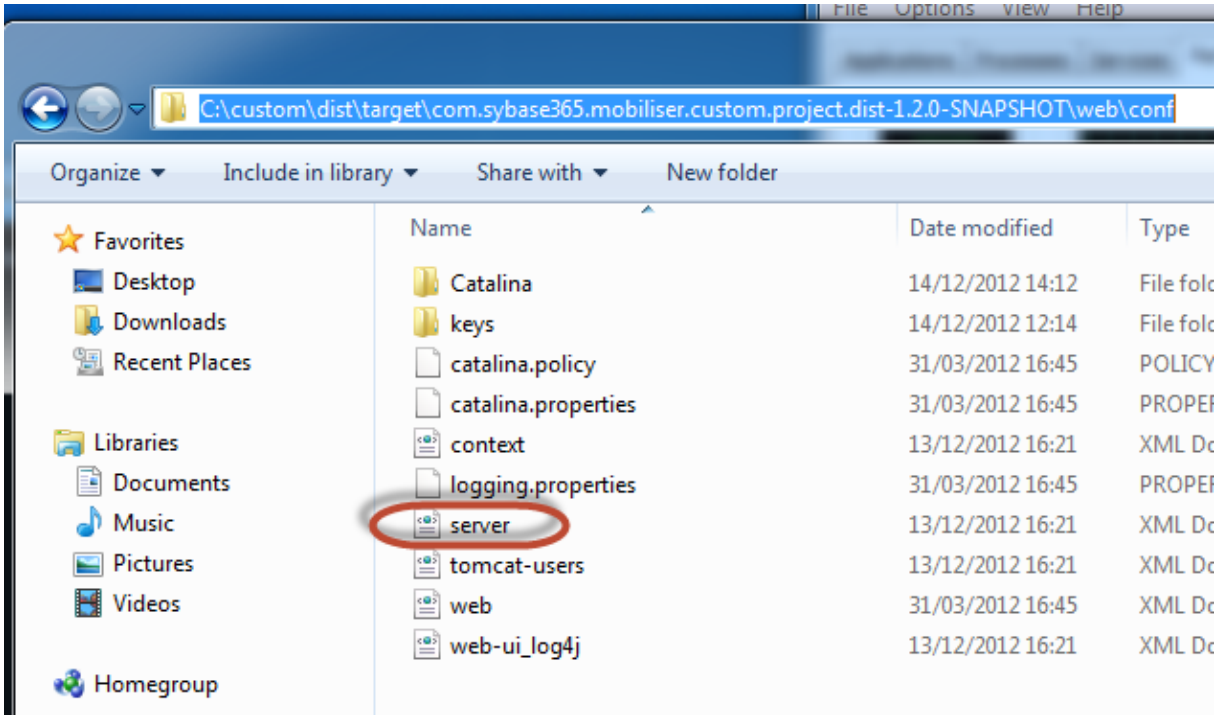
USER	PASSWORD
opsmgr	secret



15. You are now inside the Mobiliser portal:



16. As you can see the Mobiliser portal runs on the port 8082. Should you need to change this port, you can go in the folder **…\web\conf** and edit the file **server.xml**.



17. You can change the port number, save the file and restart the Mobiliser portal with the new settings.

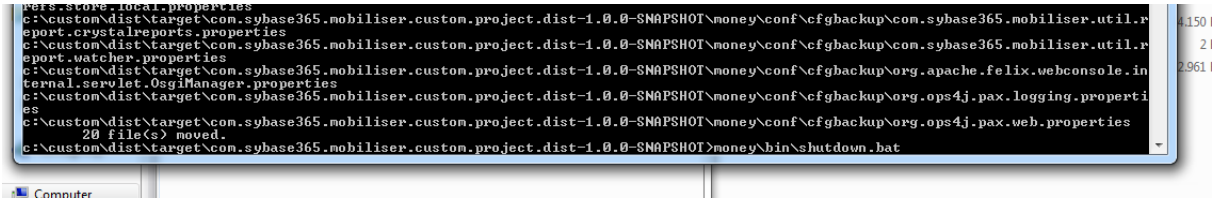
18. When you want to stop the Mobiliser portal, simply run the command

    bin\shutdown.bat

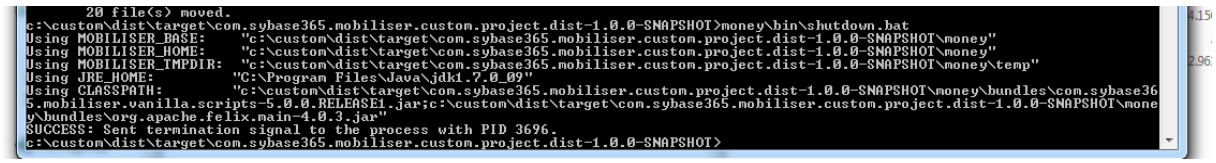
    from the “web” folder.

19. If you want to stop Money Mobiliser, you can run the command:

    money\bin\shutdown.bat



20. Money Mobiliser has successfully stopped.



4.9 Install Eclipse

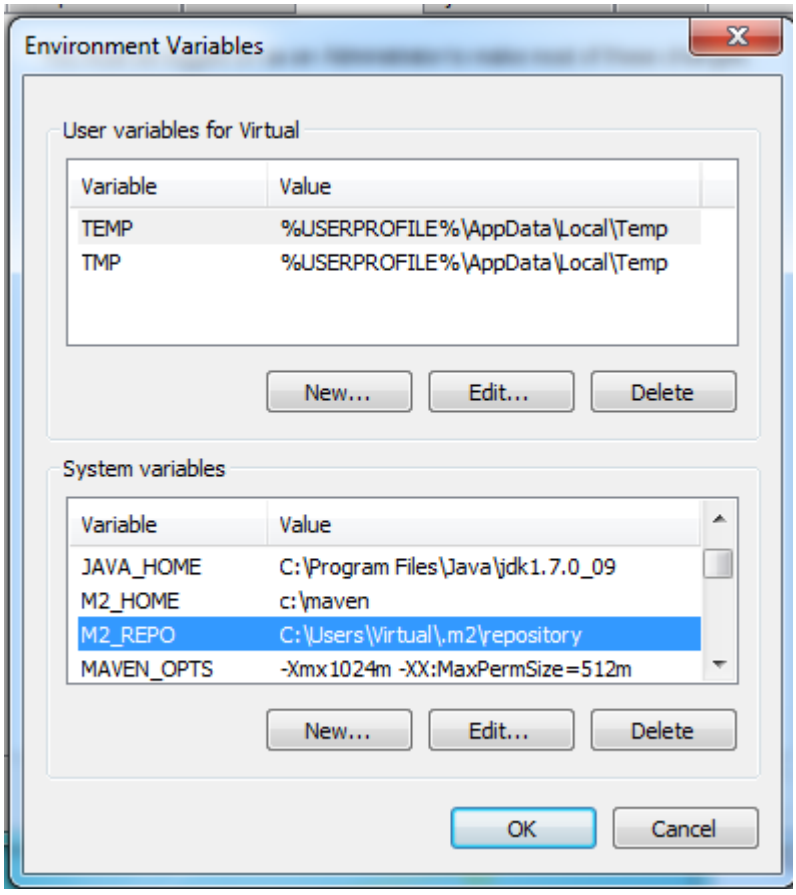
1. The last step in this guide is to install the Eclipse development environment. Eclipse can be downloaded from the following web site: http://www.eclipse.org/downloads/

2. There are several distributions. The most famous one are Juno and Indigo. In this guide, we are using Juno, so please download the Eclipse IDE for Java EE Developers for your operating system.

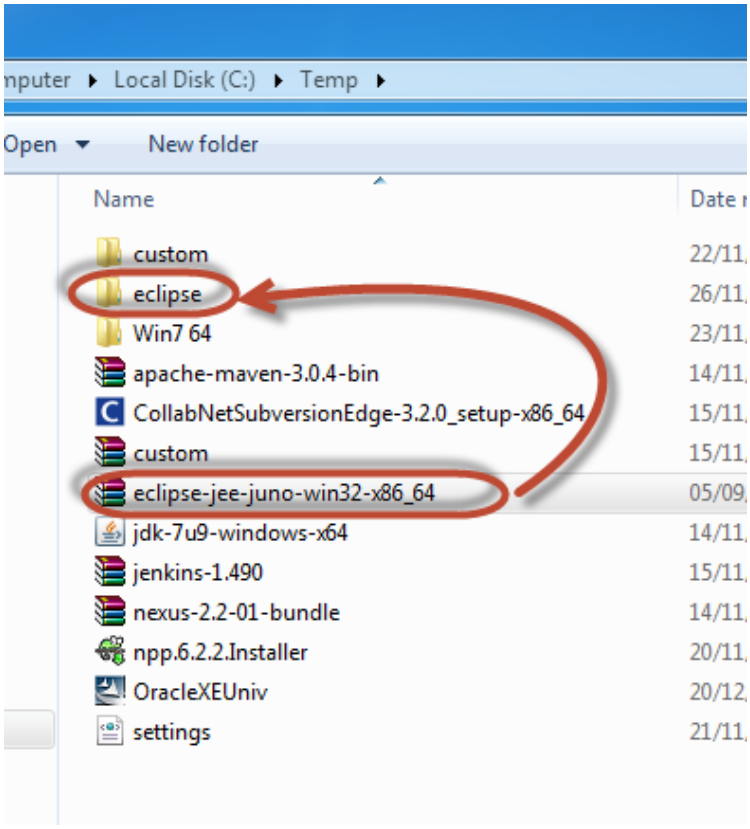
3. Once downloaded, before we start the installation of Eclipse we need to configure another Environment variable. This variable is named M2_REPO: it will tell Eclipse where to find the Maven repository. So open the **Advanced System Settings** and click on the **Environment Variables** button. Create the following two variables:

M2_REPO=<local_user_path>\.m2\repository

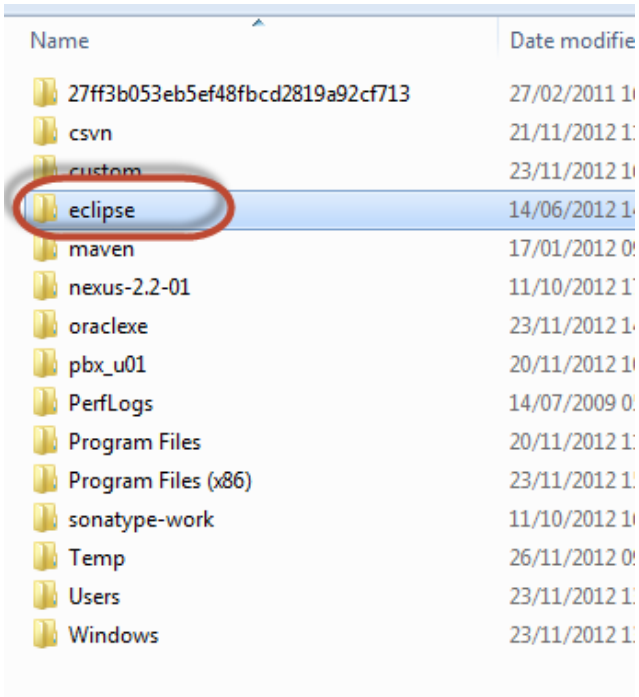
(in my case it is C:\Users\Virtual\.m2\repository)



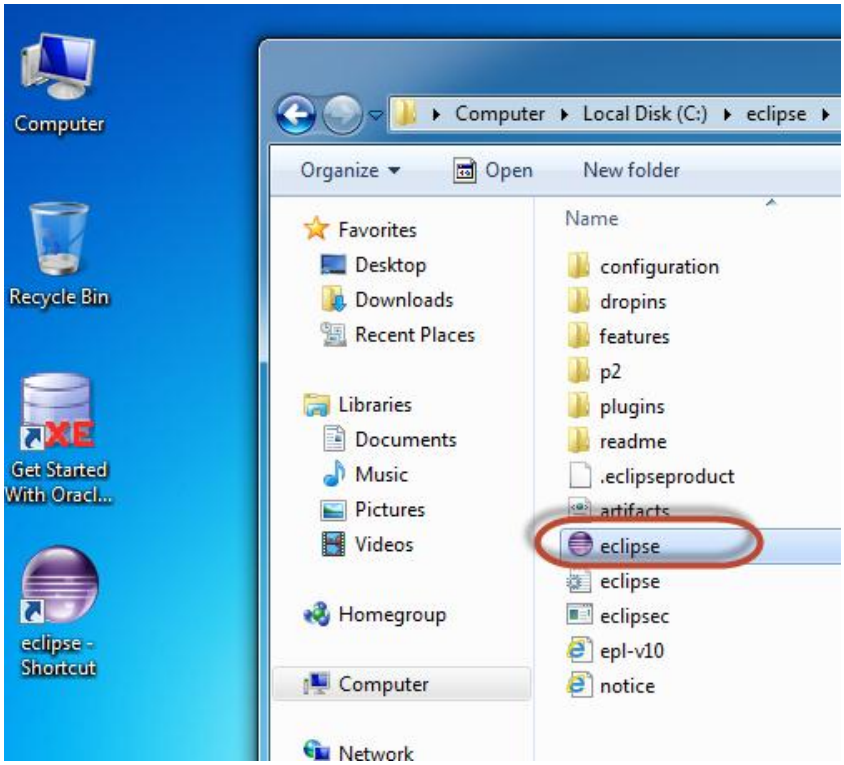
4. After this variable has been set, we can continue with the Eclipse installation. Extract the content of the downloaded package in a folder.



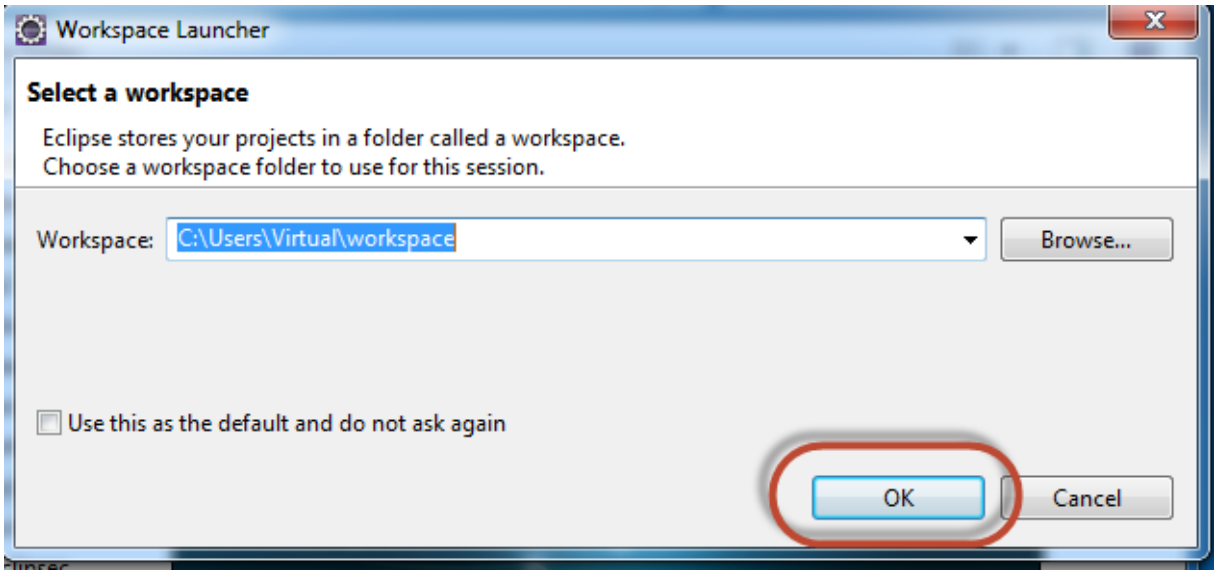
5. Move the Eclipse folder you have just extracted in a convenient location on your drive. For example, we are moving it in the root of the C drive.



6. Go inside the Eclipse folder and run the Eclipse executable. You can also create a shortcut on the desktop for your comfort.



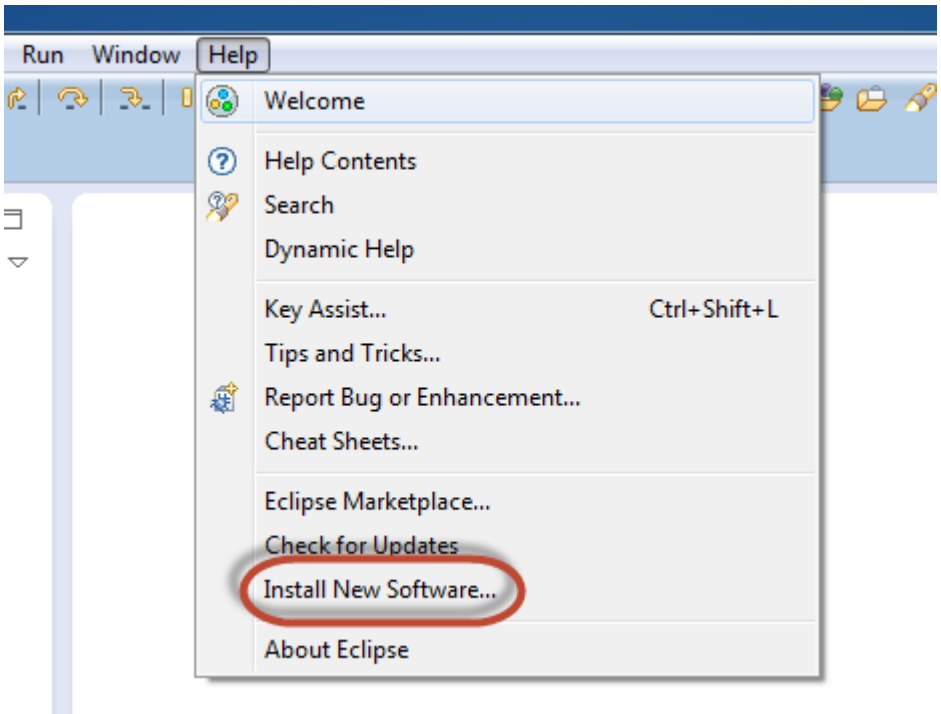
7. Create a new workspace. You can use your user folder, as in this case.



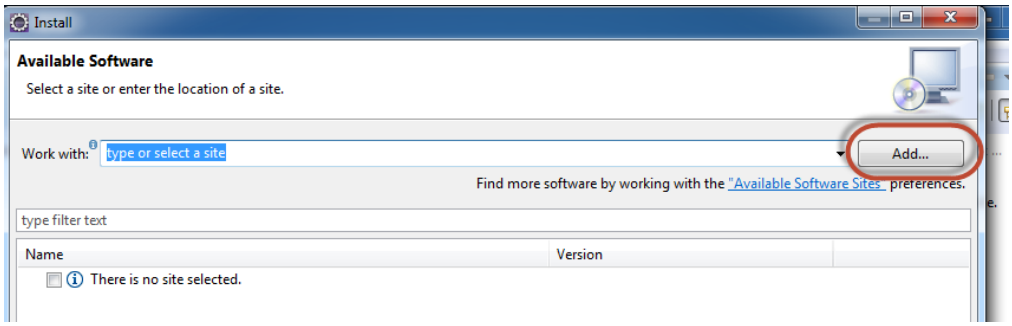
8. Once it is loaded, you can close the Welcome screen and go directly on the workbench.

9. Before we can work with it, we need to download some special plugins in order to be able to download the project from Subversion and to build it with Maven. For the first thing we have to download the **Sublcipse** plugin and for the second the **M2E** plugin.

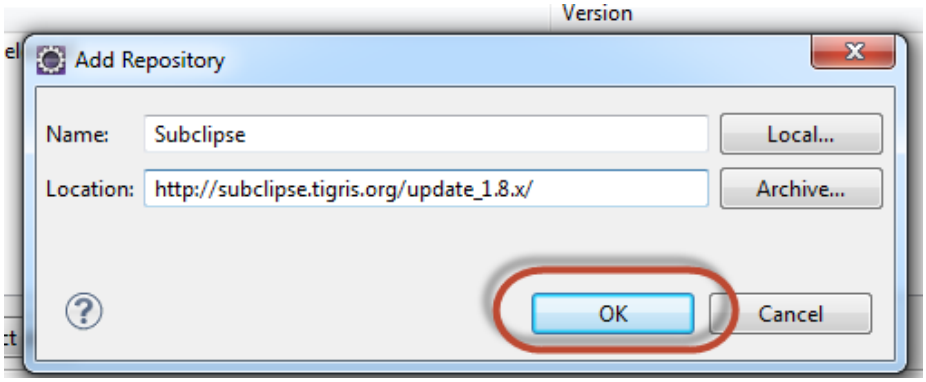
10. Go on the menu **Help → Install New Software**



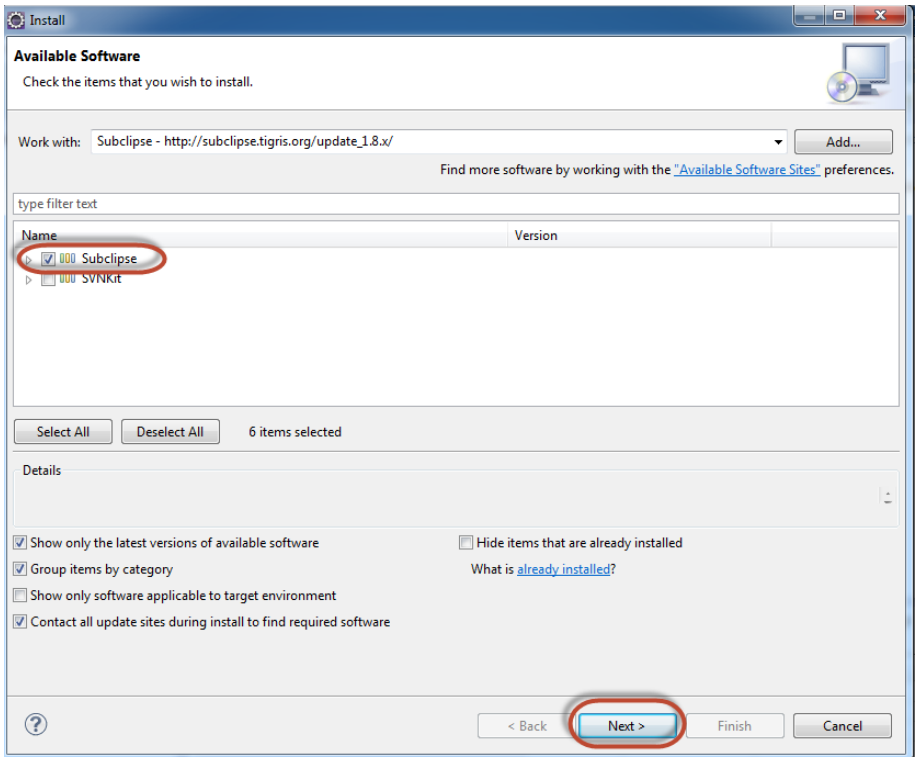
11. Click on **Add**.



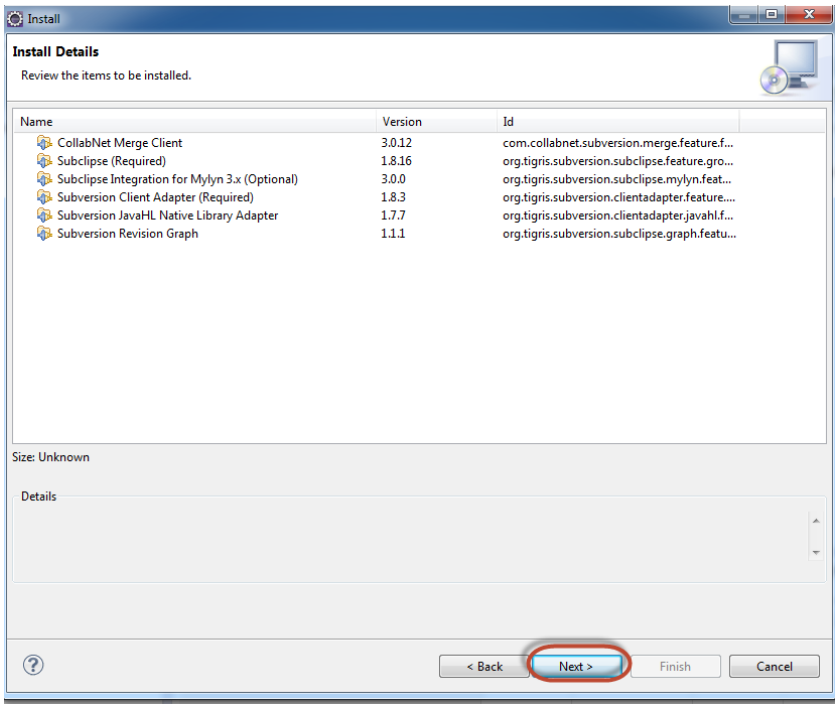
12. Enter a name (i.e. Subclipse) and the link http://subclipse.tigris.org/update_1.8.x/ from which our plugin will be downloaded.



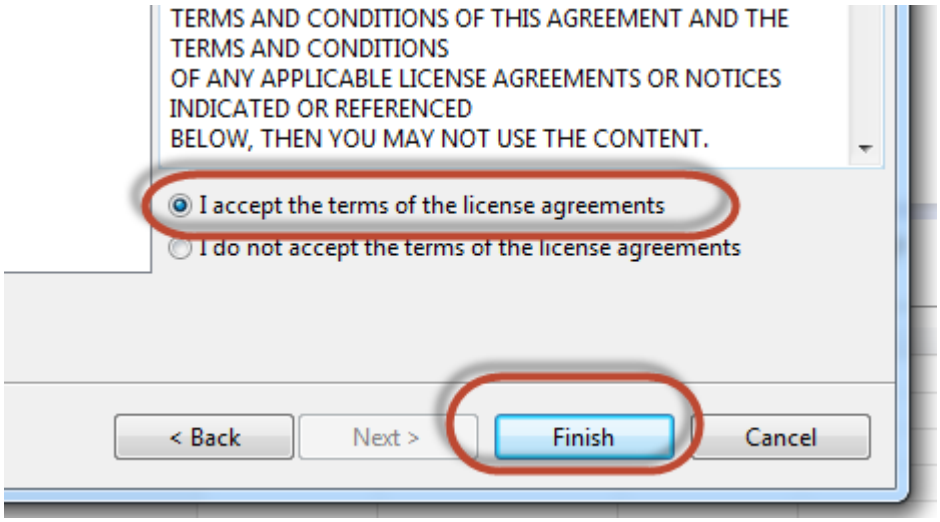
13. Select just the **Subclipse** item and click on **Next**.



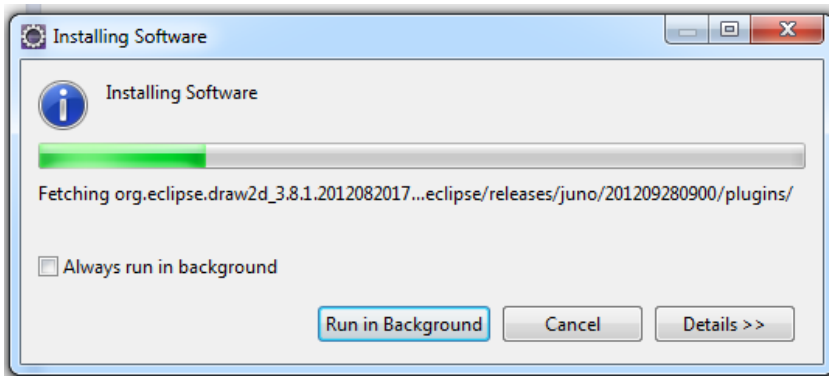
14. Click on **Next**.



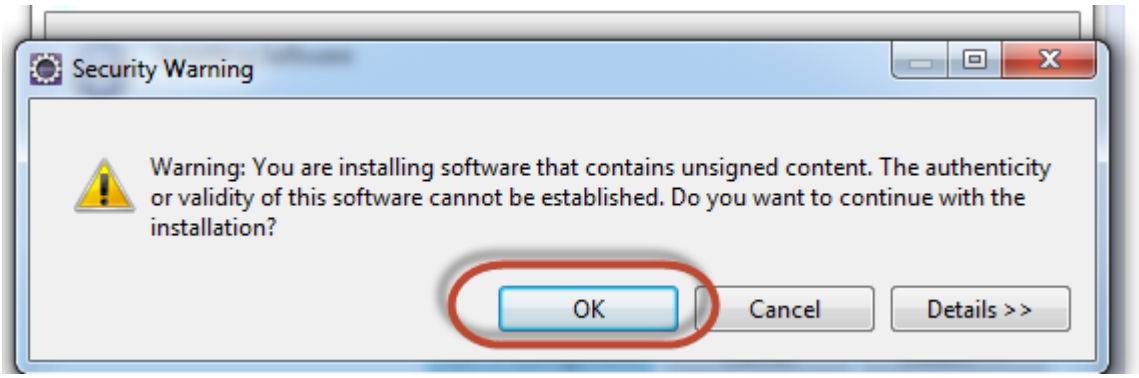
15. Accept the license agreement and click on **Finish**.



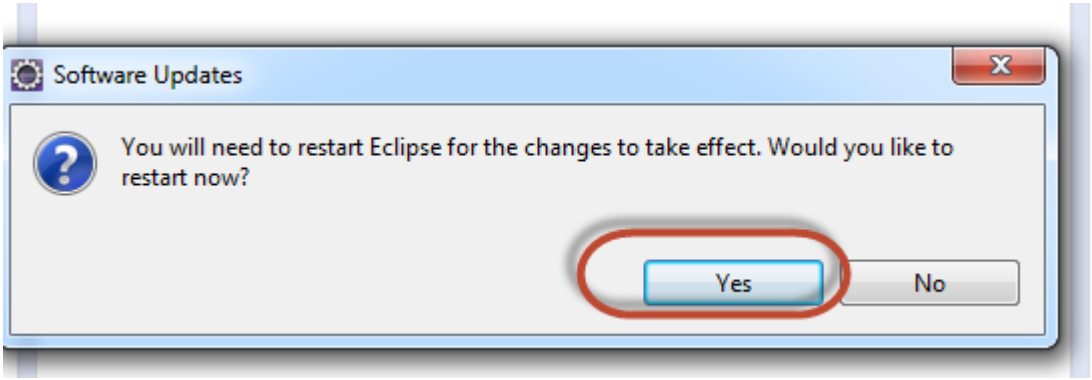
16. The plugin is being installed.



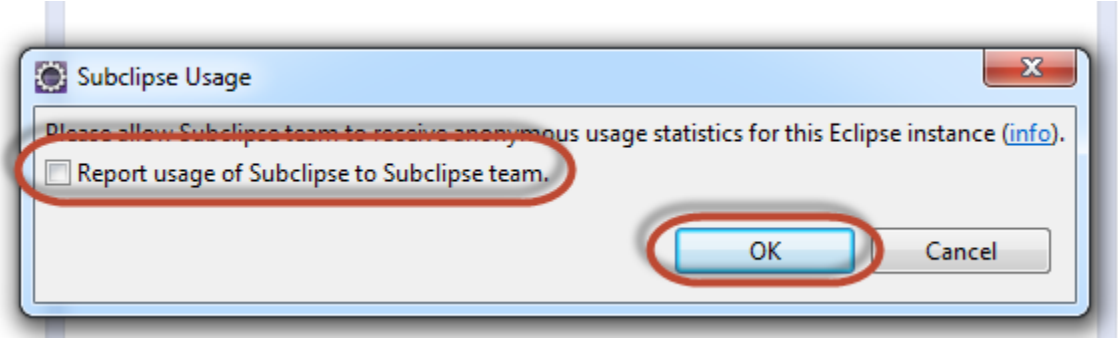
17. Click on **OK**.



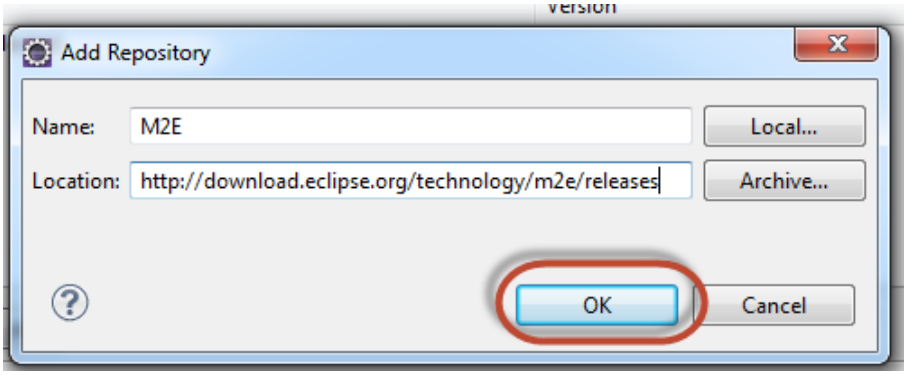
18. Restart Eclipse



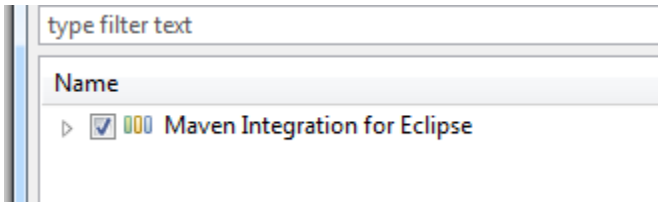
19. After restart, you can also uncheck this box and click on **OK**.



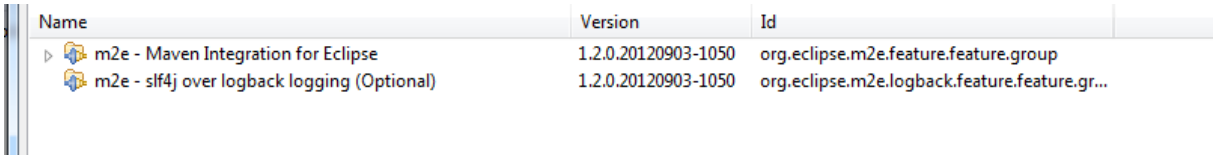
20. Now let's install the last plugin: M2E. It's Maven for Eclipse. Enter a name and the URL http://download.eclipse.org/technology/m2e/releases and click on **OK**.



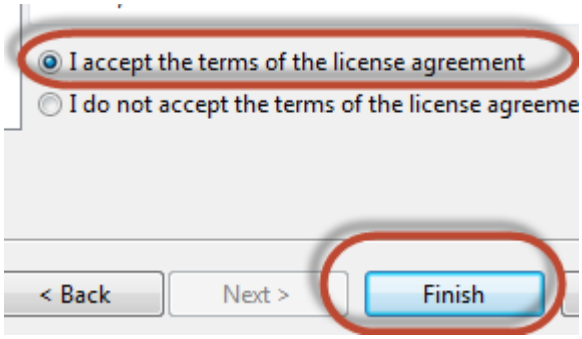
21. Select **Maven Integration for Eclipse** and click on **Next**.



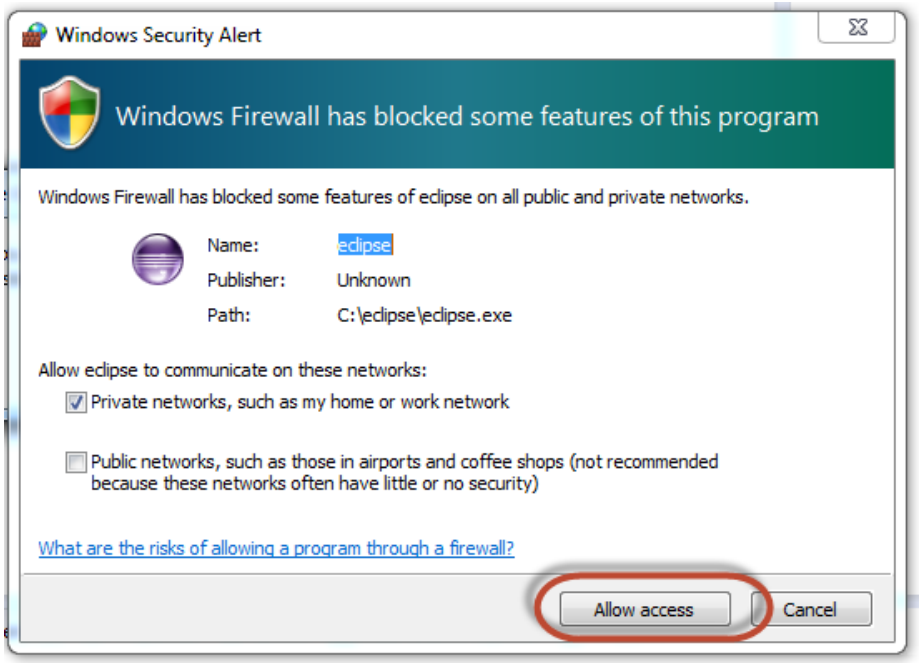
22. Click again on **Next** here.



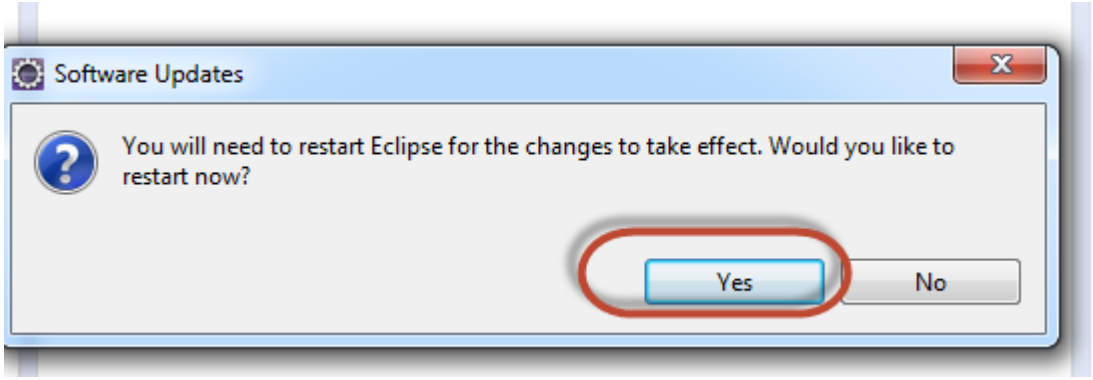
23. Accept the license agreement and click on **Finish**.



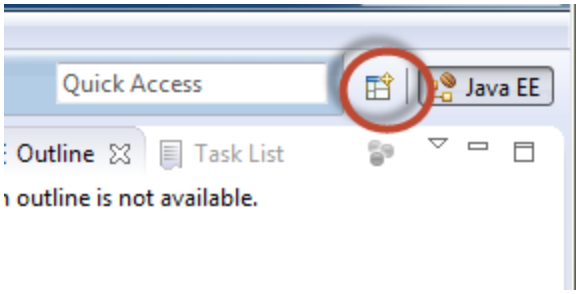
24. In case you get this message, click on **Allow access**.



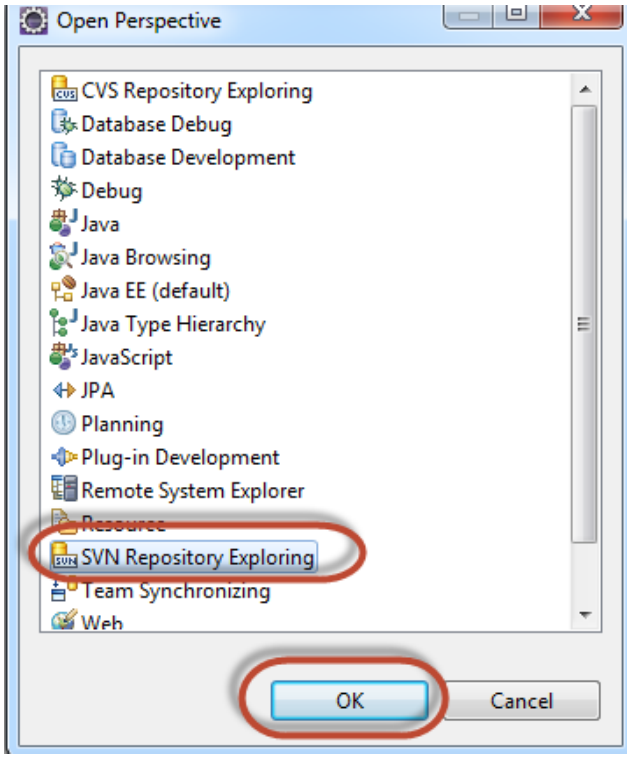
25. Restart Eclipse.



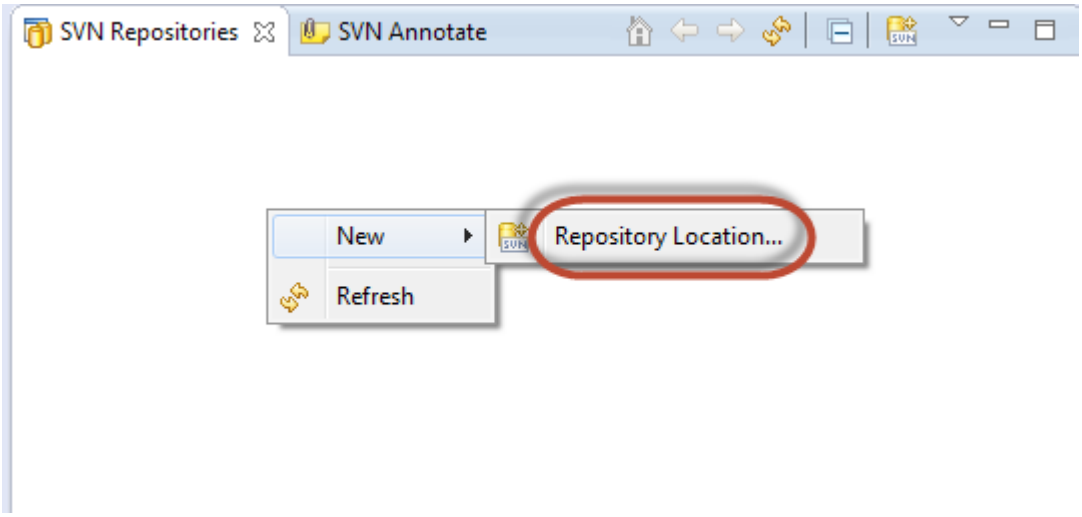
26. Once again in Eclipse, click on the **Open perspective button** on the top right corner.



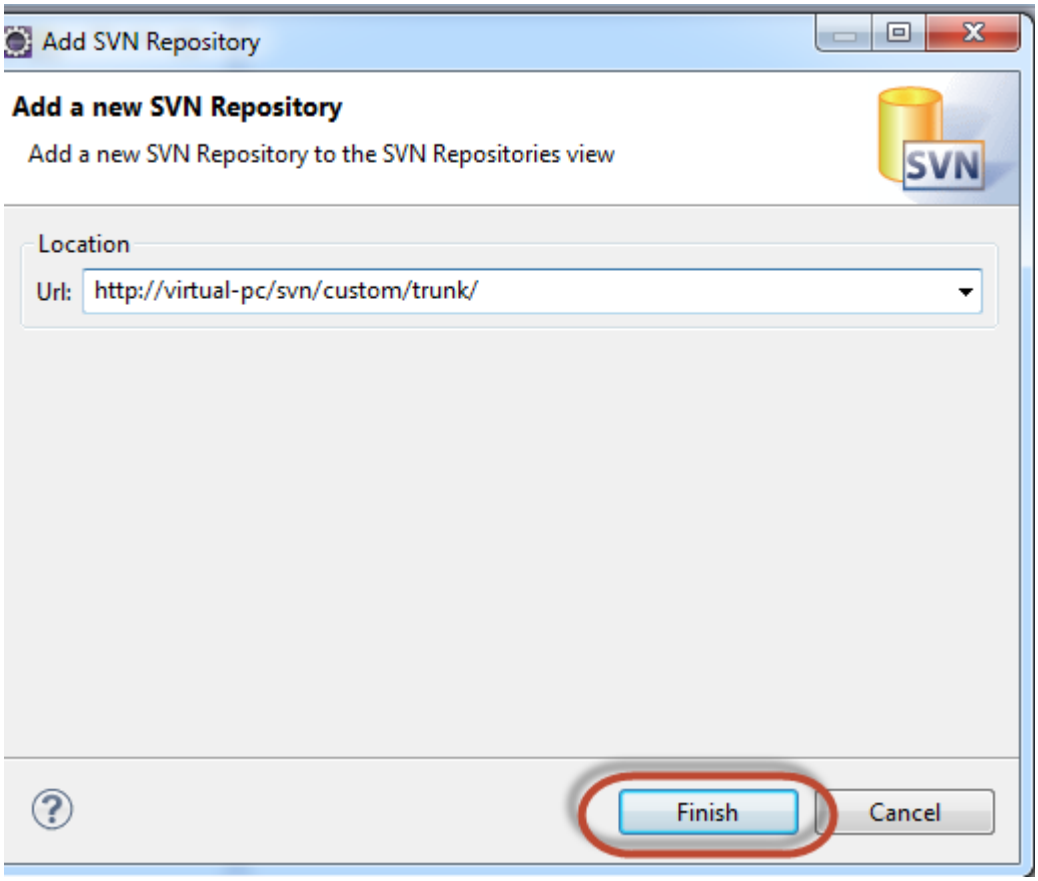
27. Select the **SVN Repository Exploring** perspective and click on **OK.**



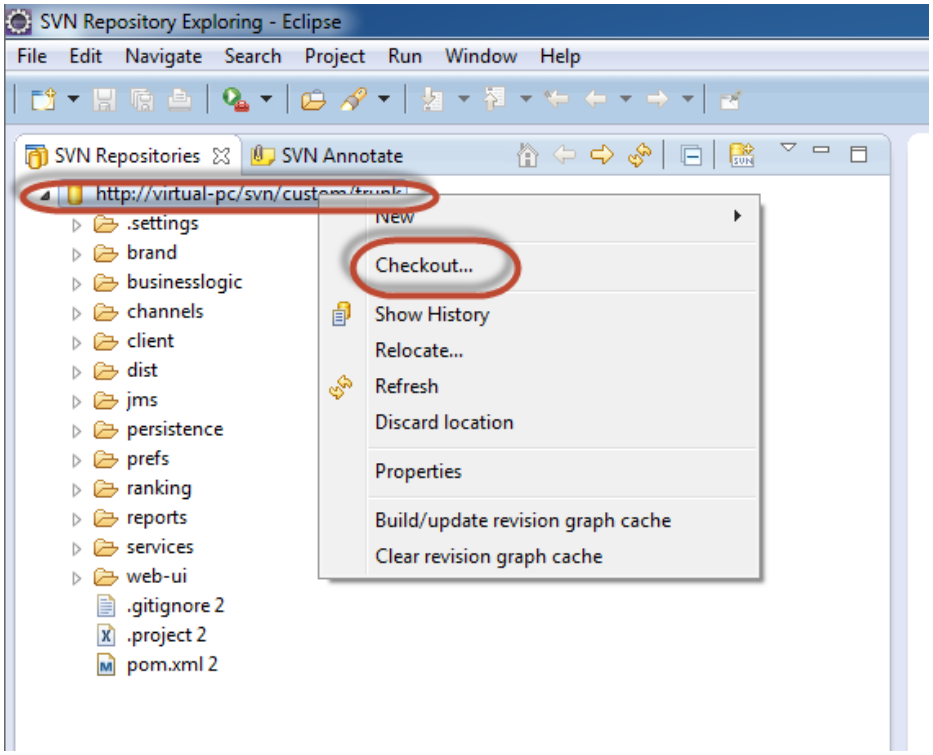
28. A new perspective will be opened. Right click on the blank space in the **SVN Repositories** window and choose **Repository Location**.



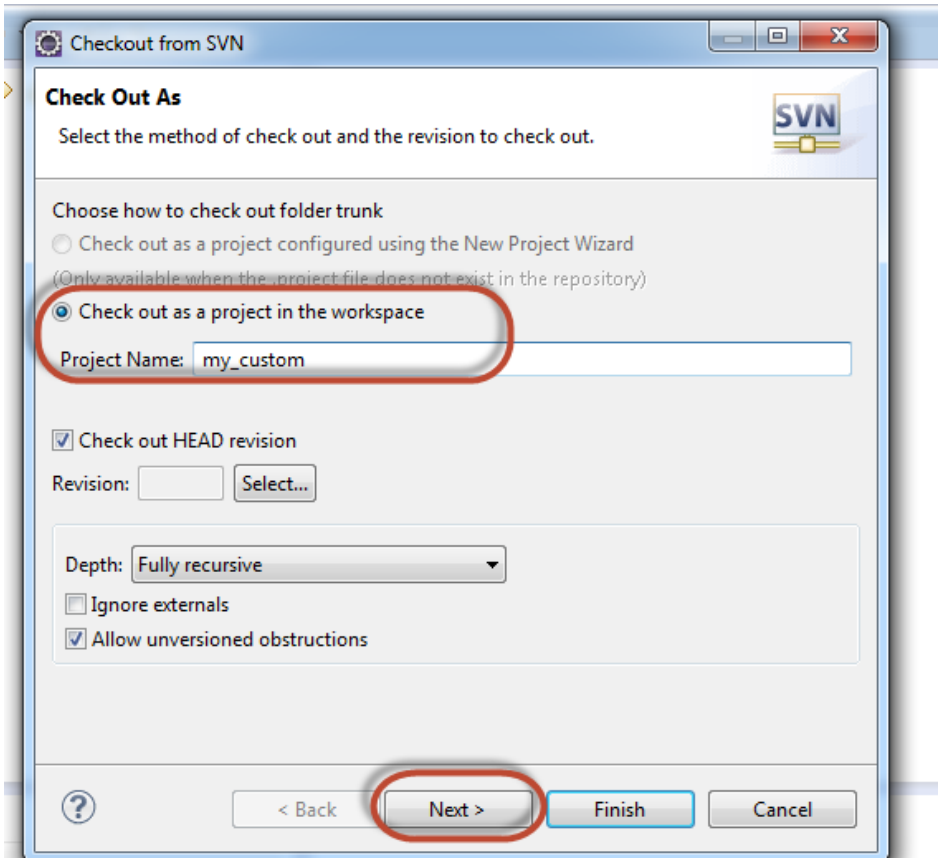
29. Enter the URL of your repository in Subversion: http://<server_name>/svn/custom/trunk/ (in my case it is http://virtual-pc/svn/custom/trunk/)



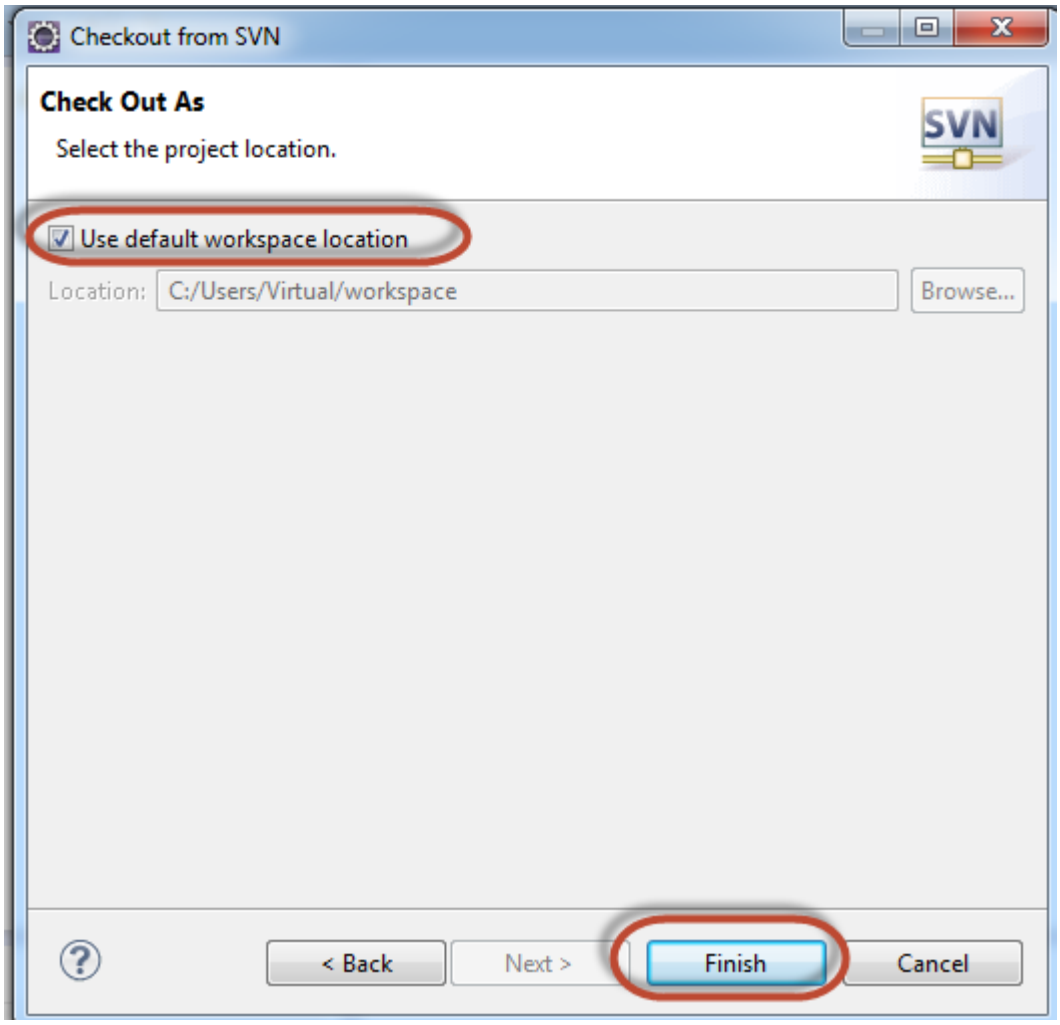
30. Right click on the repository name and select **Checkout**. The project will be downloaded in the workspace folder locally.



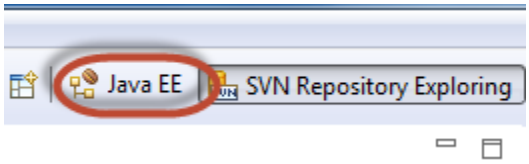
31. Select **Check out as a project in the workspace** and enter a valid name. Then click on **Next**.



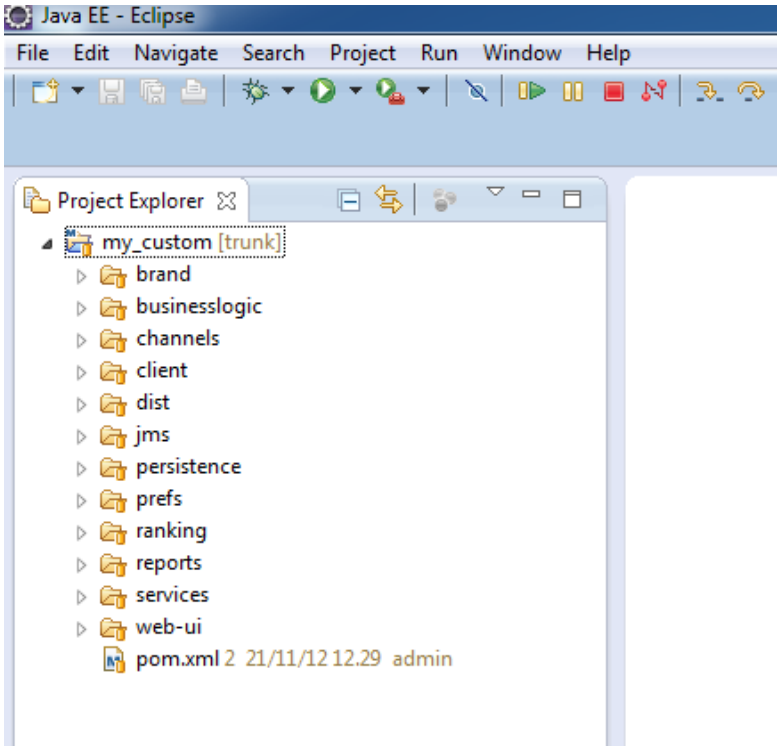
32. Use the default workspace location and click on **Finish**.



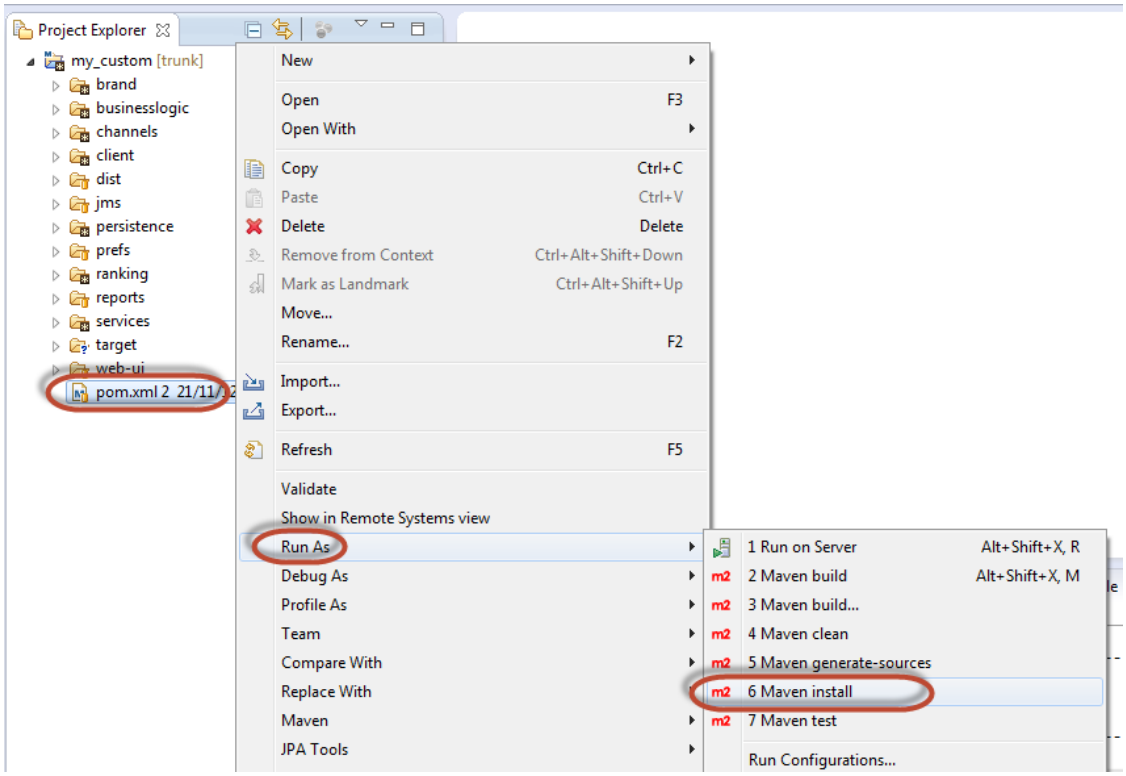
33. Switch back to the **Java EE** perspective.



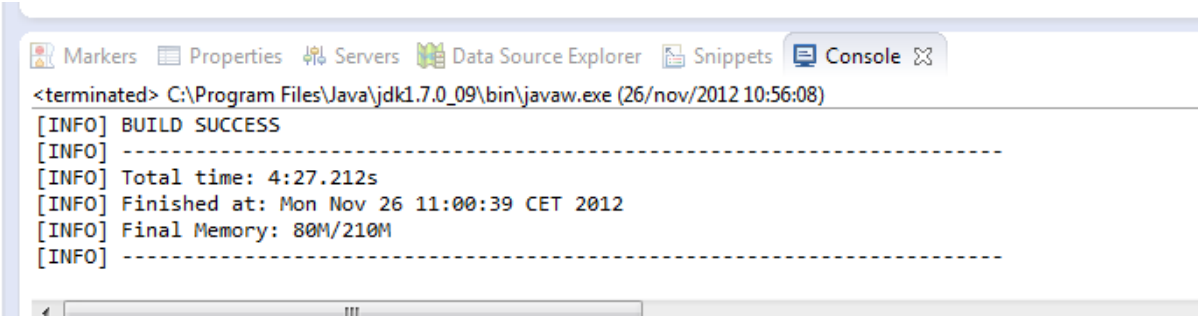
34. Your project is now in your workspace.



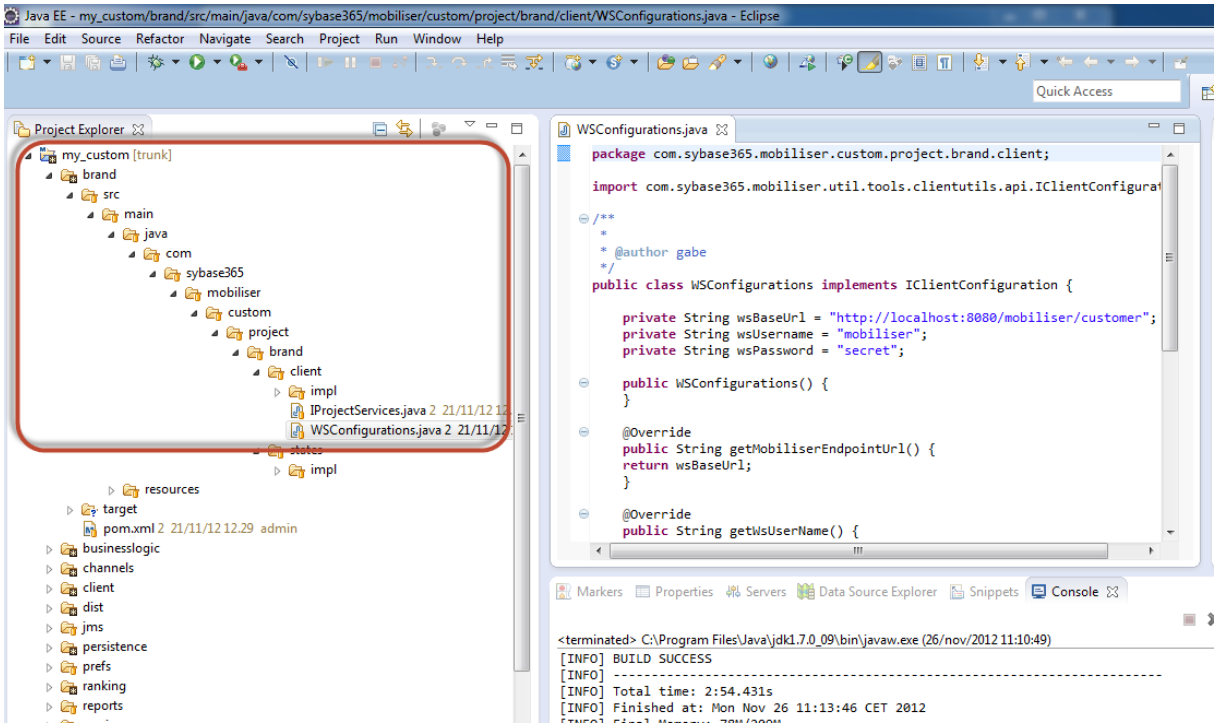
35. You can now build it as a Maven install by right clicking on the **pom.xml** file and selecting **Run As → Maven install**



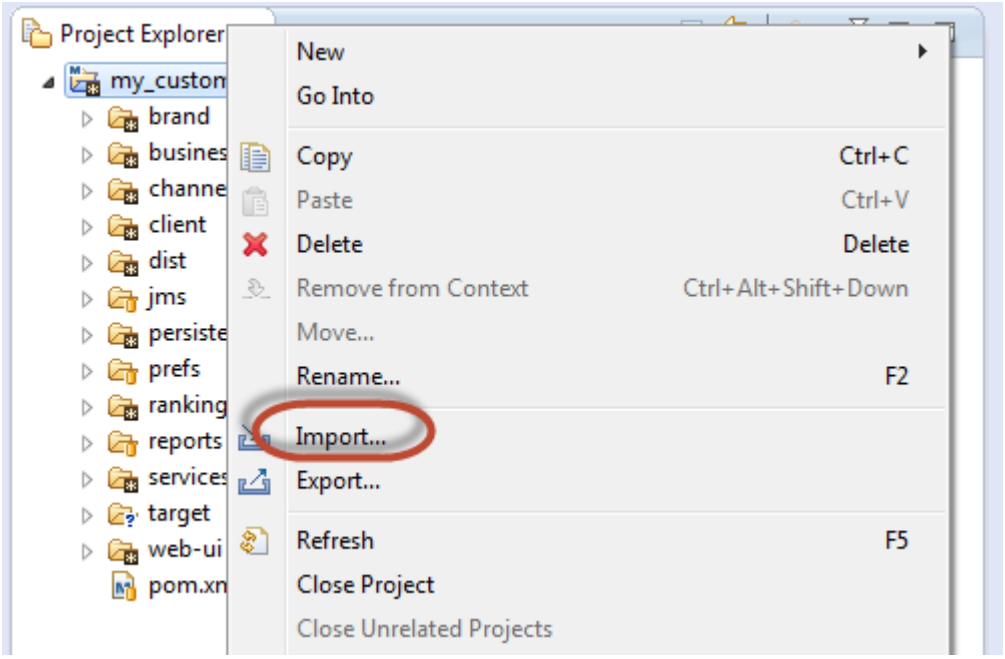
36. The project is being built and, when all is finished, in the console window you should get the message that building was successful.



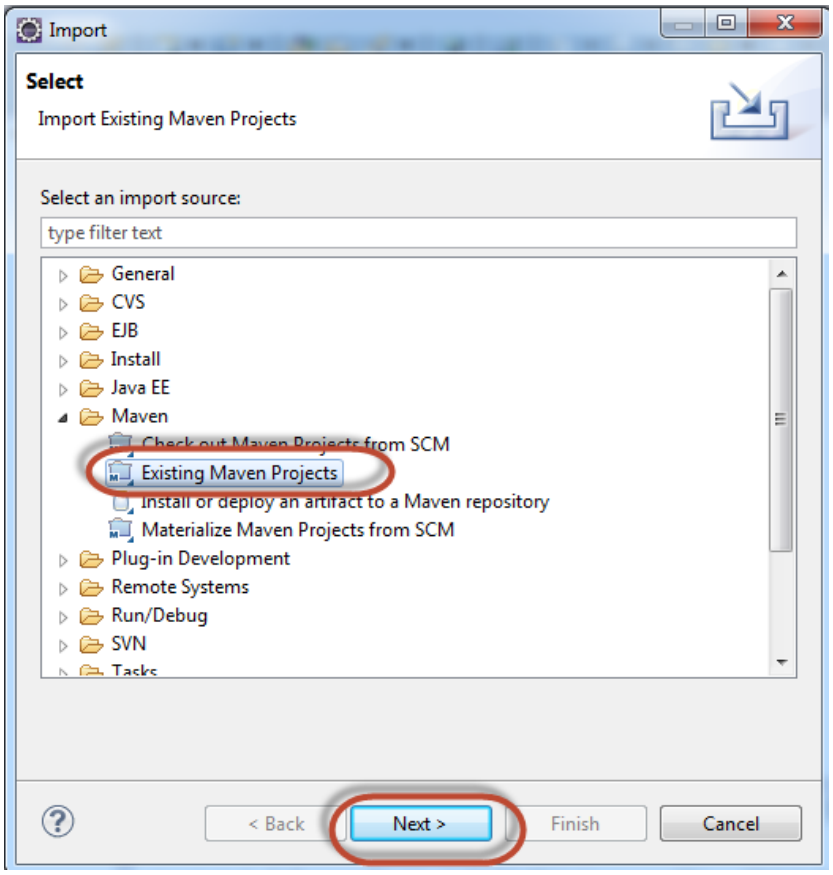
37. If we need to change the source code for example, we can drill down one of the folders until we find the needed source file. As you can see here, the folder structure can be very long.



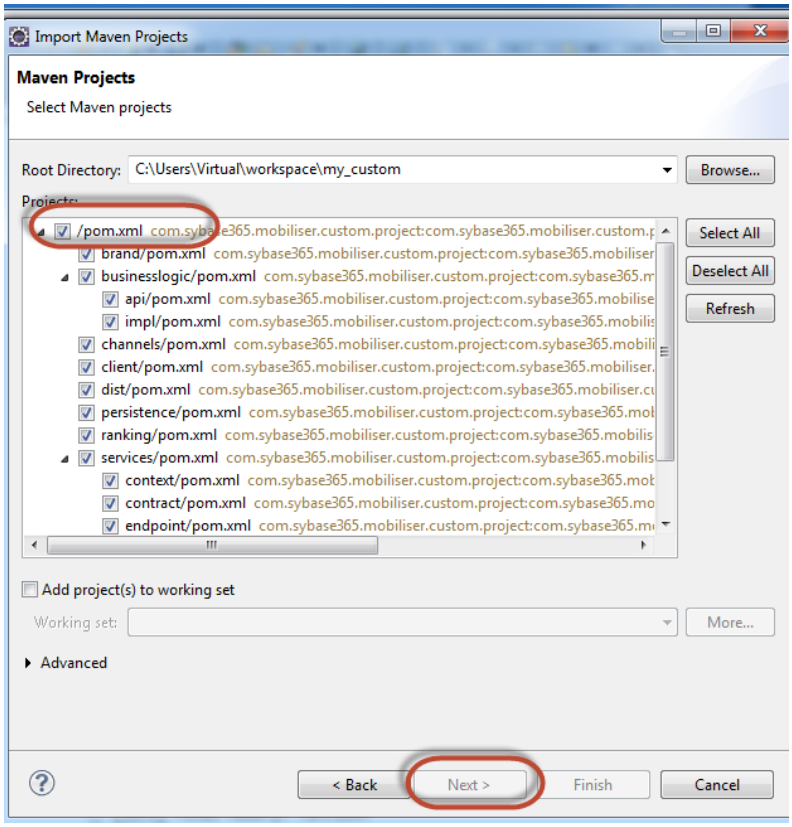
38. If we want to manage the source code in a better way, we could import the project inside Eclipse. Right click on the project name and click on **Import**.



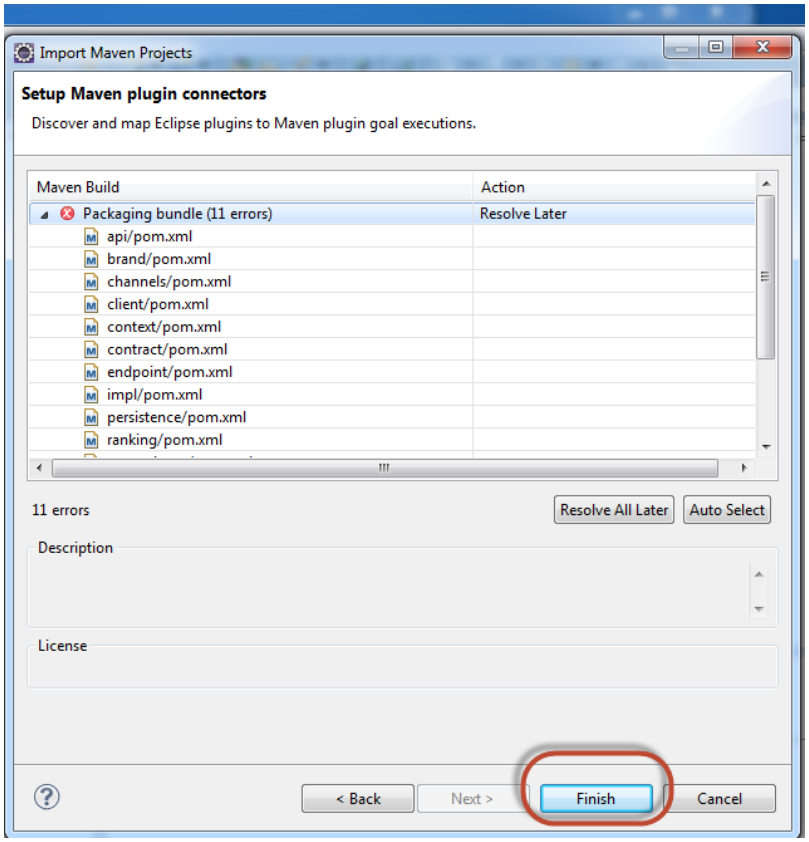
39. Select **Existing Maven Projects** and click on **Next**.



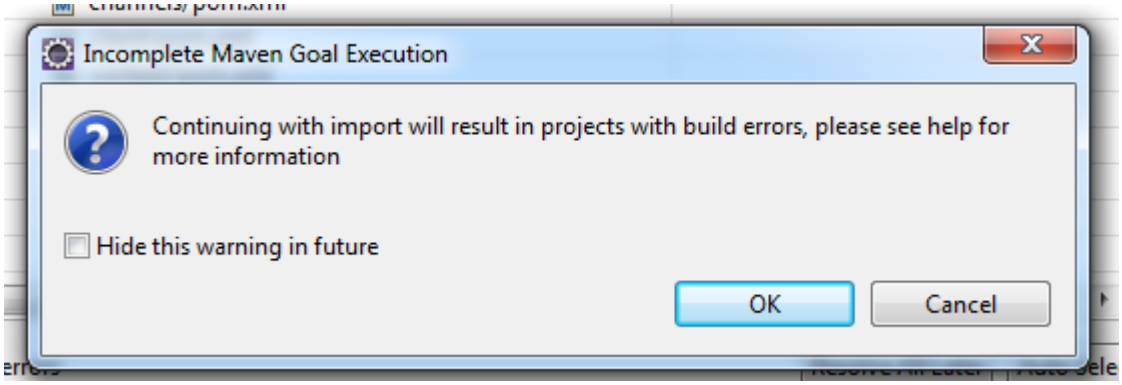
40. The Next button appears as disabled. In order to enable it, unselect and then select again the first checkbox: the button will be enabled. Click on it.



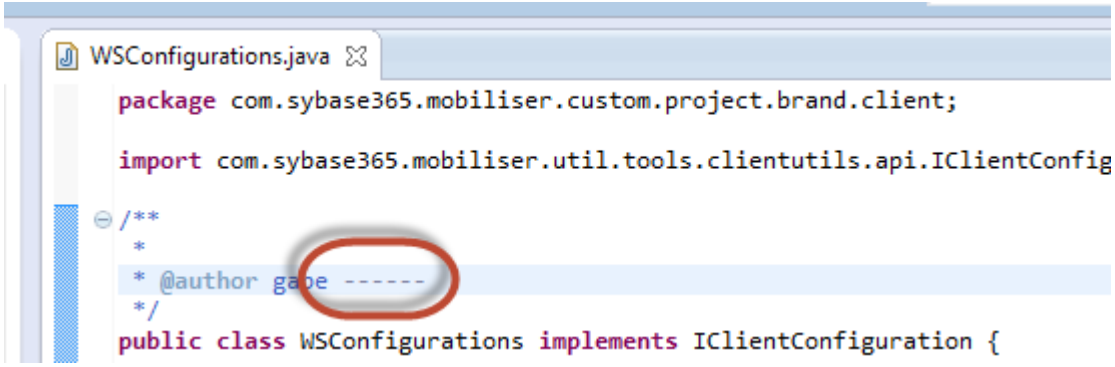
41. The errors can be ignored: click on **Finish** here.



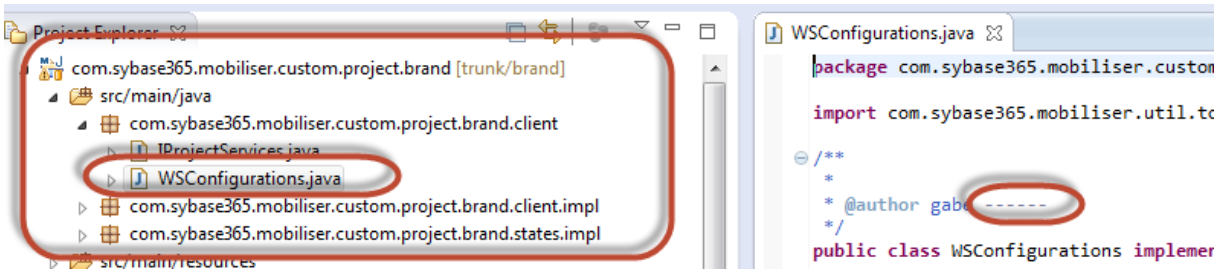
42. Click on **OK**.



43. Now if we change the file at step 37 and save it.



44. It will be much easier to reach it in the following way where the path is less long than before.



45. Your Eclipse is correctly configured.

5. Appendix

Appendix A – Software collection

Here below you can find a list of all the files used during this guide.
Link to the files repository: http://to_be_defined

Filename	Description
apache-maven-3.0.4-bin.zip	Apache Maven
CollabNetSubversionEdge-3.2.0_setup-x86_64.exe	CollabNet Subversion
custom.zip	Mobiliser source code
eclipse-jee-juno-win32-x86_64.zip	Eclipse Juno for Java EE Developers
jdk-7u9-windows-x64.exe	Java 7 update 9
jenkins-1.490.zip	Jenkins
nexus-2.2-01-bundle.zip	Sonatype Nexus
npp.6.2.2.Installer.exe	Notepad++
OracleXEUniv.exe	Oracle Database 10g Express Edition
settings.xml	Settings for Maven

www.sdn.sap.com/irj/sdn/howtoguides