



University of New York Tirana
Faculty of Engineering and Architecture
Rruga e Kavajës, pranë 21 Dhjetorit (Sheshi Ataturk)
Tirane, Shqipëri

Master of Science in Computer Science

**Distributed Systems
Manual for Laboratory Practice**

Enterprise JavaBeans

**PART I
Environment Configuration and
Execution of Examples
A Simple Banking Application with EJBs**

Prof. Dr. Marenglen Biba
Department of Computer Science,
E-mail : marenglenbiba@unyt.edu.al

1. Document Purpose

This document contains explanations on how to run the following programs: RMI Servers, RMI Client and Database Server.

For running the programs, a correct configuration of the running environment is necessary (path and classpath variables).

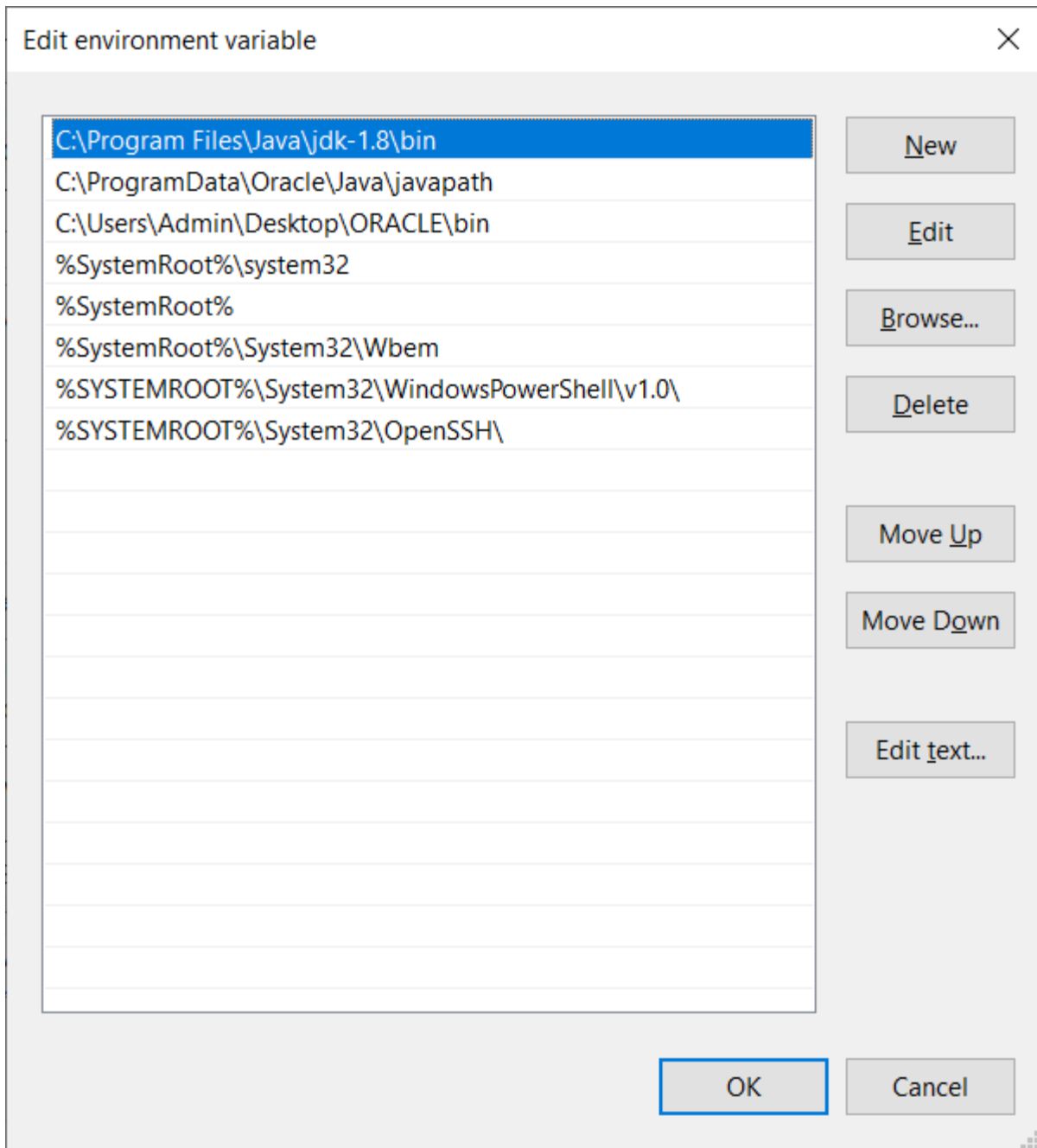
- Install Java SE (JDK) JDK8
<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>
- Install Java EE JDK7
<http://www.oracle.com/technetwork/java/javaee/downloads/java-ee-sdk-7-downloads-1956236.html>
- Install Netbeans 8.2
<https://dlc-cdn.sun.com/netbeans/8.2/final/?pagelang=>
- Install MySQL 5.0 and MySQL WorkBench 8.0

Set path and Path variables in the operating system

Click on Enviroment Variables.

Find the Path system variable and click Edit. Set the value of the variable to the directory where you have installed Java, for example:

D:\Program Files\Java\jdk1.8.0\bin



Ensure that the required JDK software is installed on your system and that the `JAVA_HOME` environment variable points to the JDK installation directory, not the Java Runtime Environment (JRE) software.

Environment Variables



User variables for Admin

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit PDF Reader\plugi...
OneDrive	C:\Users\Admin\OneDrive
Path	C:\Users\Admin\AppData\Local\Programs\Python\Python312...
TEMP	C:\Users\Admin\AppData\Local\Temp
TMP	C:\Users\Admin\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk-1.8
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files\Java\jdk-1.8\bin;C:\ProgramData\Oracle\Jav...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

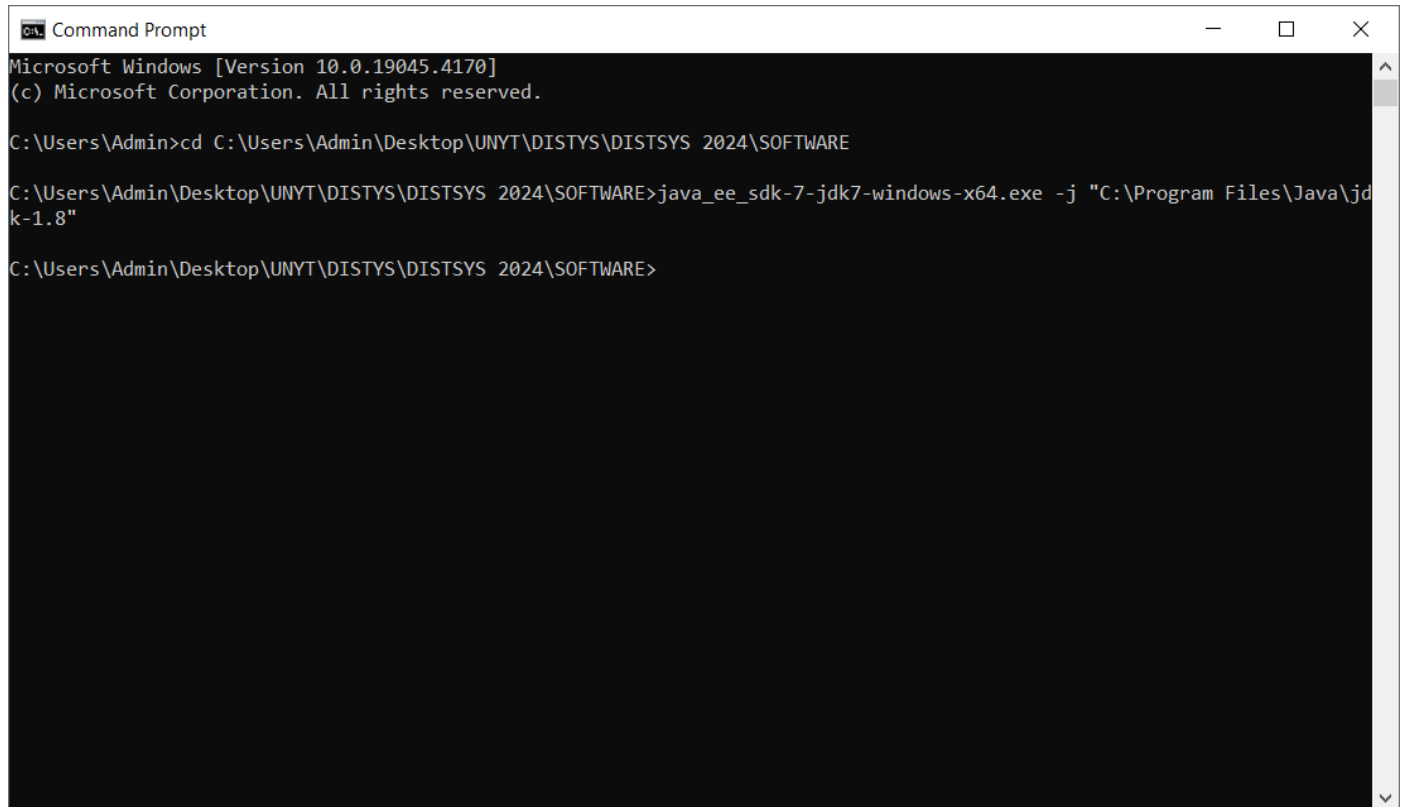
New... Edjt... Delete

OK Cancel

Download and install the Netbeans IDE by double clicking the executable installation file.

Download and install Java EE SDK.

If the .exe file does not start use the following command:



```
Command Prompt
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\Desktop\UNYT\DISTYS\DISTSYS_2024\SOFTWARE

C:\Users\Admin\Desktop\UNYT\DISTYS\DISTSYS_2024\SOFTWARE>java_ee_sdk-7-jdk7-windows-x64.exe -j "C:\Program Files\Java\jdk-1.8"

C:\Users\Admin\Desktop\UNYT\DISTYS\DISTSYS_2024\SOFTWARE>
```



Introduction

- Introduction**
- Installation Type
- Install Directory
- Update Tool
- Ready To Install
- Progress
- Config Results
- Summary

Welcome to the Java EE 7 SDK installation.

This installer will guide you through the installation process. You will shortly be able to learn the latest Java EE 7 features, and you can get started with the First Cup and Java EE Tutorials. View sample application source code and then deploy to GlassFish Server 4.0 to see them in action. You will find that Java EE 7 is a easy-to-learn feature-rich platform for developing web and enterprise applications.



ORACLE

Cancel

Back

Next



Installation Type

- Introduction
- Installation Type**
- Install Directory
- Update Tool
- Ready To Install
- Progress
- Config Results
- Summary



ORACLE

Choose installation type.

Typical Installation

Installs a GlassFish Server management domain; ideal for development or non business critical use. Please make sure that the ports 4848 and 8080 are free.

Custom Installation

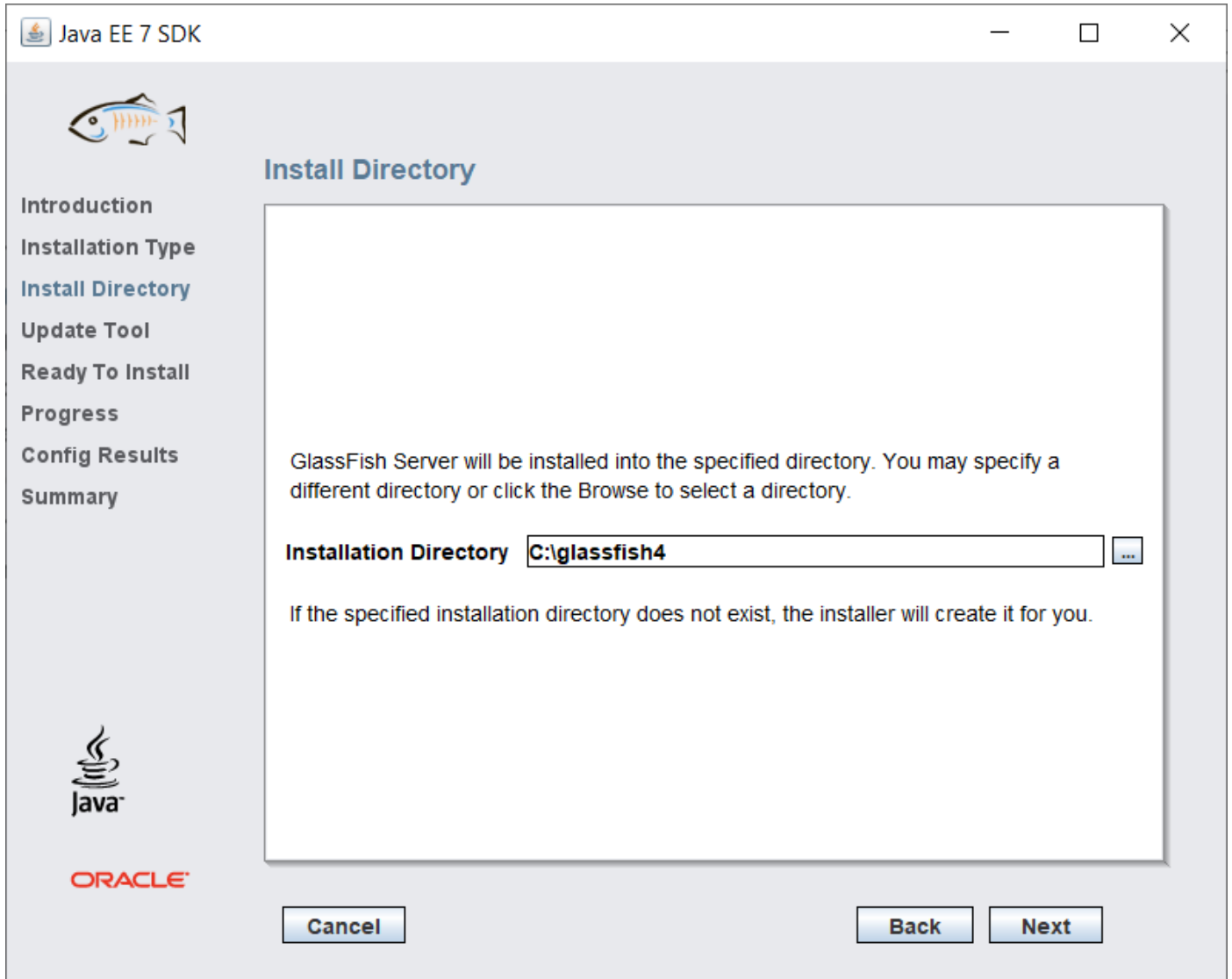
Not supported.

Cancel

Back

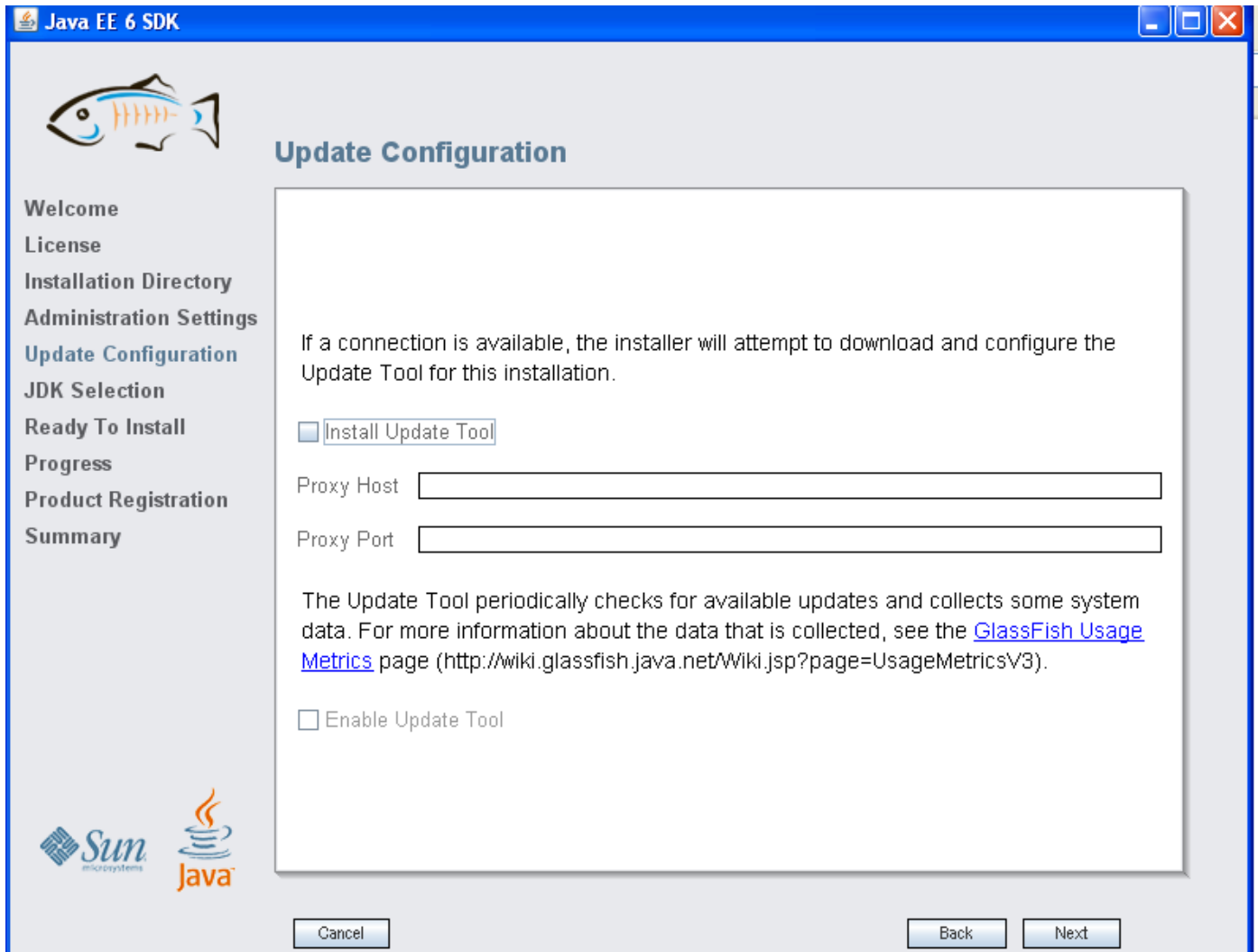
Next

Choose directory for Glassfish:



The screenshot shows a window titled "Java EE 7 SDK" with a standard Windows title bar. On the left is a sidebar with a fish icon and a list of steps: Introduction, Installation Type, **Install Directory**, Update Tool, Ready To Install, Progress, Config Results, and Summary. The main area is titled "Install Directory" and contains the following text: "GlassFish Server will be installed into the specified directory. You may specify a different directory or click the Browse to select a directory." Below this is a text box labeled "Installation Directory" containing the text "C:\glassfish4" and a browse button "...". Further down, it says "If the specified installation directory does not exist, the installer will create it for you." At the bottom are three buttons: "Cancel", "Back", and "Next". The Java logo and "ORACLE" logo are visible in the bottom left corner.

Click again Next:





Ready To Install

- Introduction
- Installation Type
- Install Directory
- Update Tool
- Ready To Install**
- Progress
- Config Results
- Summary

Java EE 7 SDK

- Install JDK
- Install Update Tool Bootstrap
- Install GlassFish Server
- Install Uninstallation Software
- Configure Update Tool Bootstrap
- Configure GlassFish Server



ORACLE

Cancel

Back

Install



Progress

- Introduction
- Installation Type
- Install Directory
- Update Tool
- Ready To Install
- Progress**
- Config Results
- Summary

Java EE 7 SDK

Modular, Lightweight, Open

- Modular architecture based on OSGi
- Fast startup, less memory consumption
- Java EE 7 Certified
- Developed in Open Source



ORACLE

Installing GlassFish Server



Cancel

Back

Next



- Introduction
- Installation Type
- Install Directory
- Update Tool
- Ready To Install
- Progress
- Config Results**
- Summary



ORACLE

Config Results

The configuration has succeeded. Please see the output below.

```
Domain domain1 allows admin login as user "admin" with no password.  
Login information relevant to admin user name [admin]  
for this domain [domain1] stored at  
[C:\Users\Admin\gfclient\pass] successfully.  
Make sure that this file remains protected.  
Information stored in this file will be used by  
administration commands to manage this domain.  
Command create-domain executed successfully.
```

Starting domain

```
Executing command :C:\glassfish4\glassfish\bin\asadmin.bat start-domain  
domain1
```

```
C:\glassfish4\glassfish\bin\asadmin.bat start-domain domain1  
Attempting to start domain1.... Please look at the server log for more  
details.....
```

Cancel

Configure again

Next



Summary

- Introduction
- Installation Type
- Install Directory
- Update Tool
- Ready To Install
- Progress
- Config Results
- Summary



ORACLE

Overall Status: Complete

Please see the [detailed summary report](#) for an overview of this session, including [next steps](#) for using this installation. Please see the [log file](#) for detailed information.

[2024-04-11-14-47-install-summary.html](#)

[2024-04-11-14-47-install.log](#)

Product Name	Status
	Installed
Update Tool Bootstrap	Installed
GlassFish Server	Installed
Uninstallation Software	Installed
Update Tool Bootstrap	Configured
GlassFish Server	Configured

Cancel

Back

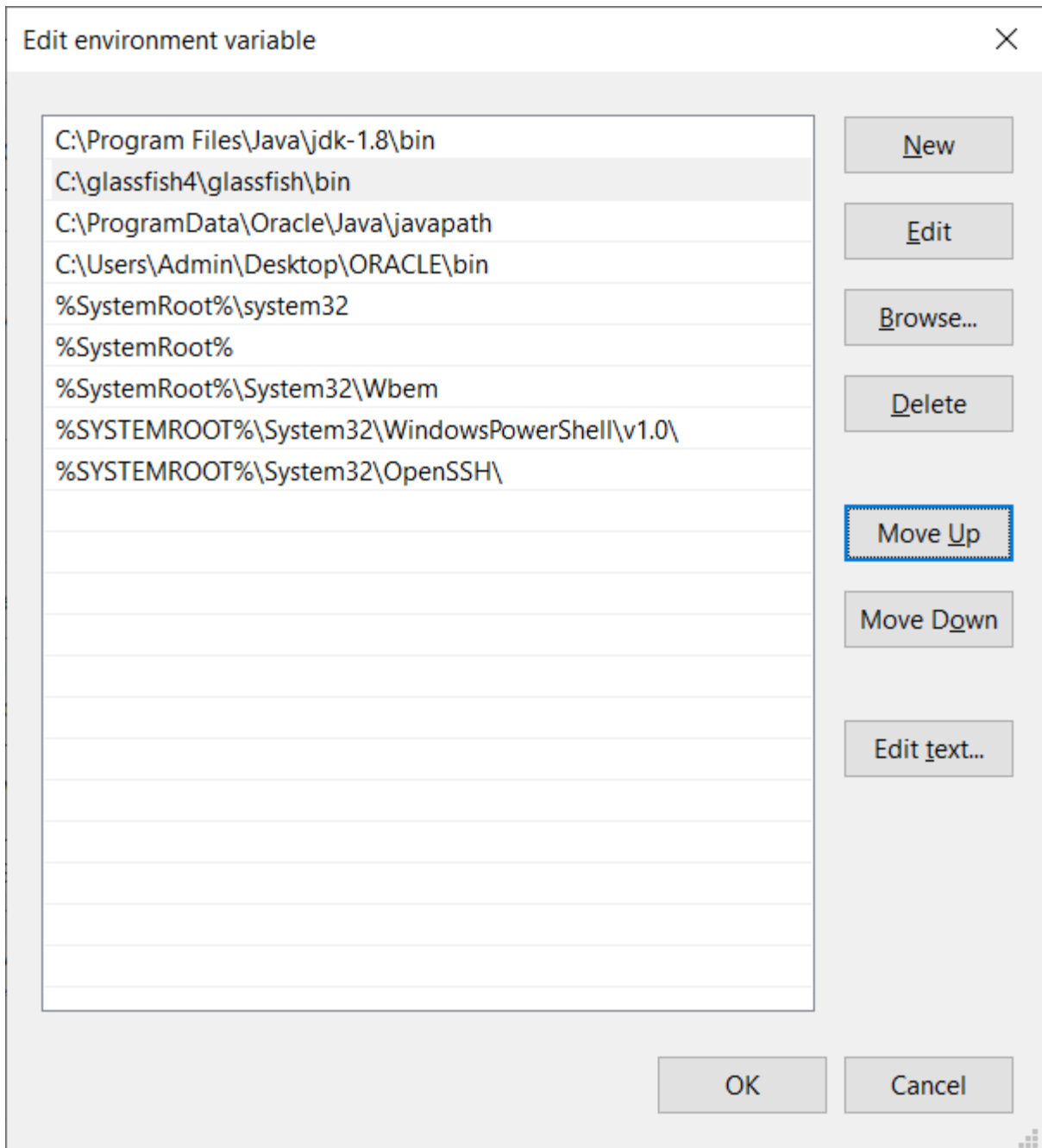
Exit

Only if “Overall Status” is “Complete”, your installation has been performed appropriately.

In order for the examples of this tutorial to execute you need to set the PATH with the directory of Glassfish as follows:

Click on Environment Variables:

Find the Path variable and click Edit.



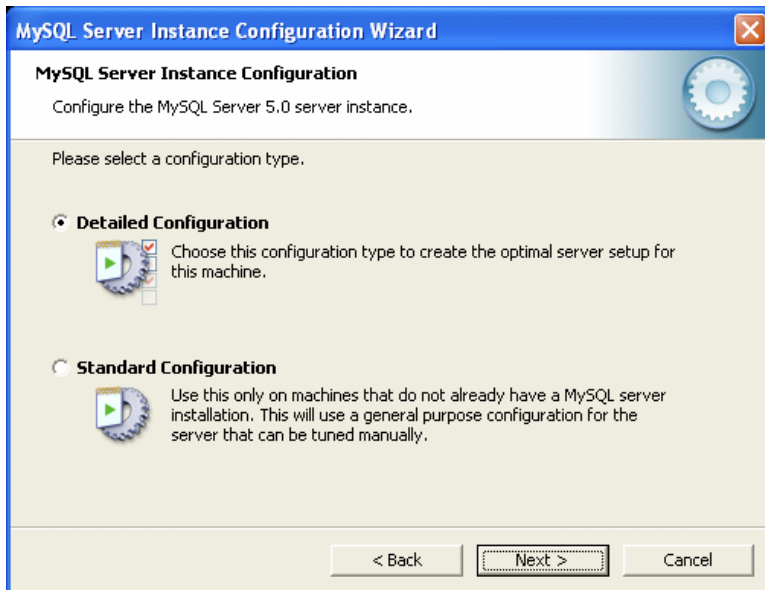
In the variable value add the path of Glassfish.

Download and Install MySQL Server:

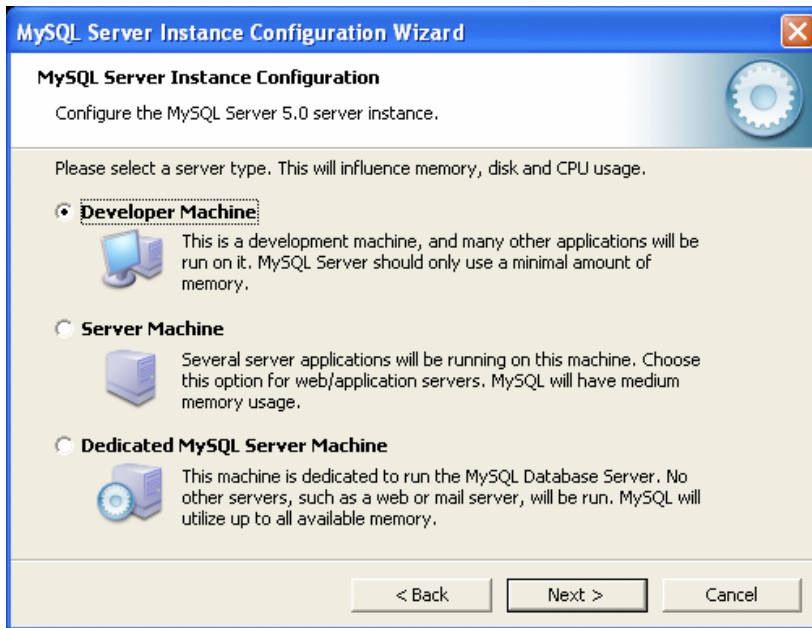
After you download use MySQL Server Instance Config Wizard



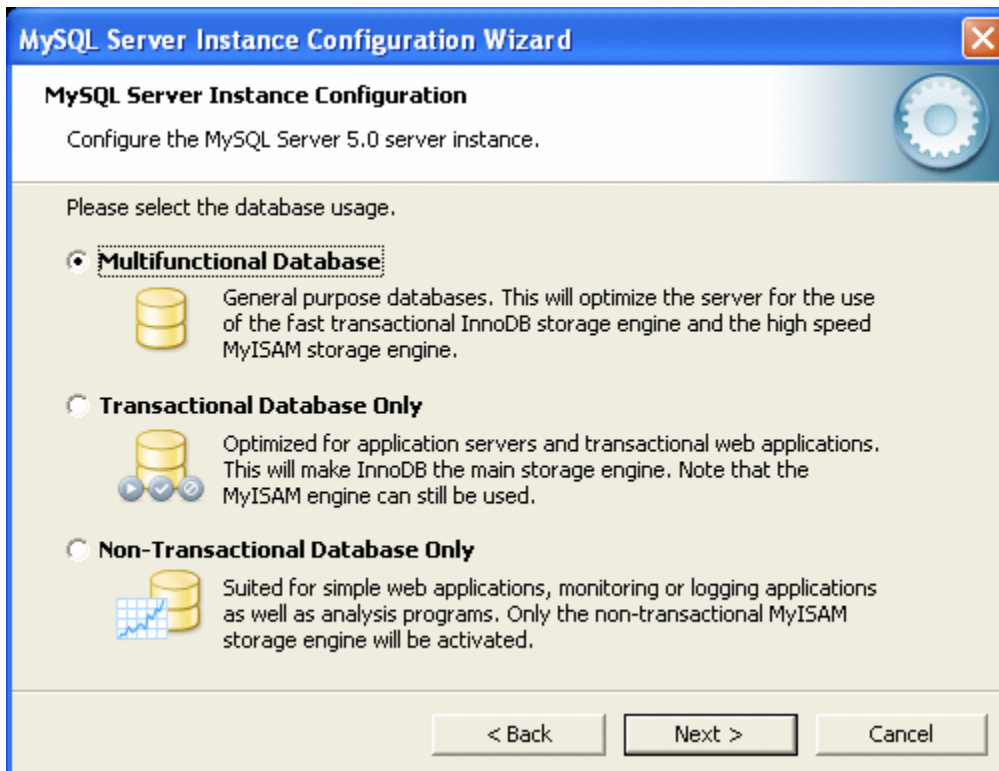
Choose the detailed configuration:



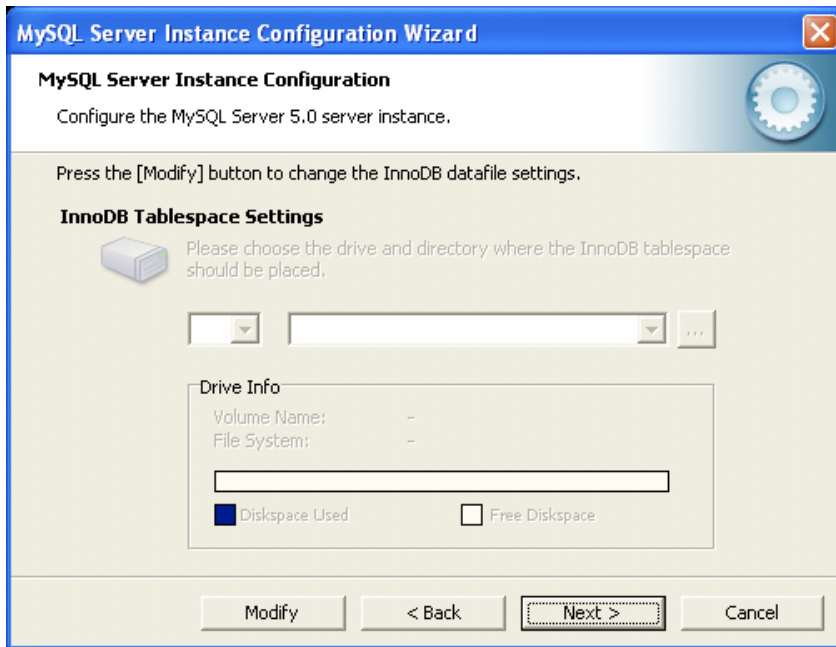
Choose developer machine:



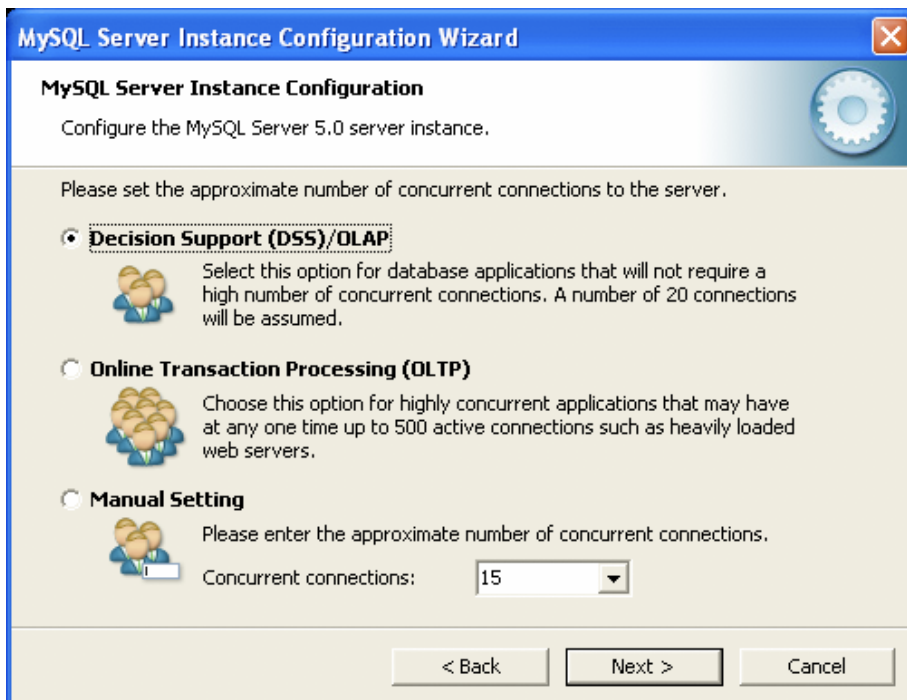
Choose multifunctional:



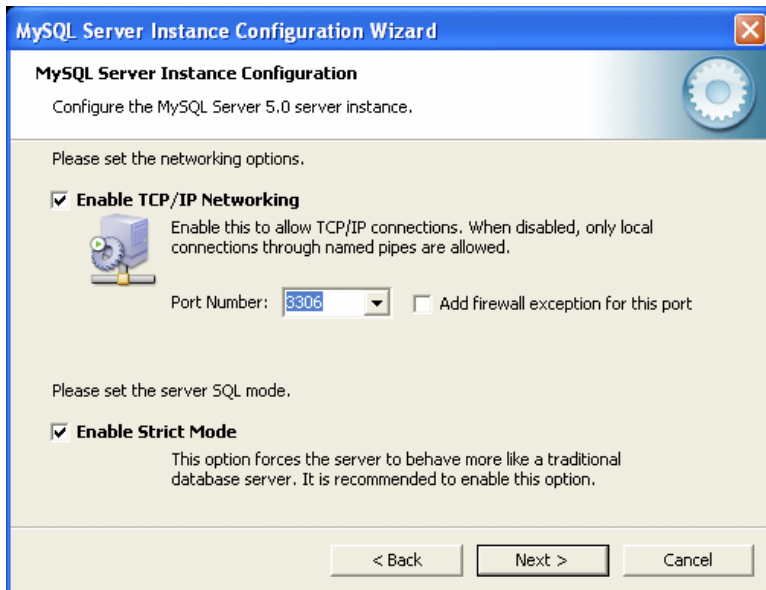
Choose Drive:



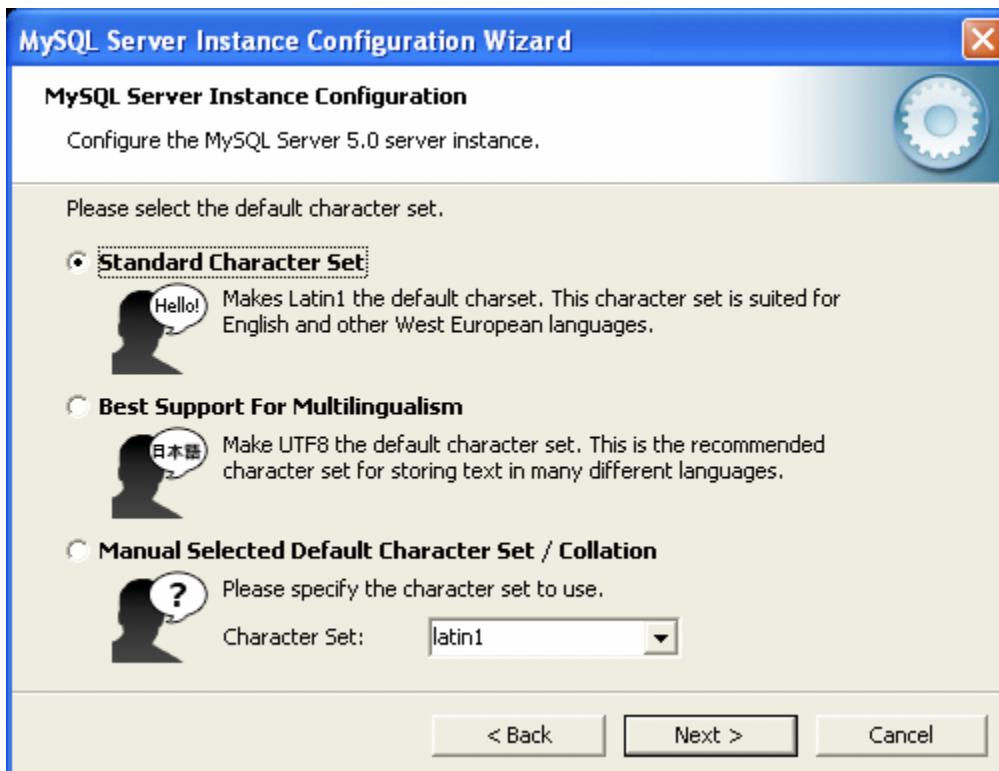
Choose decision support:



Perform the following checks:



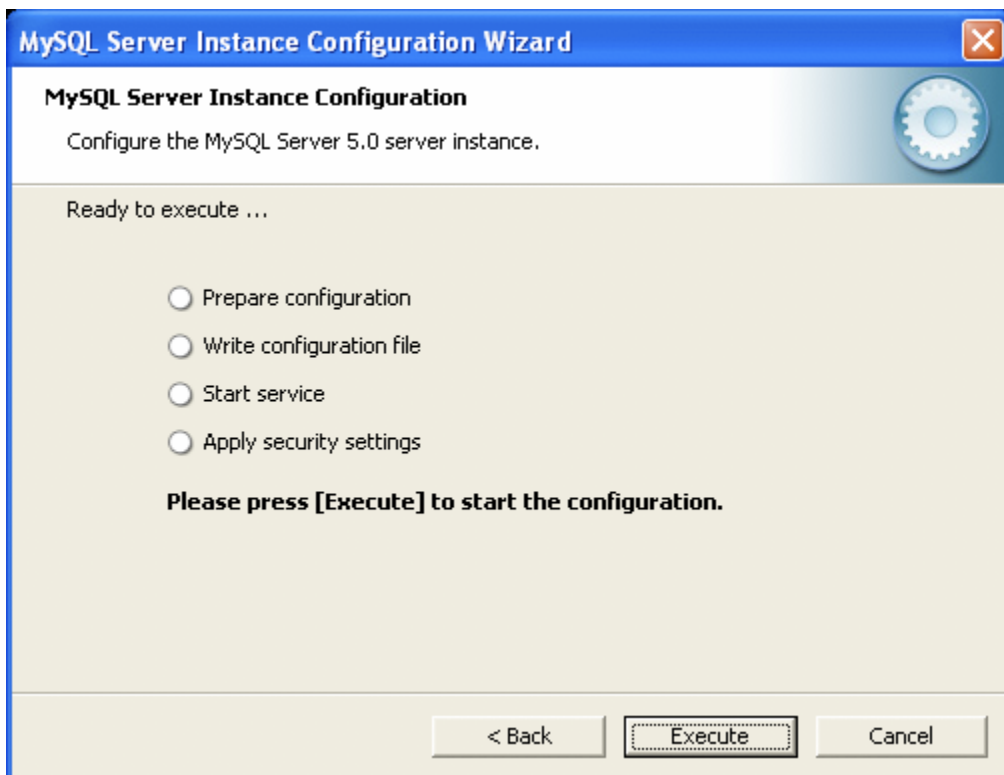
Best Support For Multilingualism: Choose this option if you want to use `utf8` as the default server character set. This is a Unicode character set that can store characters from many different languages.



Set the password for root:



Press Execute:



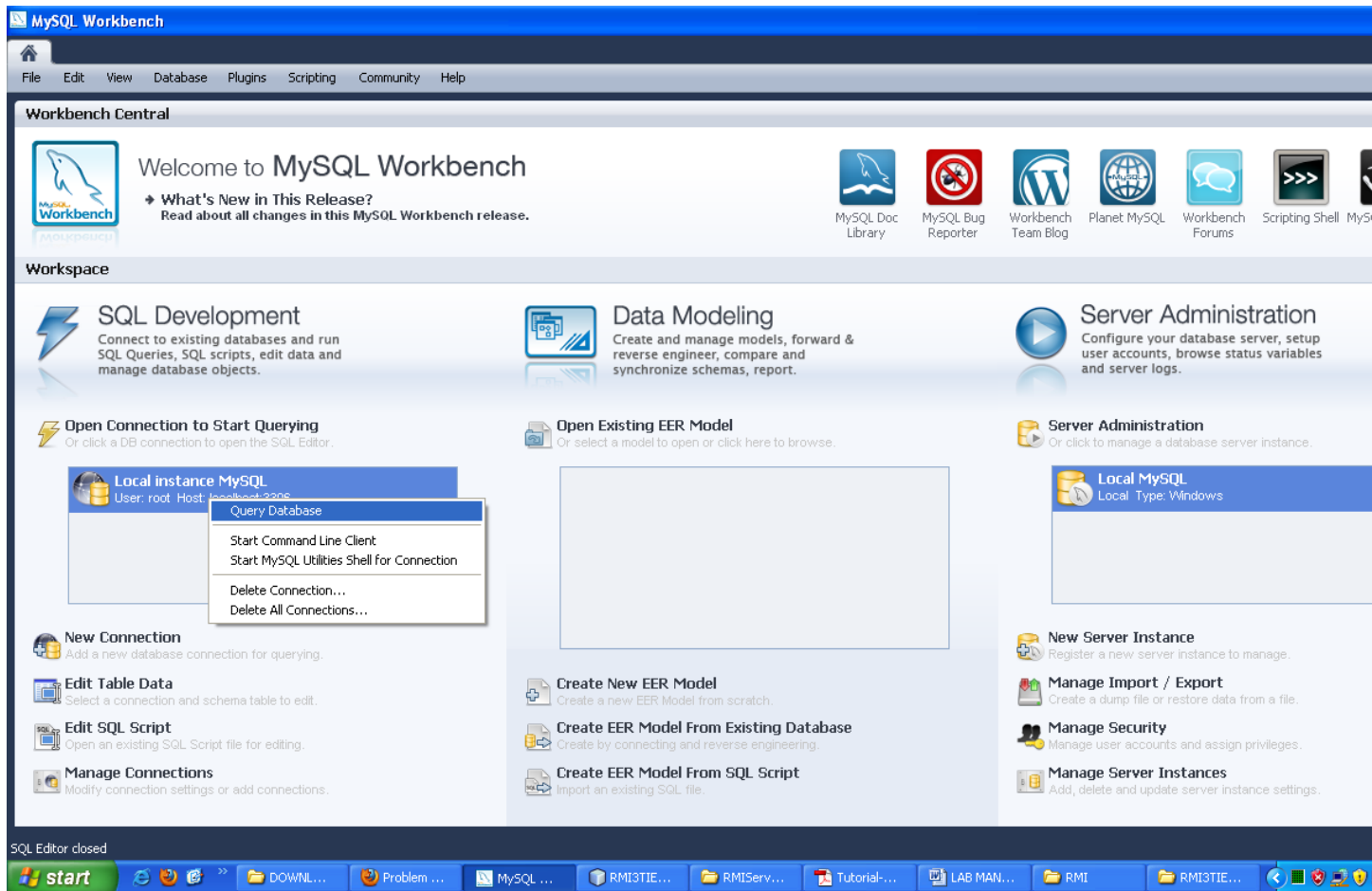
Restart the computer and the installation should be complete.

Install the MySQL Workbench.

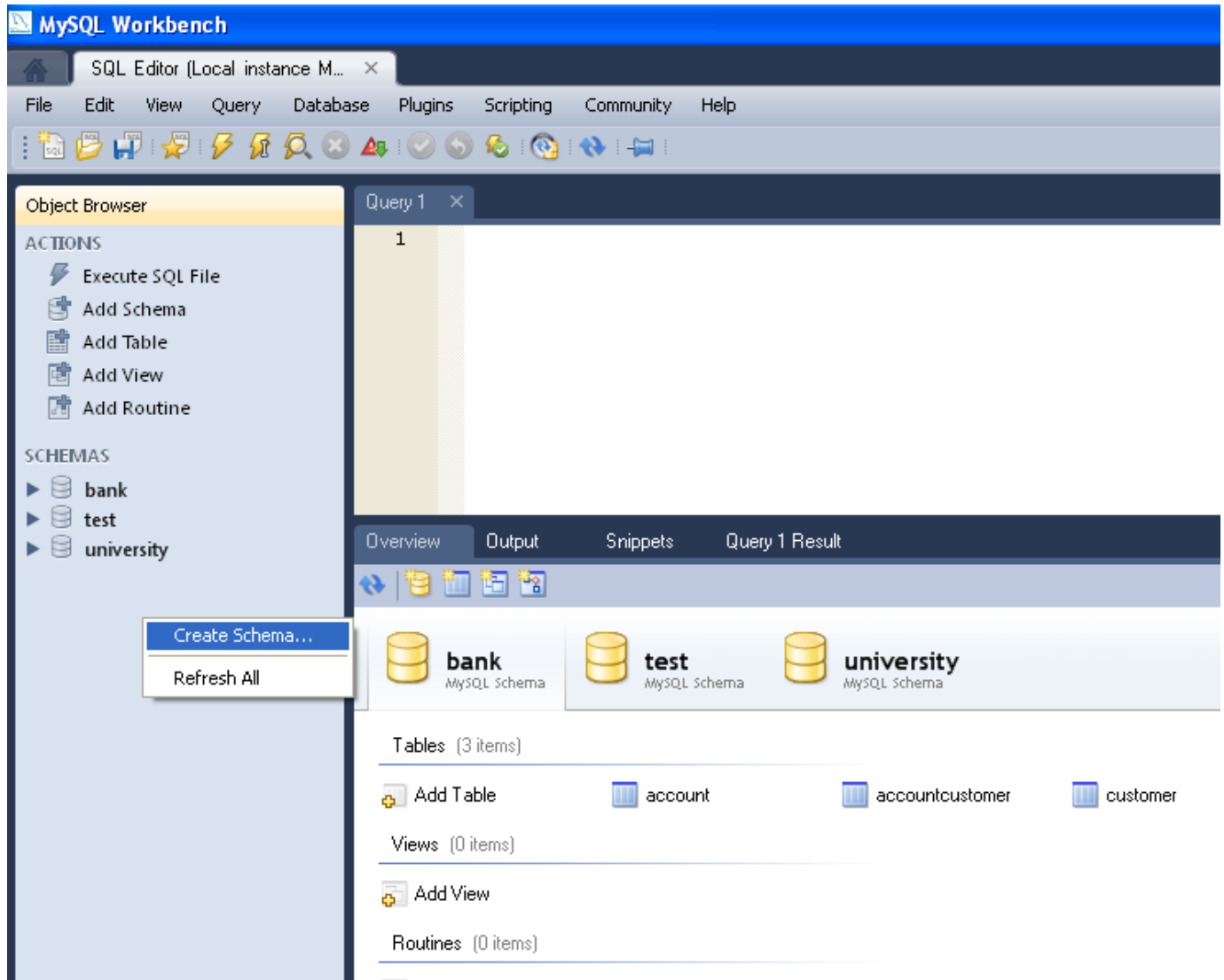
You can create the database in two ways:

1. By commands in the MySQL console
2. By graphical user interface in MySQL Workbench

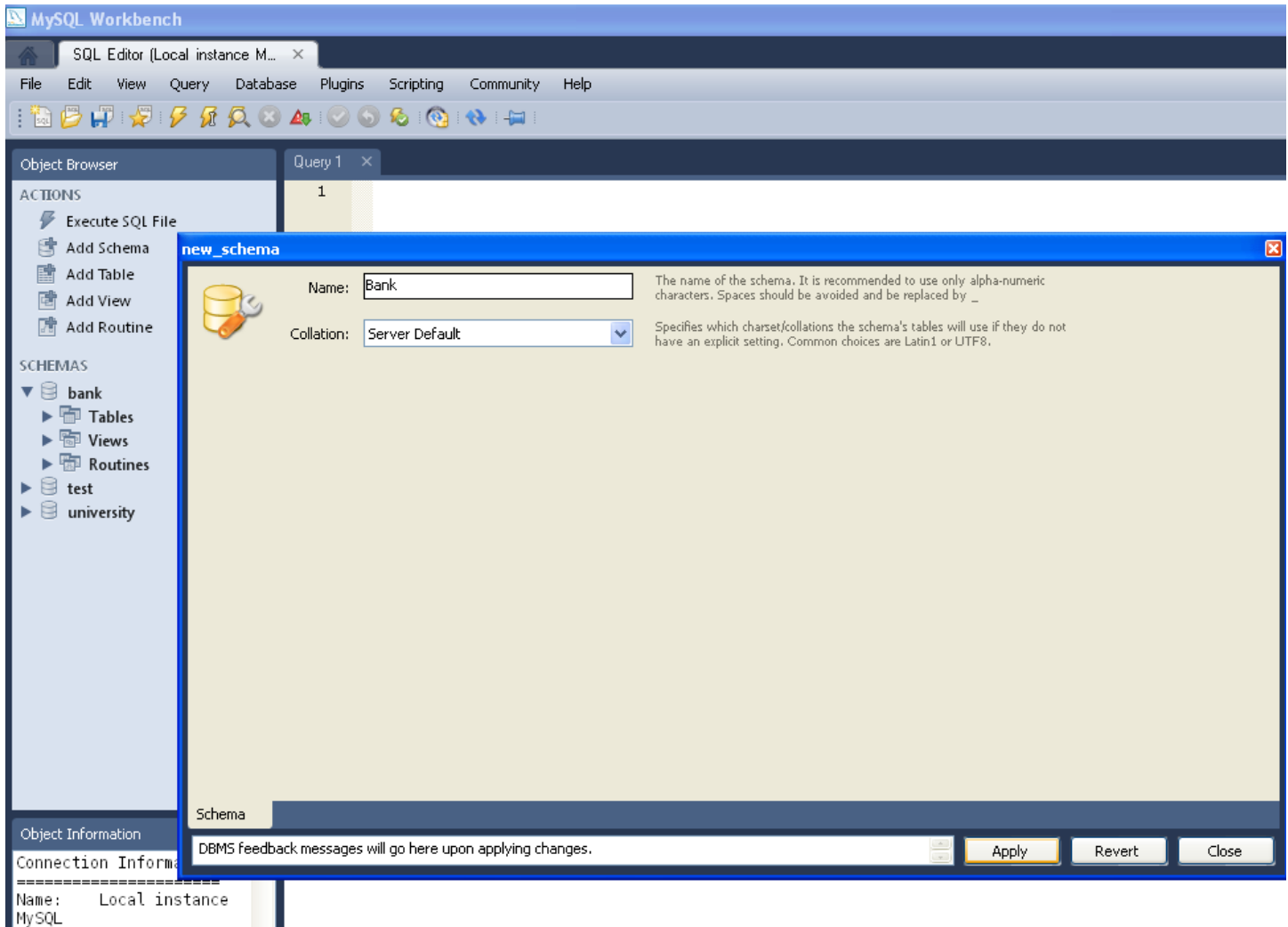
Click on local instance with the right and click Query Database.



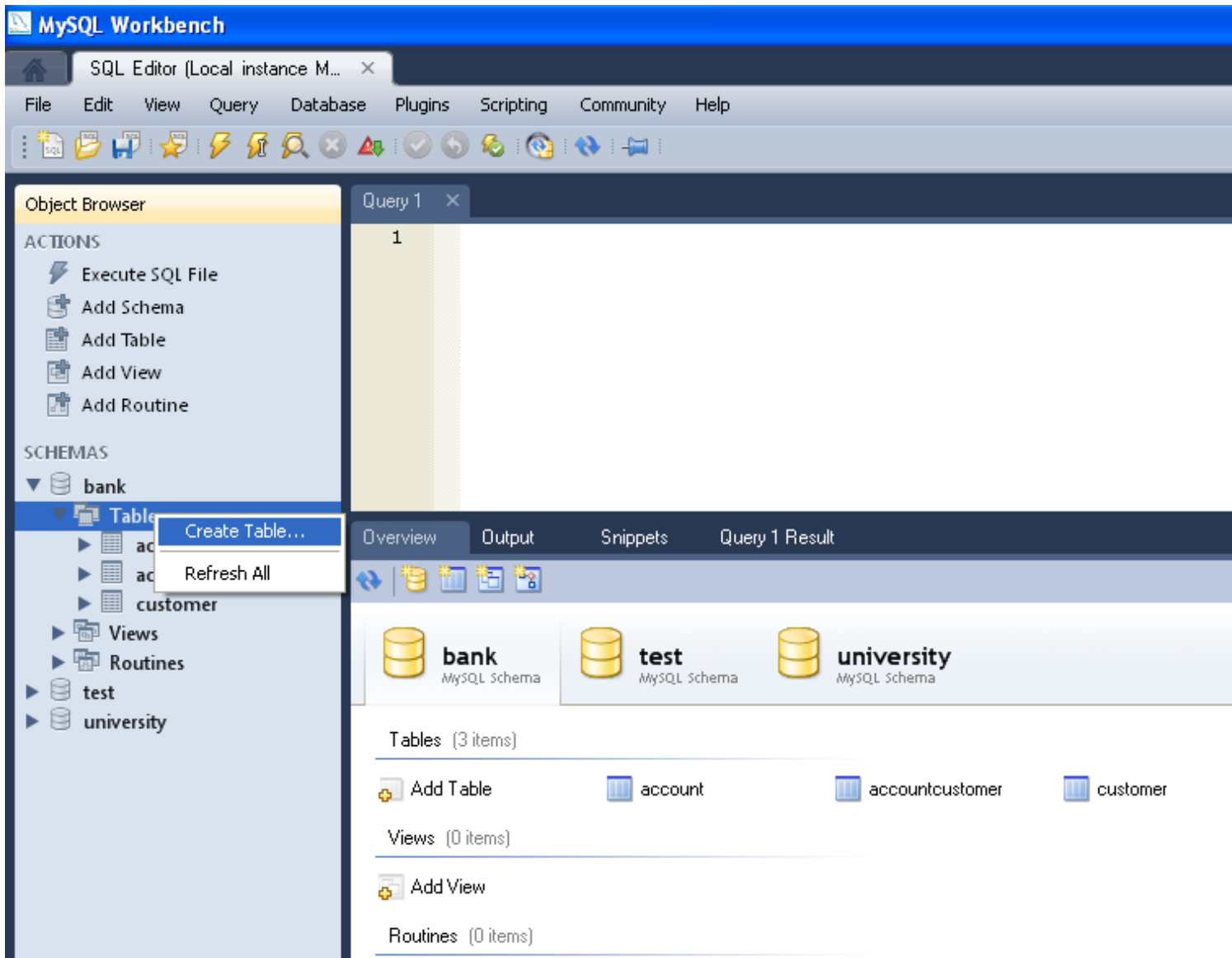
Click with the right and select create schema.



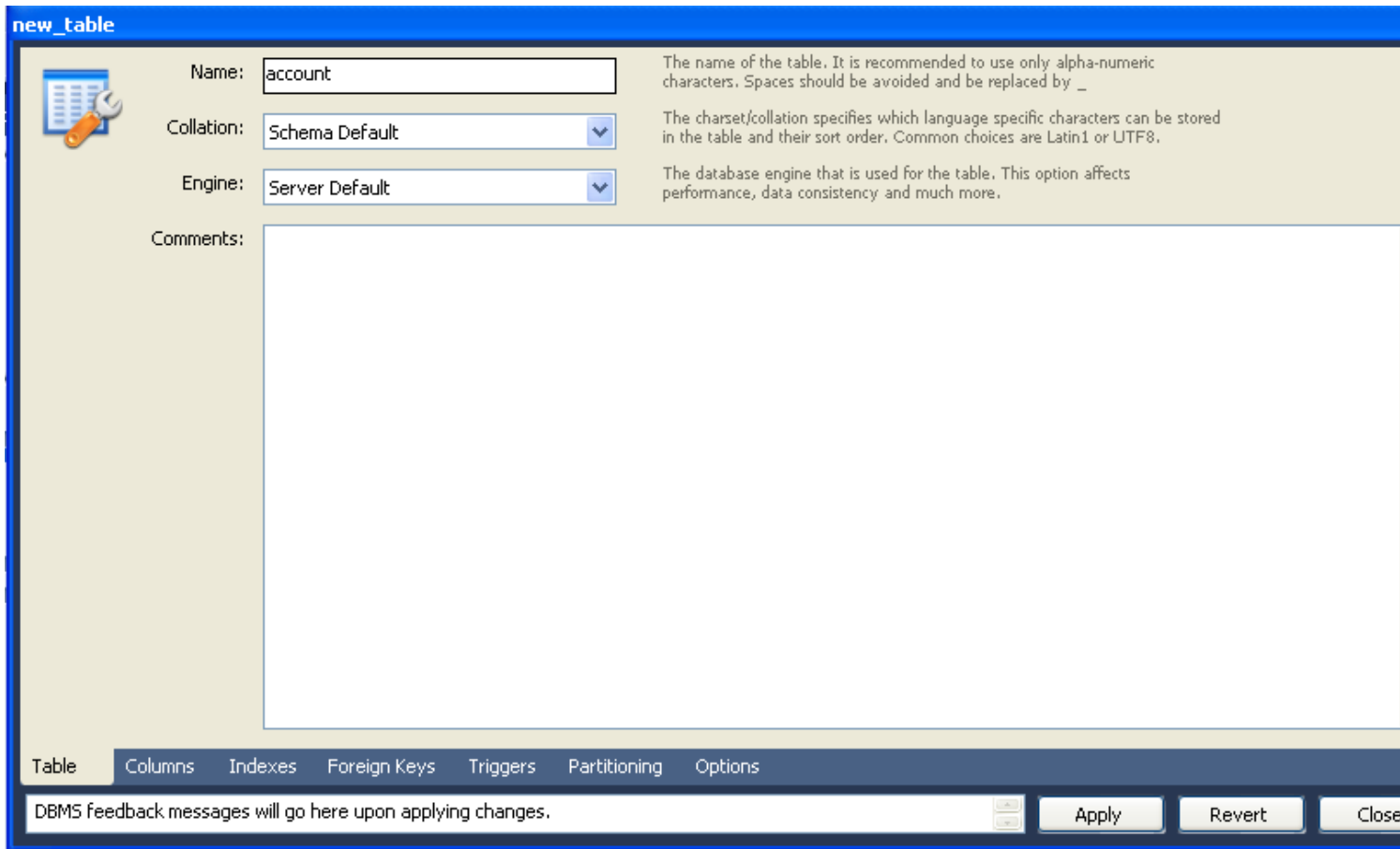
Give a name to the database: and press Apply.



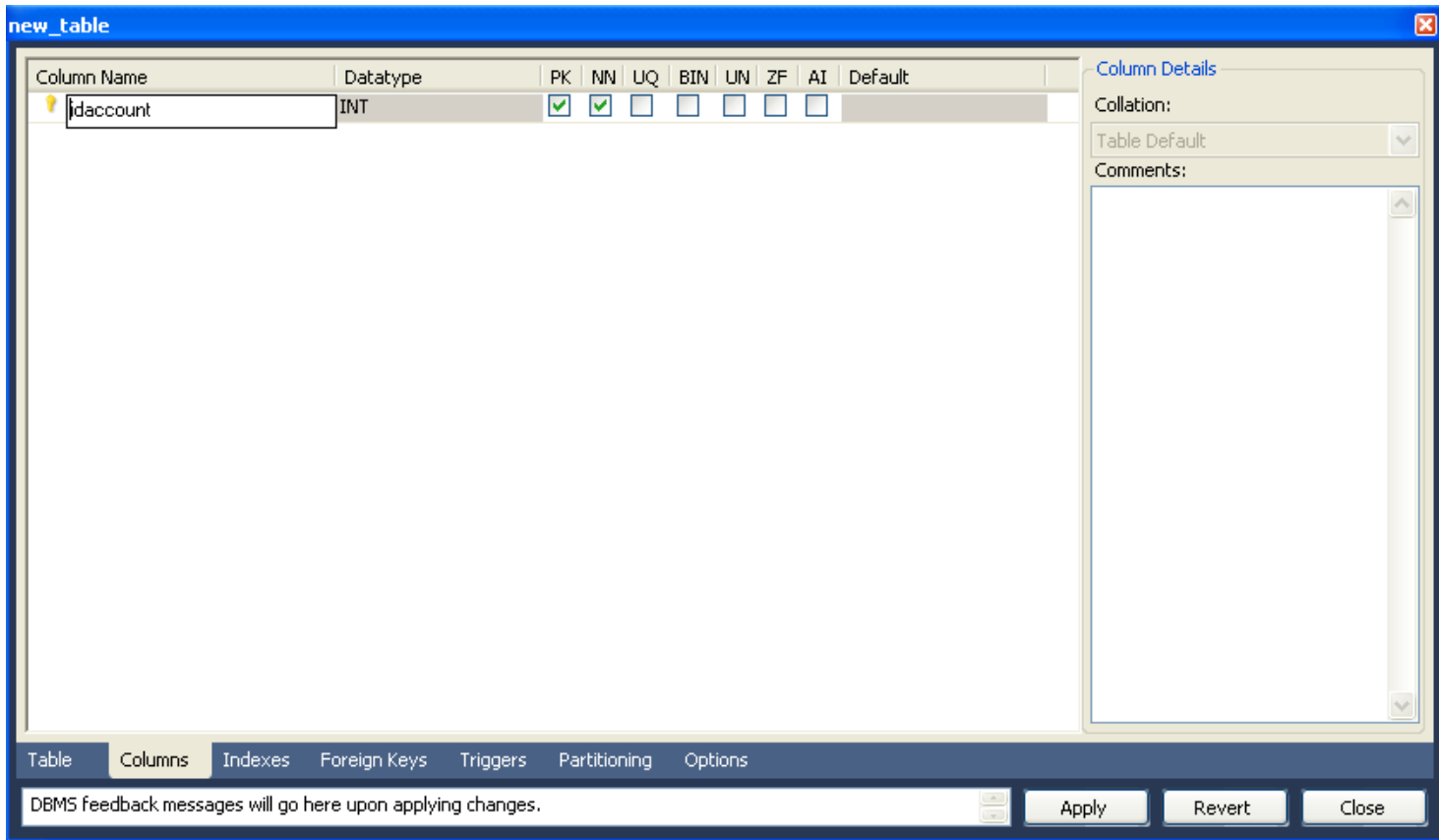
Click with the right on the Tables options and select Create Table:



Choose a name for the Table:



Click on Columns and add the columns for the table. At the end click Apply.



Following the above procedure create three tables:

Table Account

Fields: IdAccount (int), Balance (float)

Table Customer

Fields: idCustomer(int), Name (Varchar), surname (Varchar)

Table AccountCustomer

Fields: idAccount, IdCustomer

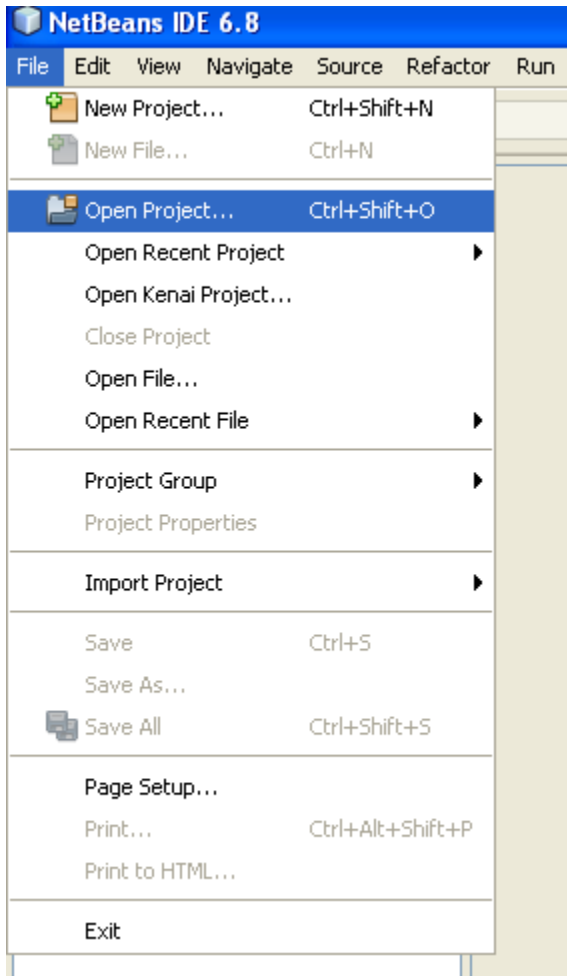
2. Running the EJB examples

You will be given some examples that illustrate the EJBs.

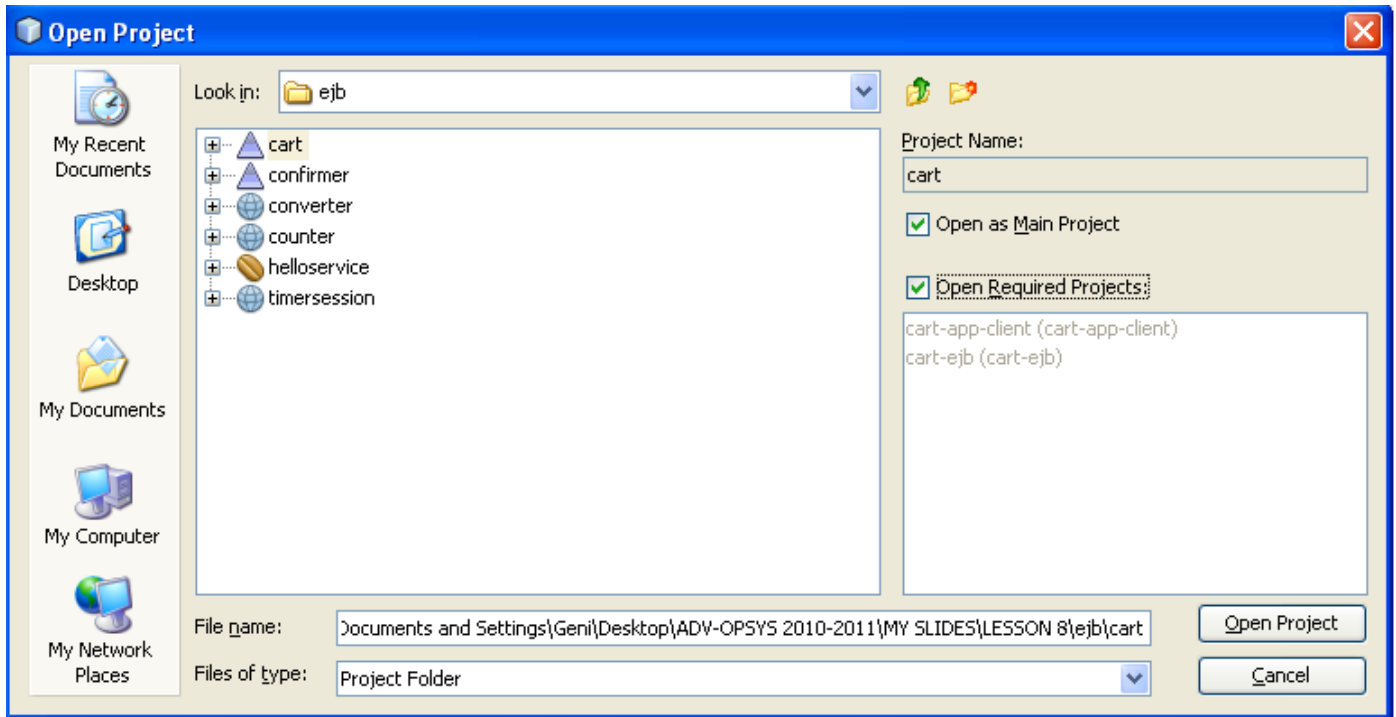
- CartBean: a stateful session bean that is accessed by a remote client
- CounterBean: a singleton session bean
- HelloServiceBean: a stateless session bean that implements a web service

- `TimerSessionBean`: a stateless session bean that sets a timer

To open an example perform these steps:



Select the main project:



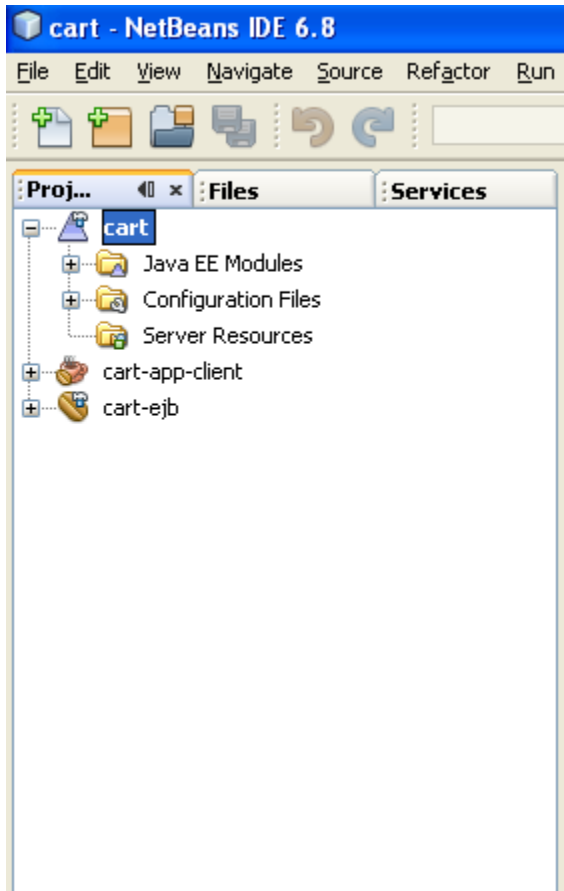
Check the options: “Open as Main Project” and “Open Required Projects” and press the button “Open Project”. The following window will appear:

Running the Cart example

The cart example represents a shopping cart in an online bookstore and uses a stateful session bean to manage the operations of the shopping cart. The bean's client can add a book to the cart, remove a book, or retrieve the cart's contents. To assemble cart, you need the following code:

Session bean class (CartBean)

Remote business interface (Cart)



In order to properly execute the example perform the following steps:

Create the following file named main.xml (this file comes also together with this manual):

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- Copyright 1997-2007 Sun Microsystems, Inc. All rights reserved.
```

```
 $Id: main.xml,v 1.7 2007/07/18 21:50:57 msreddy Exp $ -->
```

```

<!-- main.xml: this is the file that should be included by the project
      build files. It will figure out whether it is running from inside Netbeans
      or command line and include appropriate tasks.
      @Author: Inderjeet Singh -->
<project name="main" default="dummy-default">

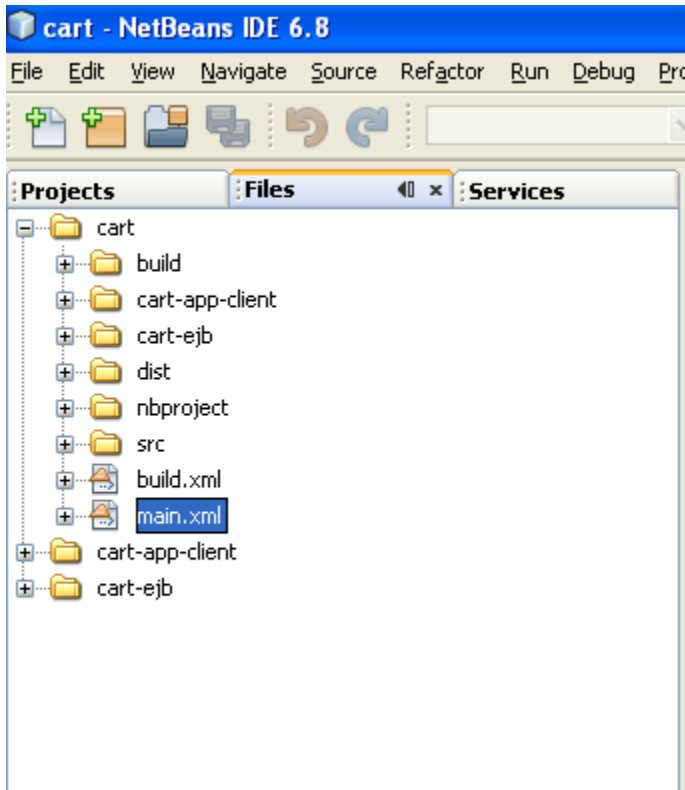
  <condition property="common-ant-tasks-file"
            value="\${ant.file}/../nbproject/build-impl.xml"
            else="\${ant.file.main}/../command-line-ant-tasks.xml">
    <and>
      <isset property="netbeans.home"/>
      <available file="\${ant.file}/../nbproject/build-impl.xml"/>
    </and>
  </condition>

  <import file="\${common-ant-tasks-file}"/>

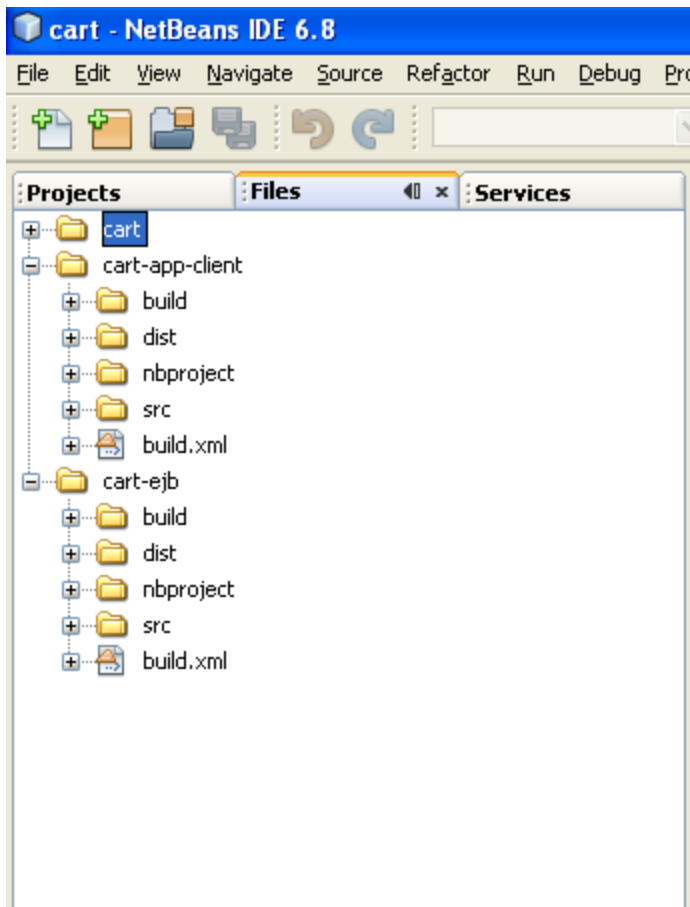
  <target name="dummy-default"/>
</project>

```

Copy the file under the directory of the project you want to open (The window below is switched to tab Files in Netbeans).

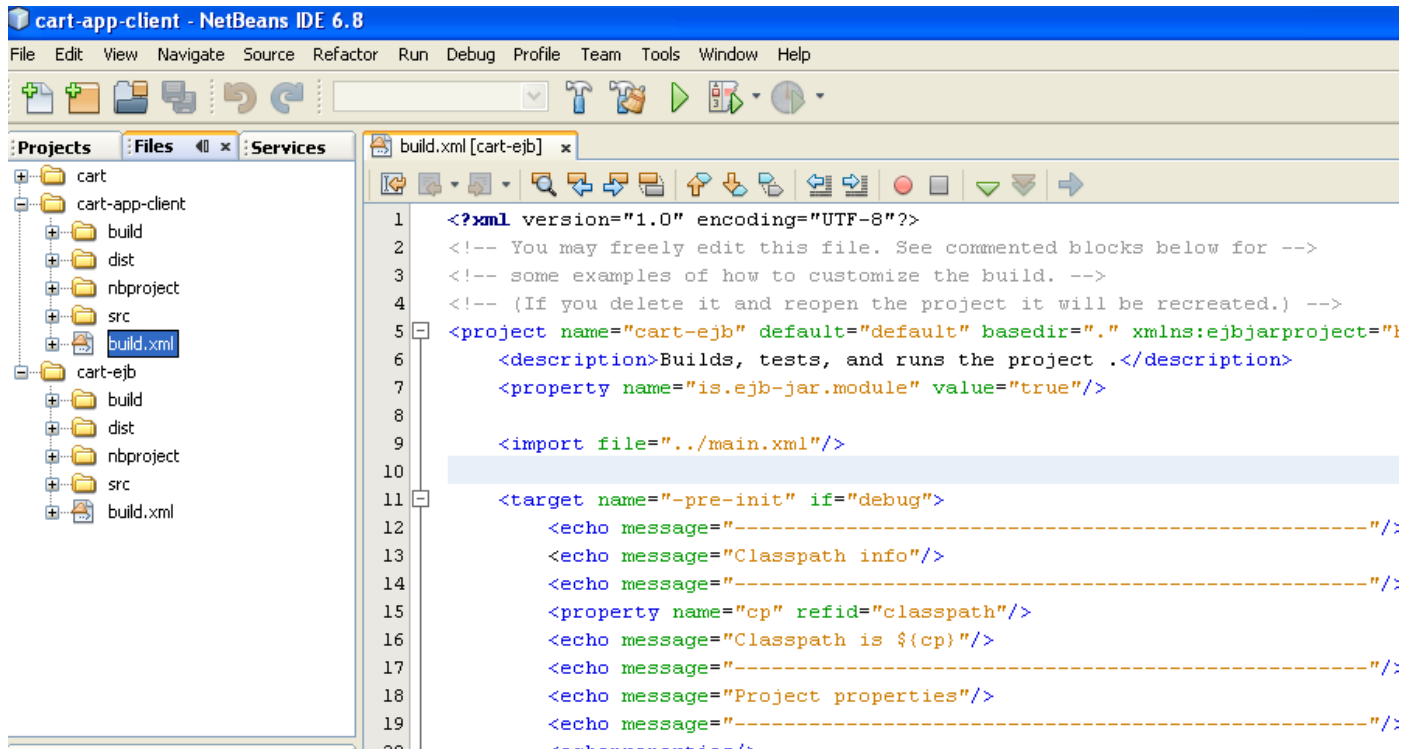


In the project that you are building, locate the files build.xml as follows:



In each of these files add the following line:

```
<import file="../main.xml"/>
```

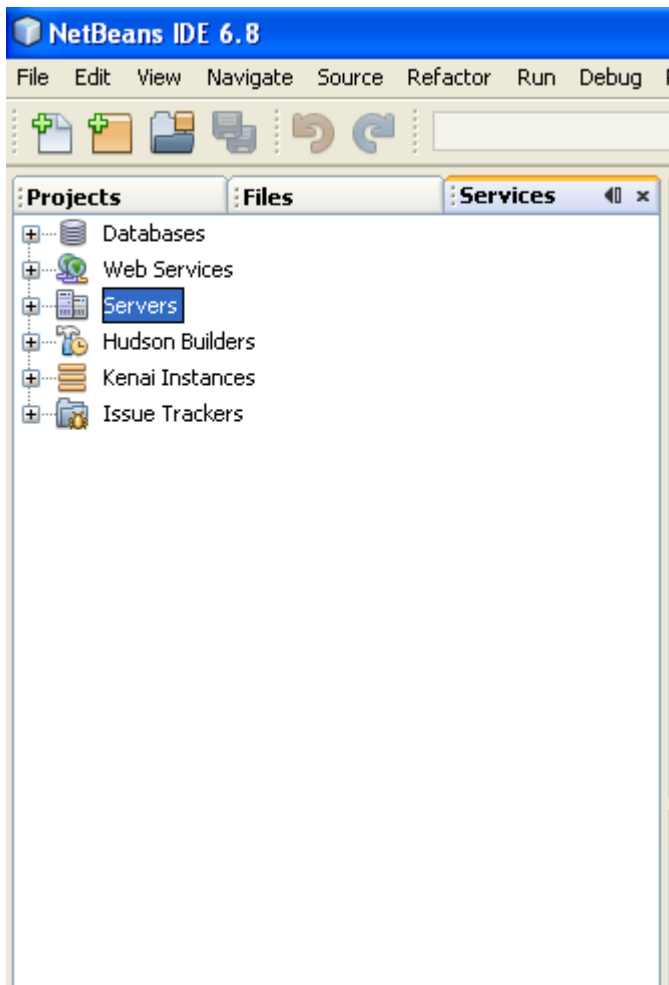


Running the examples:

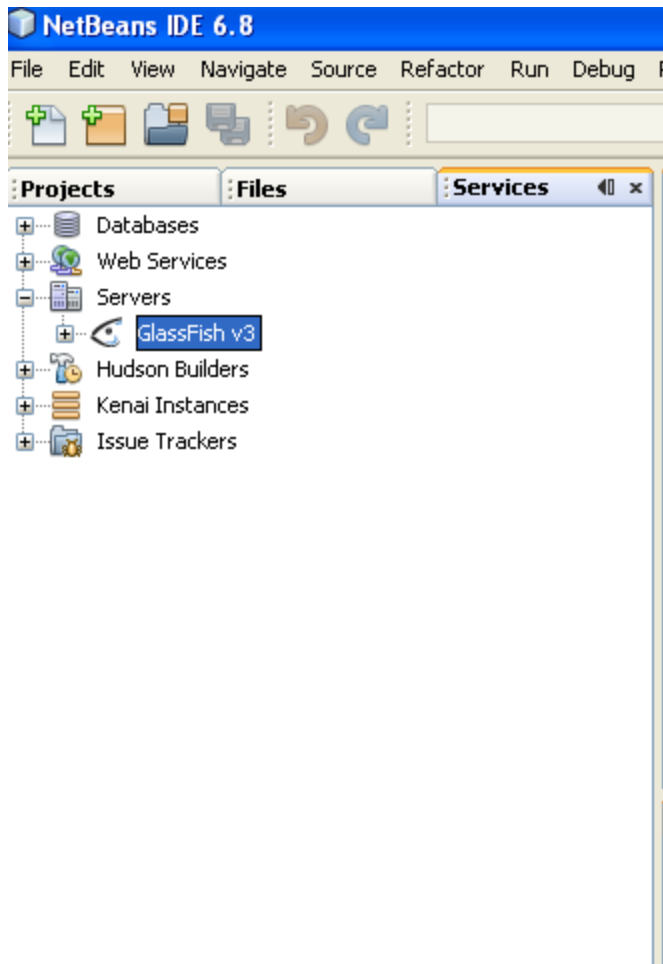
Once the application is loaded you need to start the Glassfish server:

To run the application you first need to start the server. You can start the server with the following operations:

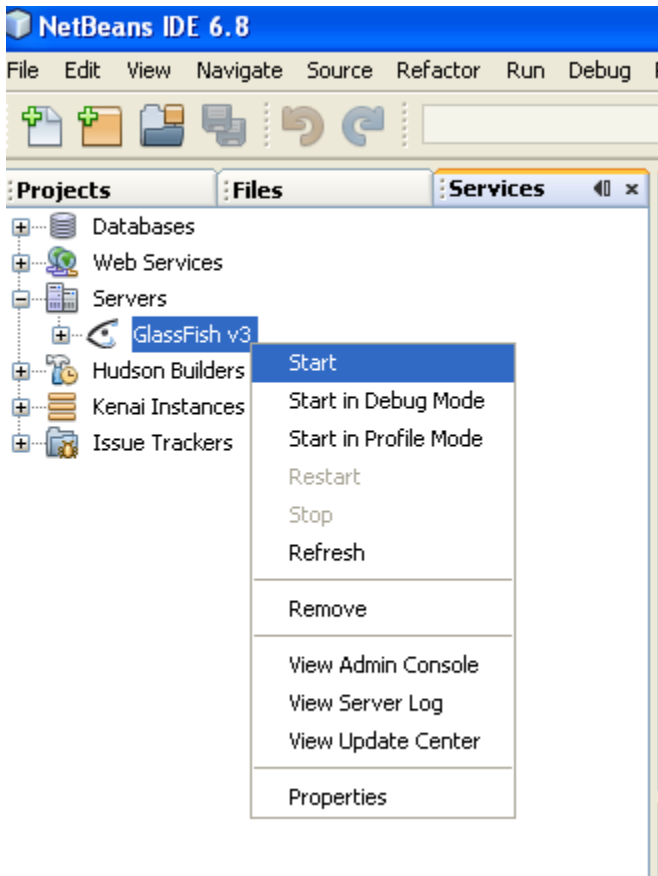
Click the tab Services in NetBeans:



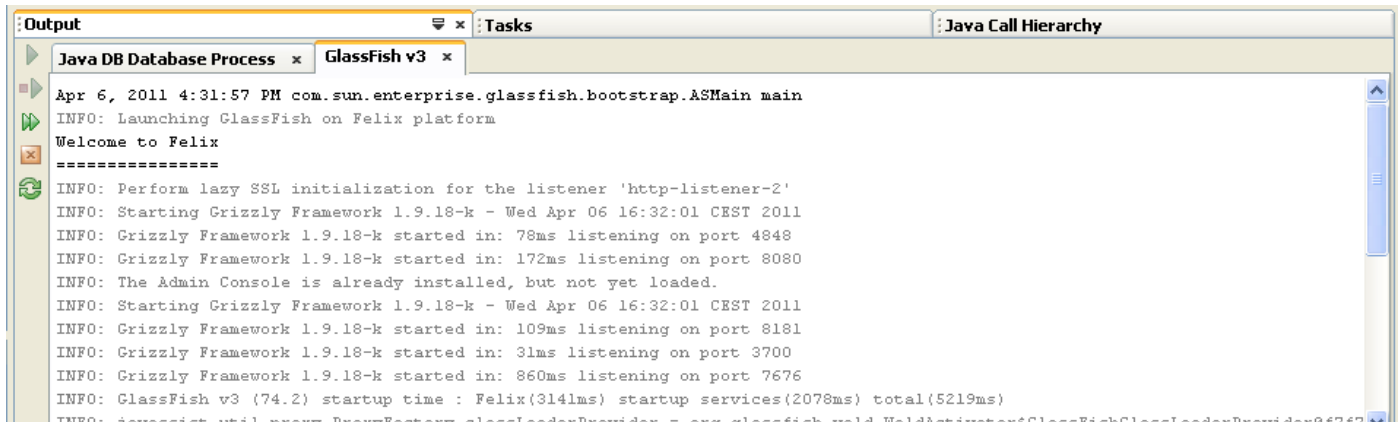
Select Servers:



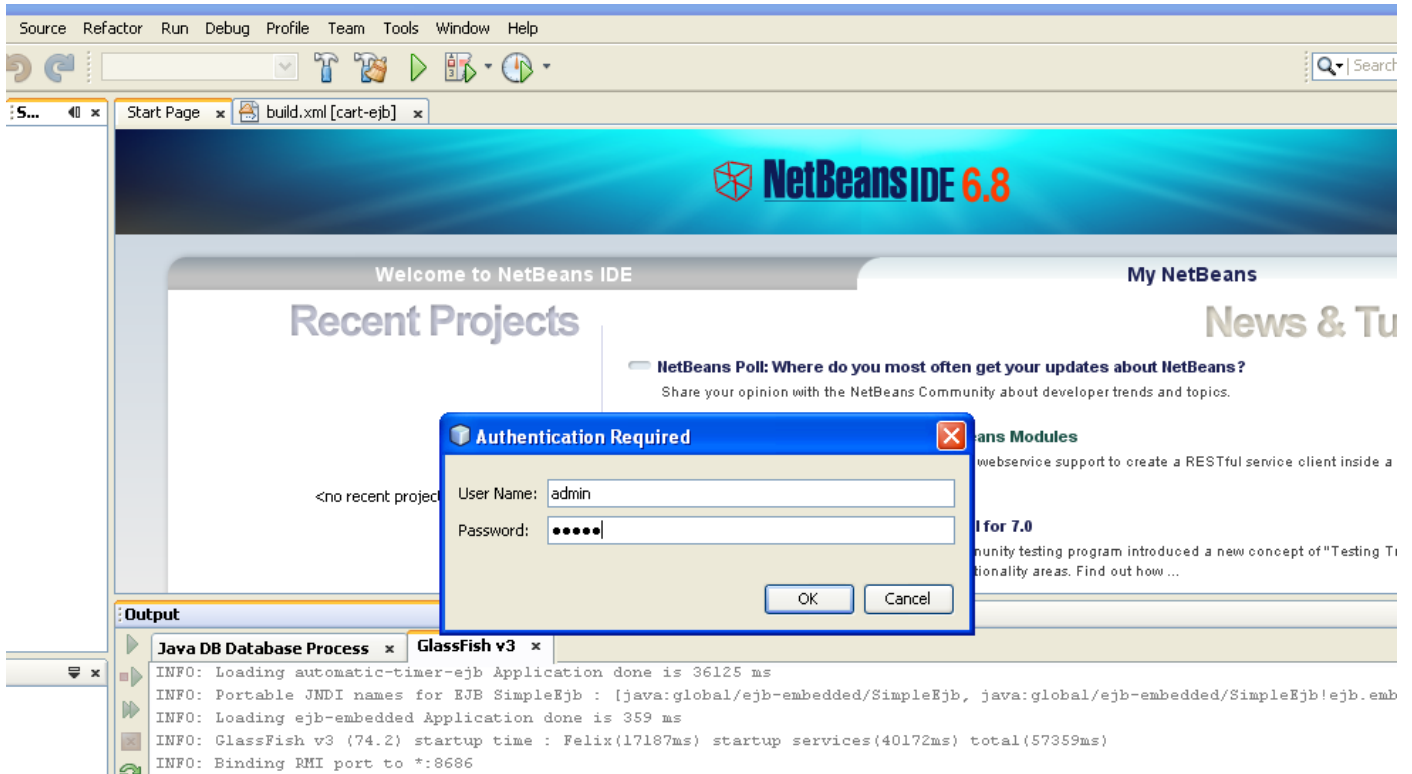
Click with the right on GlassFish and select “Start”



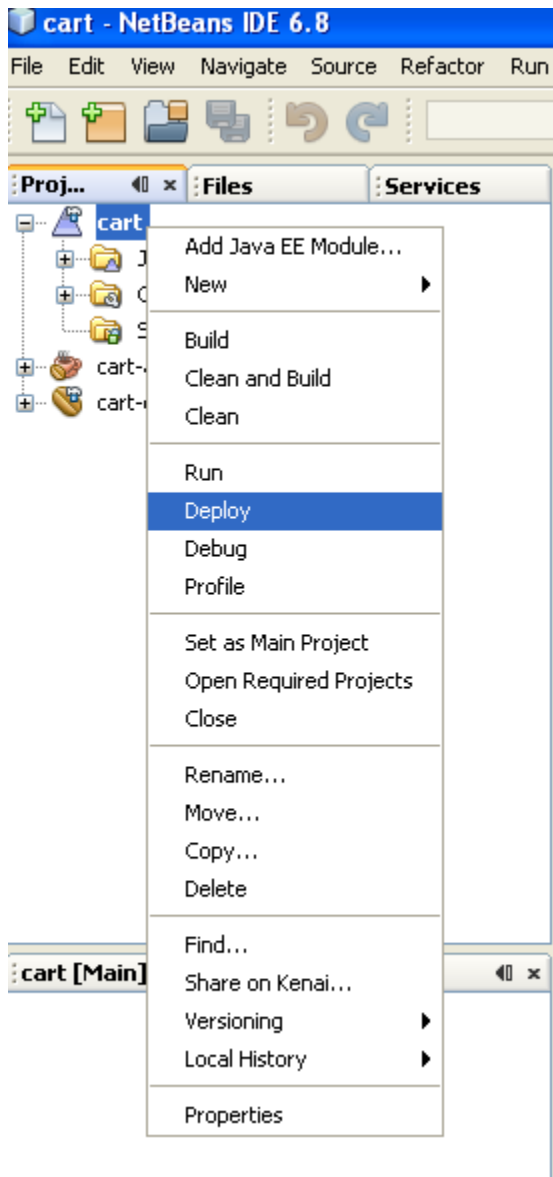
In the console of NetBeans you will see:



The admin password will be required if you provided one at the installation step:



After this you can deploy the application that you developed by selecting “Deploy” as follows:



You will see the following:

<no recent project>

module or Java application.

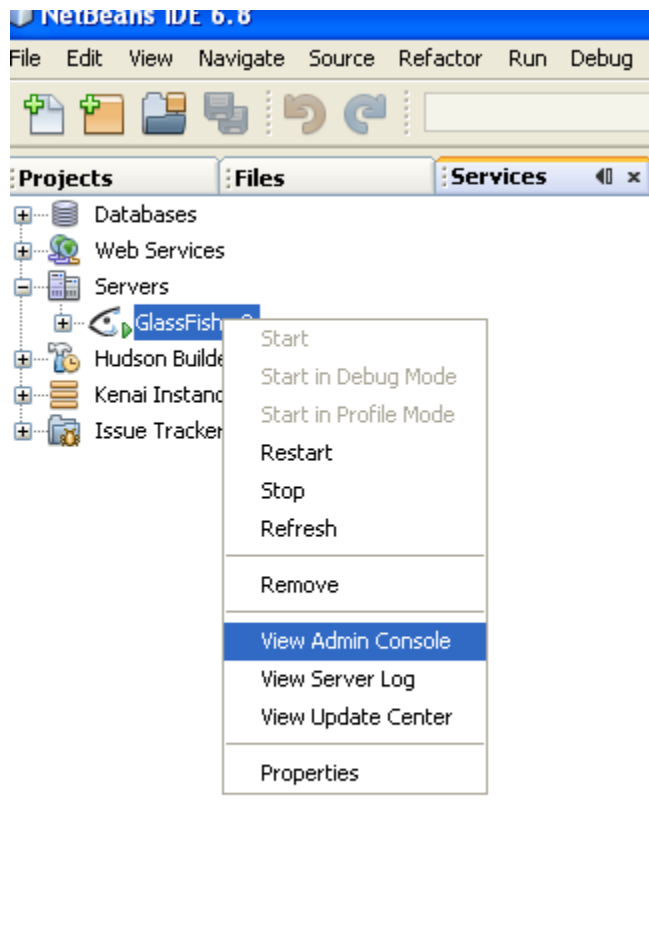
Output

Tasks

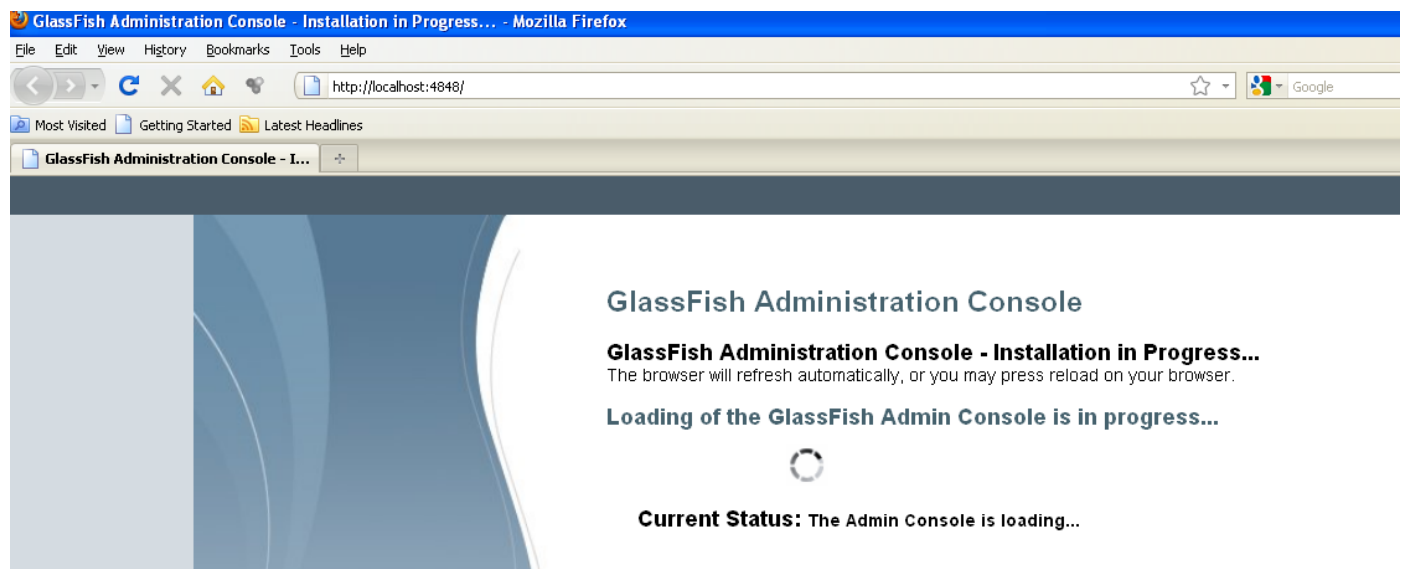
Java DB Database Process x GlassFish v3 x cart (run-deploy) x

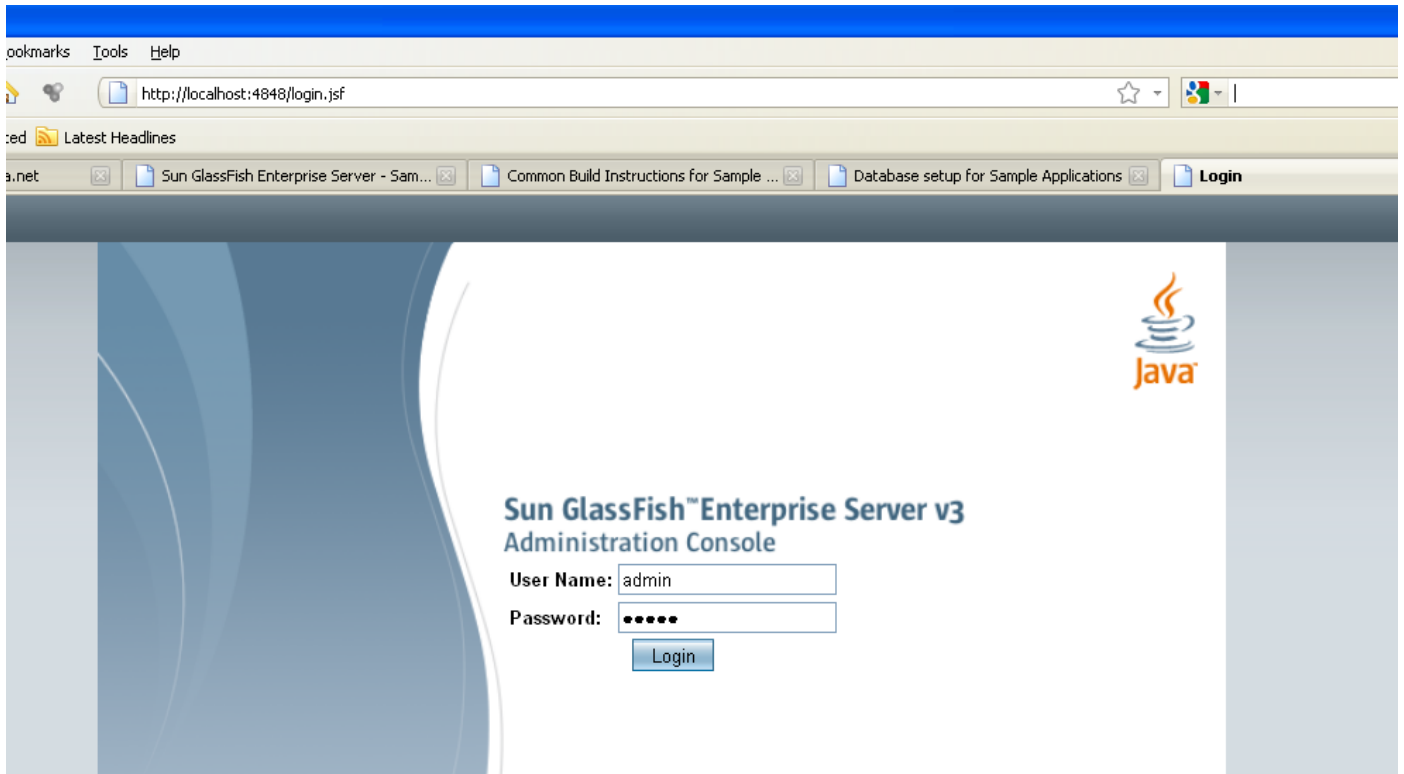
```
cart-ejb.library-inclusion-in-manifest:
cart-ejb.dist-ear:
pre-pre-compile:
pre-compile:
do-compile:
post-compile:
compile:
pre-dist:
post-dist:
dist-directory-deploy:
pre-run-deploy:
Initial deploying cart to D:\Documents and Settings\Geni\Desktop\ADV-OPSYS 2010-2011\MY SLID
Completed initial distribution of cart
post-run-deploy:
run-deploy:
BUILD SUCCESSFUL (total time: 5 seconds)
```

If you want to access the administration console of Glassfish, click as follows:



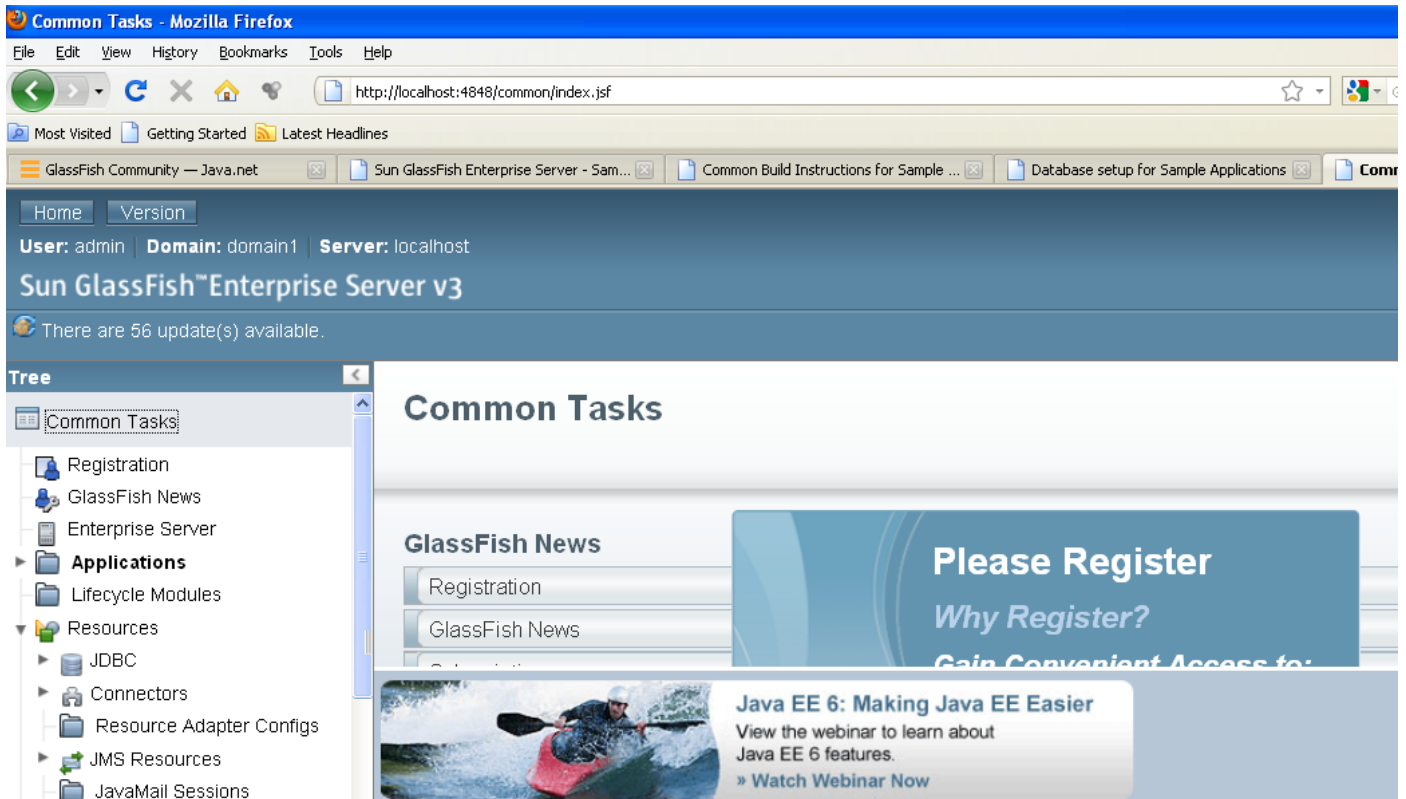
The following window will appear:



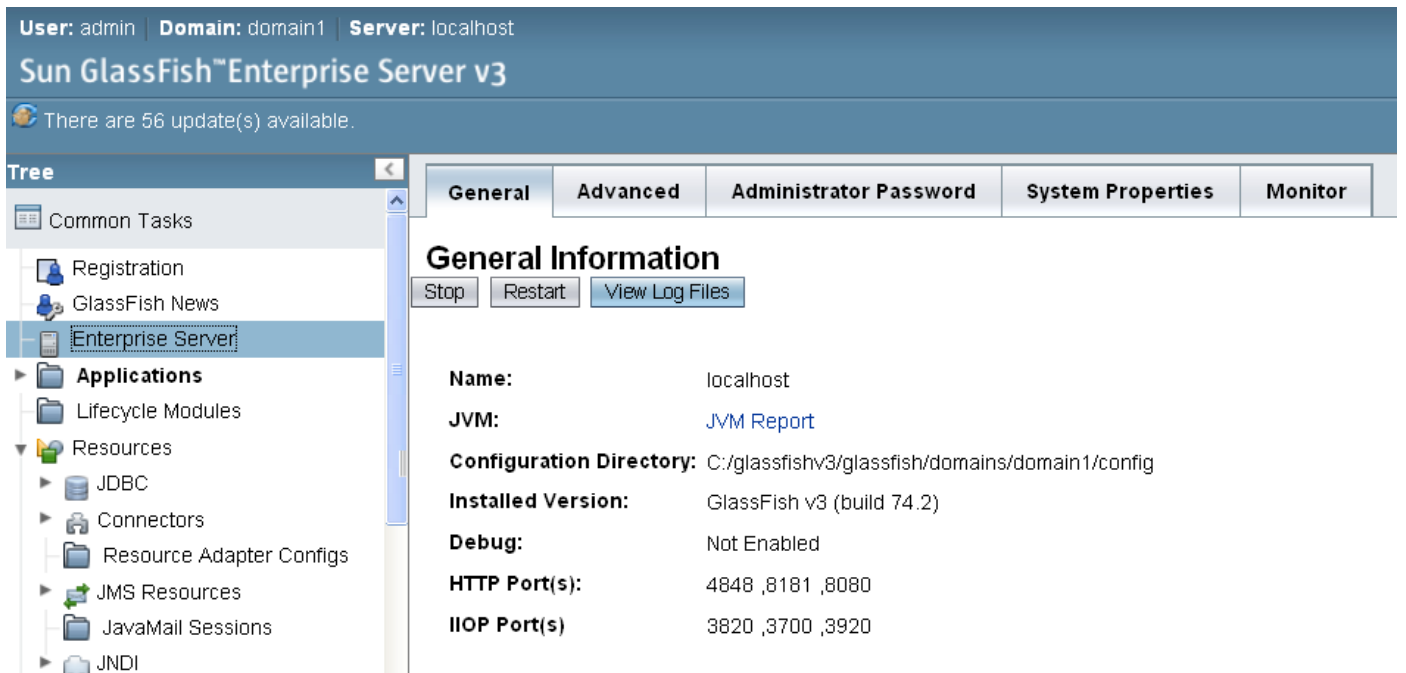


Insert the password of admin and press Login.

You are then forwarded to the console of the administration as follows:



If you want to change the settings of the server click on “Enterprise Server”:



If you want to see already deployed applications click on “Applications” and select the application as follows:

Home | Version | Logout

User: admin | Domain: domain1 | Server: localhost

Sun GlassFish™ Enterprise Server v3

There are 56 update(s) available.

Tree

- Common Tasks
- Registration
- GlassFish News
- Enterprise Server
 - Applications**
 - cart
 - Lifecycle Modules
 - Resources
 - JDBC
 - Connectors
 - Resource Adapter Configs
 - JMS Resources
 - JavaMail Sessions
 - JNDI
 - Configuration

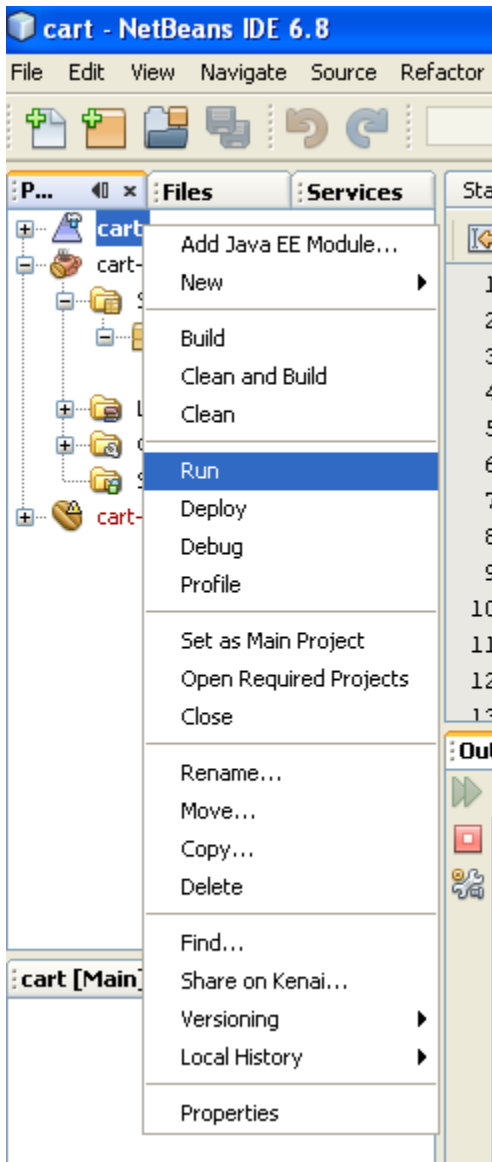
Applications

Applications can be enterprise or web applications, or various kinds of modules.

Deployed Applications (1)

| [Deploy...](#) | [Undeploy](#) | [Enable](#) | [Disable](#) | Filter:

	Name	Enabled	Engines	Action
<input type="checkbox"/>	cart	✓	[appclient, ejb]	Redeploy Restart



The execution will generate the following:

pre-run-deploy:

Undeploying ...

Initial deploying cart to D:\Documents and Settings\Geni\Desktop\ADV-OPSYS 2010-2011\MY SLIDES\LESSON 8\ejb\cart\dist\gfdeplo

Completed initial distribution of cart

Initializing...

post-run-deploy:

run-deploy:

Warning: Could not find file C:\glassfishv3\glassfish\domains\domain1\generated\xml\cart\cartClient.jar to copy.

Copying 1 file to D:\Documents and Settings\Geni\Desktop\ADV-OPSYS 2010-2011\MY SLIDES\LESSON 8\ejb\cart\dist

Copying 4 files to D:\Documents and Settings\Geni\Desktop\ADV-OPSYS 2010-2011\MY SLIDES\LESSON 8\ejb\cart\dist\cartClient

Apr 7, 2011 11:12:19 AM com.sun.enterprise.transaction.JavaEETransactionManagerSimplified initDelegates

INFO: Using com.sun.enterprise.transaction.jts.JavaEETransactionManagerJTSDelegate as the delegate

Retrieving book title from cart: Infinite Jest

Retrieving book title from cart: Bel Canto

Retrieving book title from cart: Kafka on the Shore

Removing "Gravity's Rainbow" from cart.

Caught a BookException: "Gravity's Rainbow" not in cart.

Java Result: 1

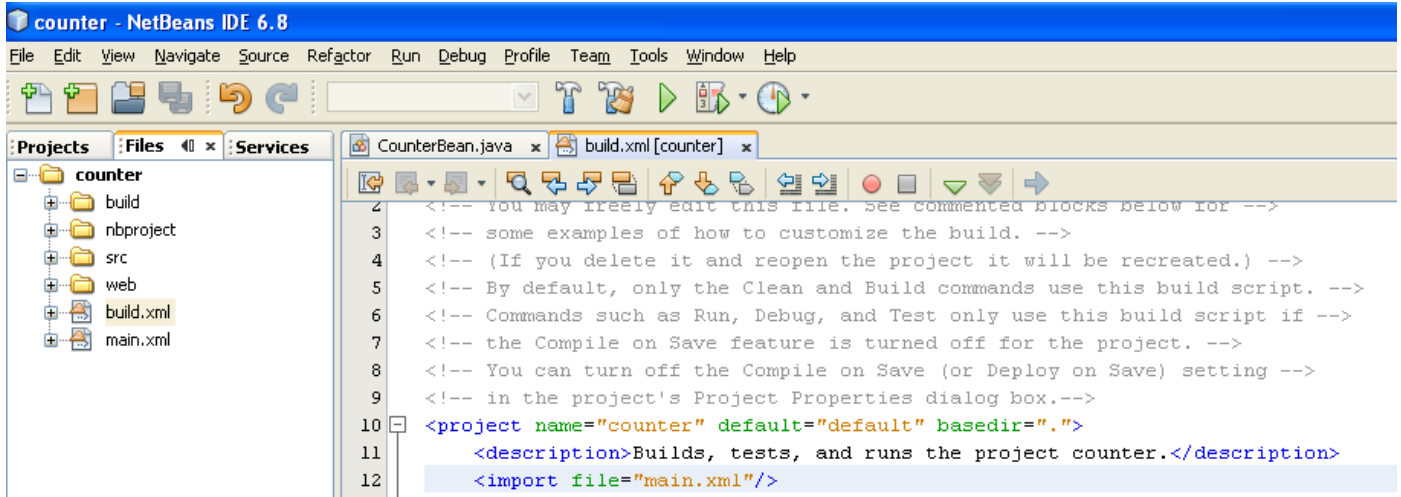
run-cart-app-client:

run:

BUILD SUCCESSFUL (total time: 24 seconds)

Running the Counter Example.

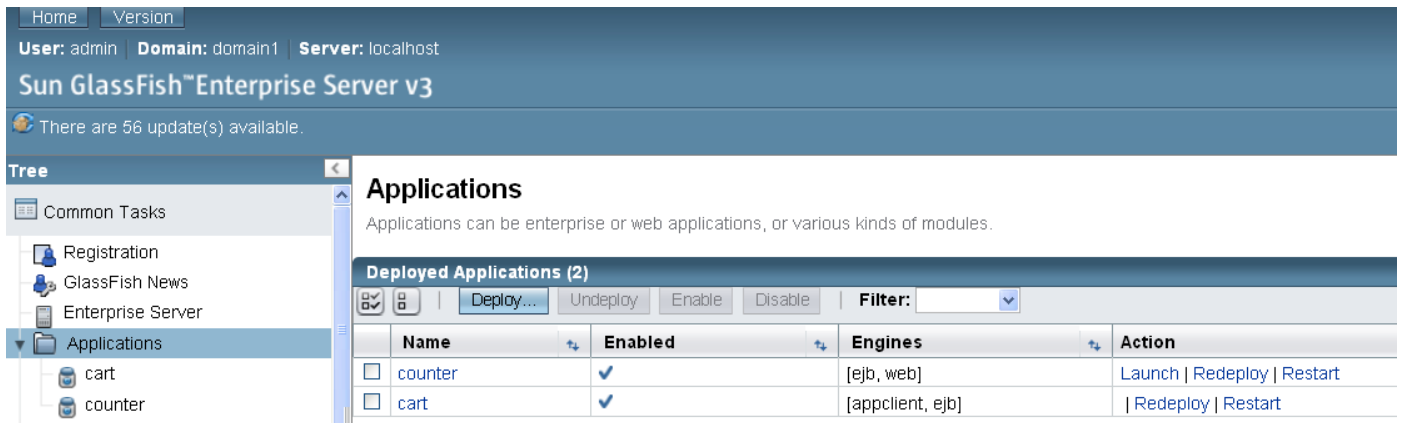
To properly build the application you need to edit the file build.xml:



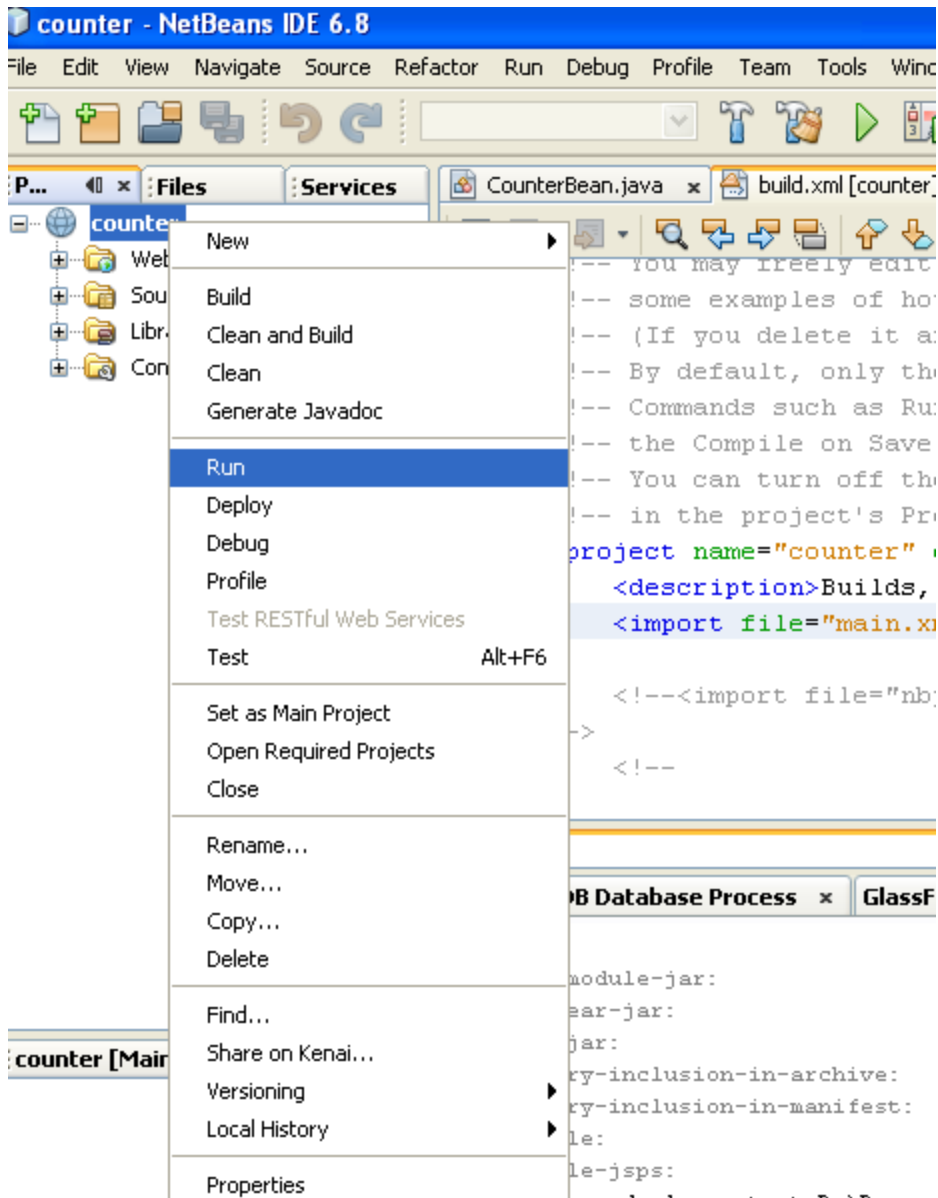
You should add to this file the following line again after copying the file main.xml:

```
<import file="main.xml"/>
```

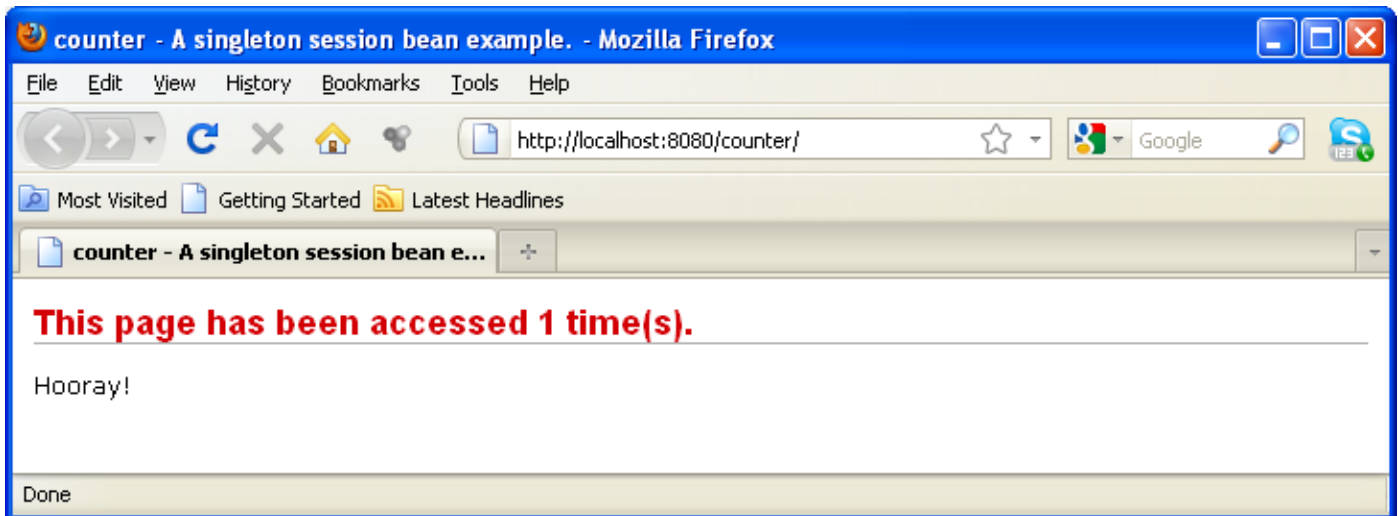
Once you Deploy the application you can check the status in the admin console:



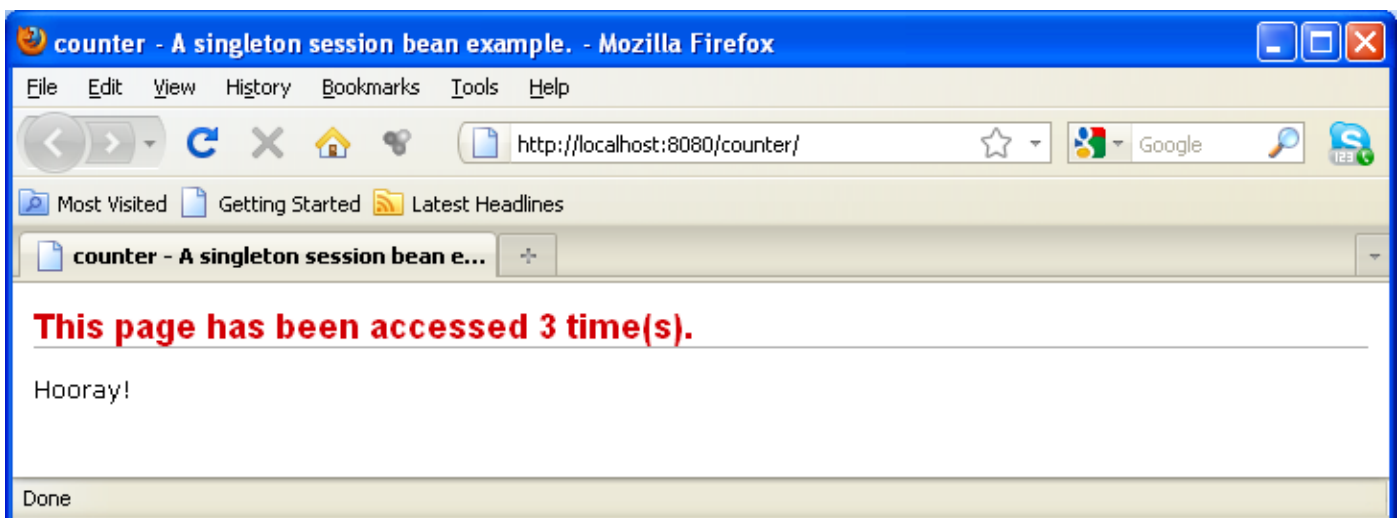
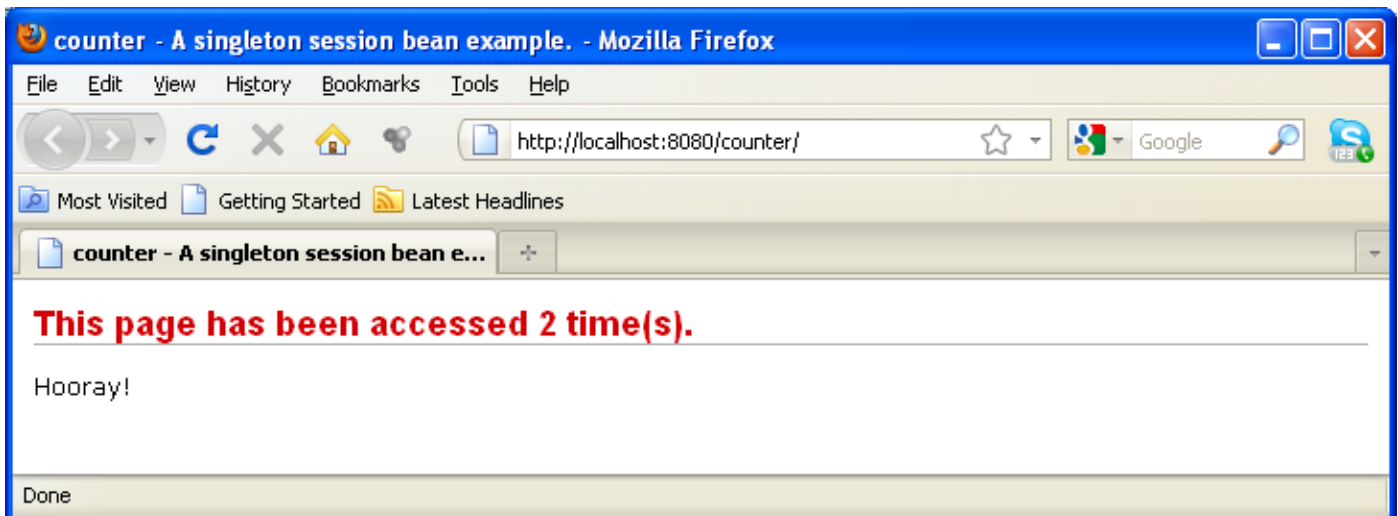
To run the application click on Run as follows:



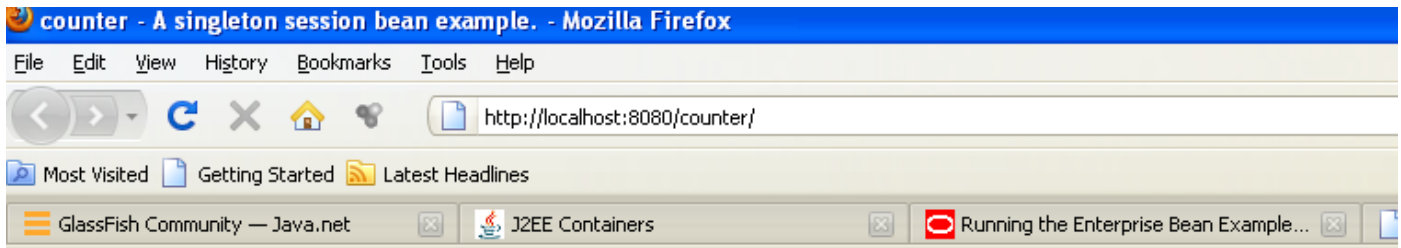
You will see the following:



If you press Refresh the number of access will grow everytime you access the page:



Even if you close the window and access it again with Run, you will see that the application still continues to count the access:




This page has been accessed 4 time(s).

Hooray!

Only if you restart the application, the counter will restart from zero. You can restart the application from the admin console as follows:

Deployed Applications (2)				
Deploy... Undeploy Enable Disable Filter: <input type="text"/>				
	Name	Enabled	Engines	Action
<input type="checkbox"/>	counter	✓	[ejb, web]	Launch Redeploy Restart
<input type="checkbox"/>	cart	✓	[appclient, ejb]	Redeploy Restart

Click Restart and you will see in yellow “The application has been restarted”:

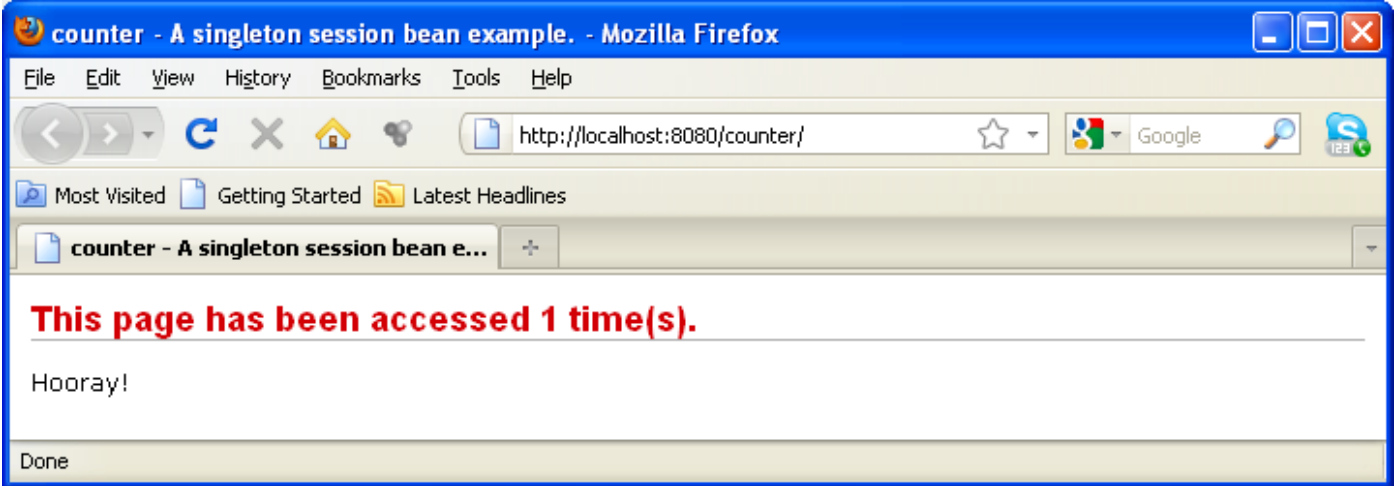
 **The application has been successfully restarted.**

Applications

Applications can be enterprise or web applications, or various kinds of modules.

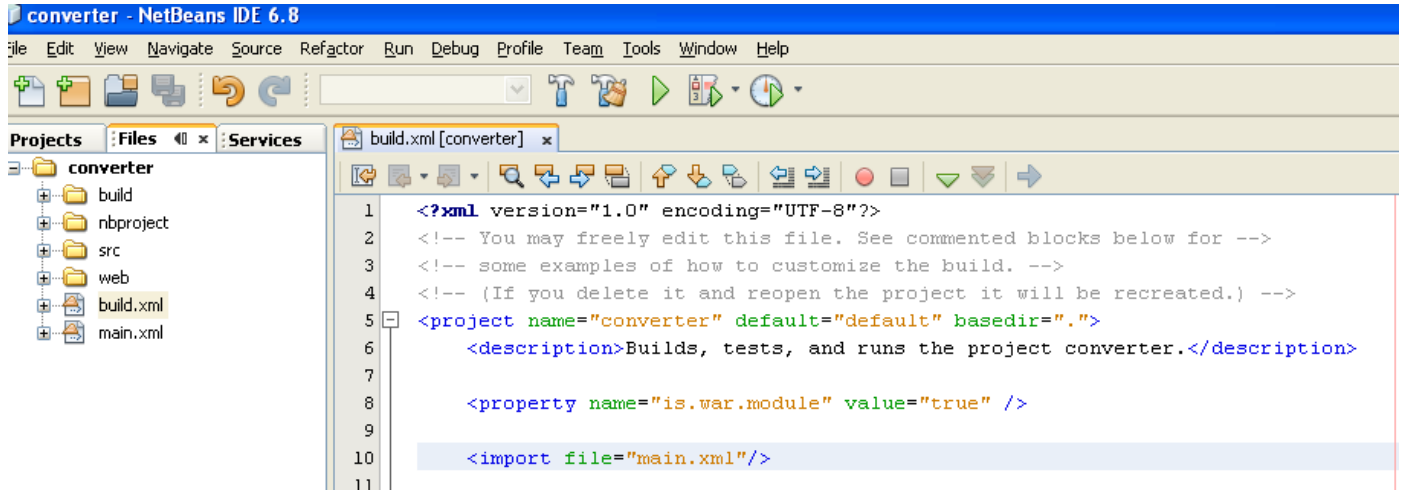
Deployed Applications (2)				
Deploy... Undeploy Enable Disable Filter: <input type="text"/>				
	Name	Enabled	Engines	Action
<input type="checkbox"/>	counter	✓	[ejb, web]	Launch Redeploy Restart
<input type="checkbox"/>	cart	✓	[appclient, ejb]	Redeploy Restart

If you access the page again, by doing “Run” under Netbeans or “Launch” in the above window you will see:



Running the Converter:

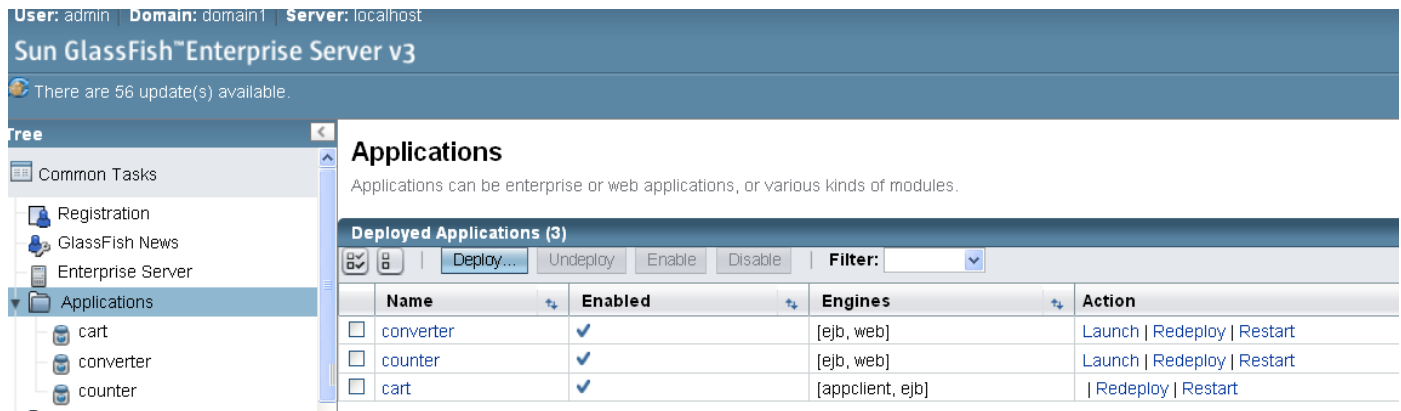
To properly build the application you need to edit the file build.xml:



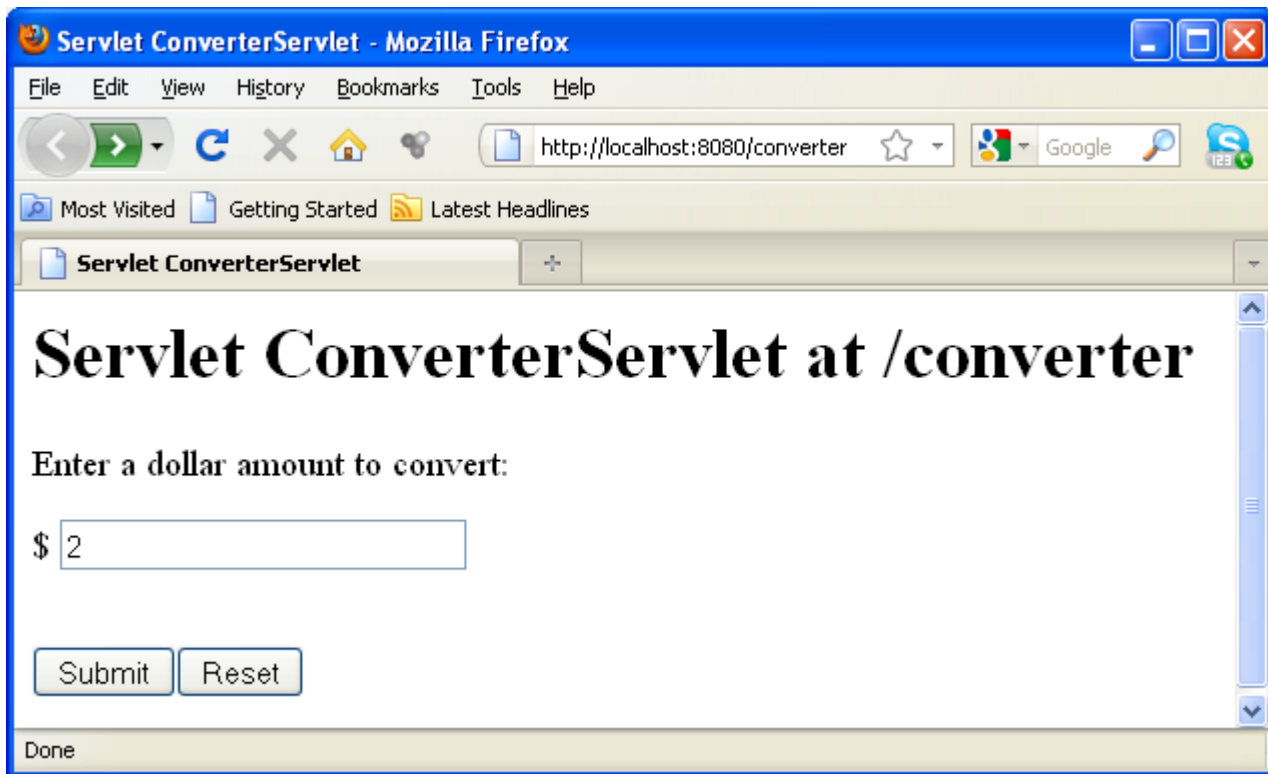
You should add to this file the following line again after copying the file main.xml:

```
<import file="main.xml"/>
```

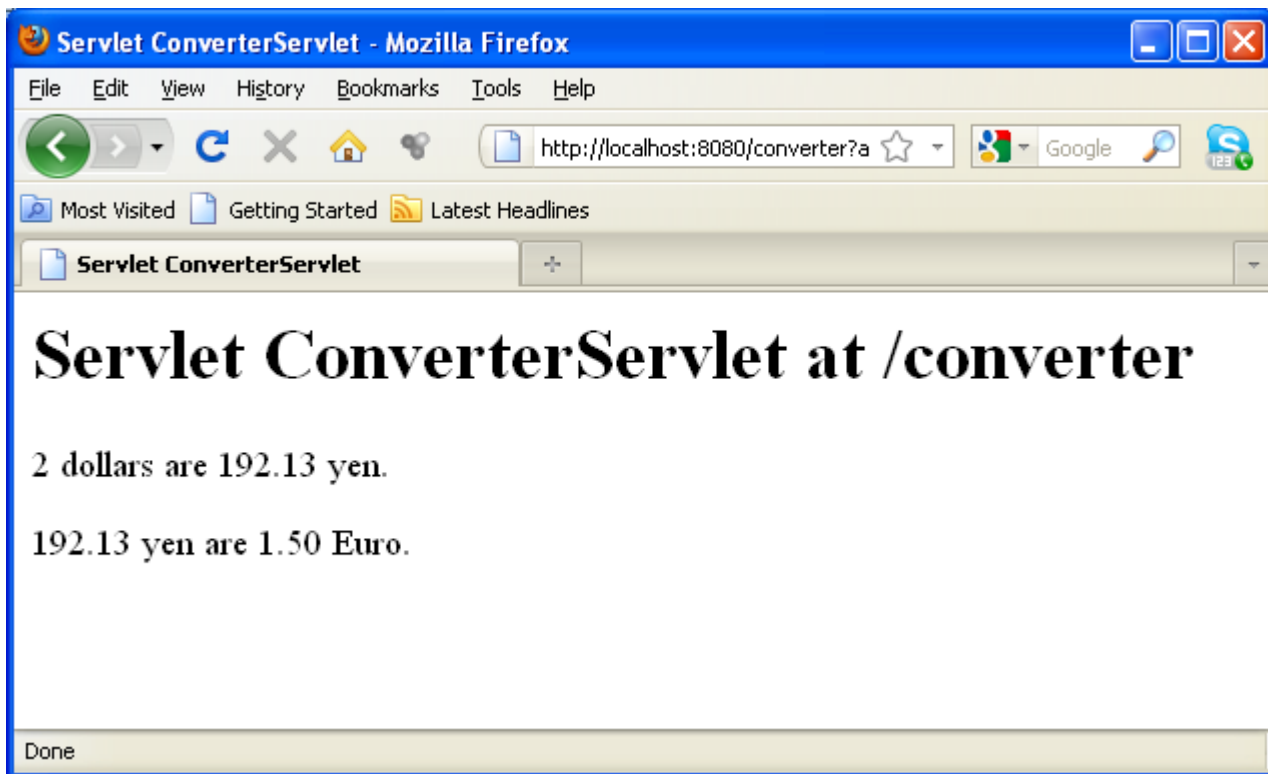
Once you Deploy the application you can check the status in the admin console:



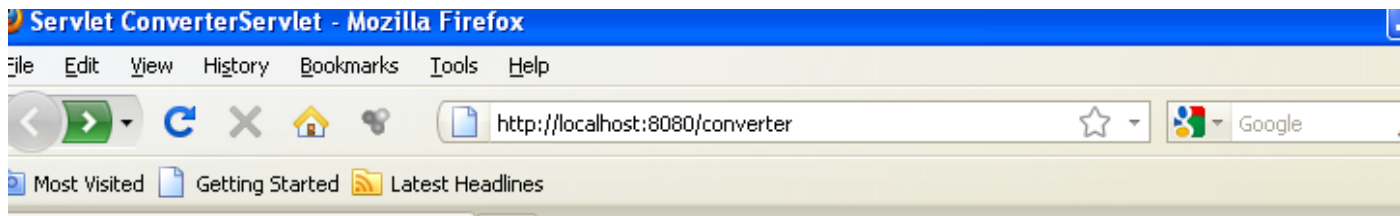
You can launch the application either by doing “Run” under Netbeans or “Launch” in the above window and you will see the following:



If you enter an amount and press “Submit” you will get the following:

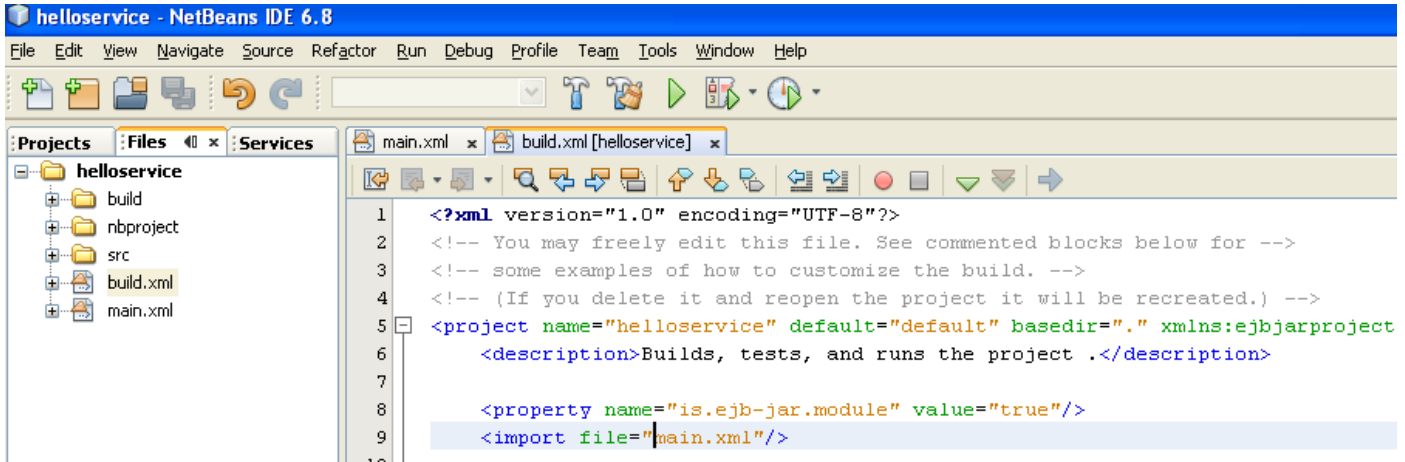


The application now is alive at this address:



Running the HelloService example:

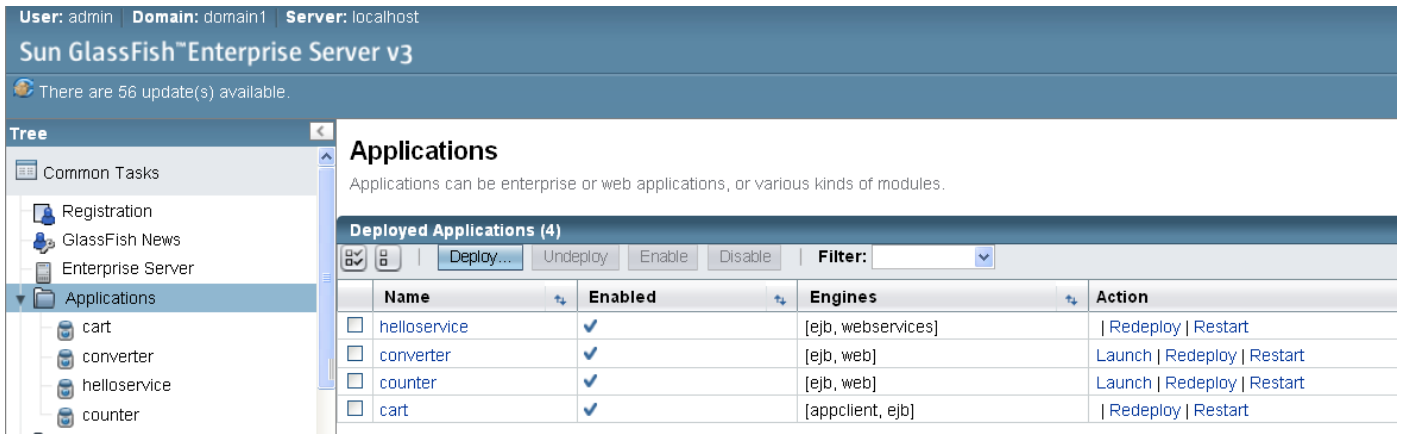
To properly build the application you need to edit the file build.xml:



You should add to this file the following line again after copying the file main.xml:

```
<import file="main.xml"/>
```

Once you Deploy the application you can check the status in the admin console:



To Test the Service without a Client

The GlassFish Server Administration Console allows you to test the methods of a web service endpoint. To test the `sayHello` method of `HelloServiceBean`, follow these steps.

1. **Open the Administration Console by opening the following URL in a web browser:**

http://localhost:4848/

2. In the left pane of the Administration Console, select the Applications node.
3. In the Applications table, click `helloservice`.

The screenshot shows the 'Edit Application' page in the Administration Console. The left sidebar shows a tree view with 'Applications' expanded and 'helloservice' selected. The main content area has the following details:

- Name:** helloservice
- Status:** Enabled
- Description:** [Empty text box]
- Location:** file:/D:/Documents%20and%20Settings/Geni/Desktop/ADV-OPSYS%202010-2011/MY%20SLIDES/LESSON%208/ejb/helloservice/build/jar/
- Libraries:** [Empty list]

Below the details is a table titled 'Modules and Components (2)':

Module Name	Engines	Component Name	Type	Action
helloservice	[ejb, webservices]	-----	-----	
helloservice		HelloServiceBean	StatelessSessionBean	View Endpoint

4. In the Modules and Components table, click View Endpoint.

The screenshot shows the 'Web Service Endpoint Information' page. The left sidebar is the same as in the previous screenshot. The main content area displays the following information:

- Application Name:** [helloservice](#)
- Tester:** [/HelloServiceBeanService/HelloServiceBean?Tester](#)
- WSDL:** [/HelloServiceBeanService/HelloServiceBean?wsdl](#)
- Endpoint Name:** HelloServiceBean
- Service Name:** [http://helloservice.ejb.javaee.tutorial.sun.com/](#)
- Port Name:** HelloServiceBeanPort
- Deployment Type:** 109
- Implementation Type:** EJB
- Implementation Class Name:** com.sun.tutorial.javaee.ejb.helloservice>HelloServiceBean
- Endpoint Address URI:** [/HelloServiceBeanService/HelloServiceBean](#)
- Namespace:** [com.sun.tutorial.javaee.ejb.helloservice>HelloServiceBean](#)
- Description:**

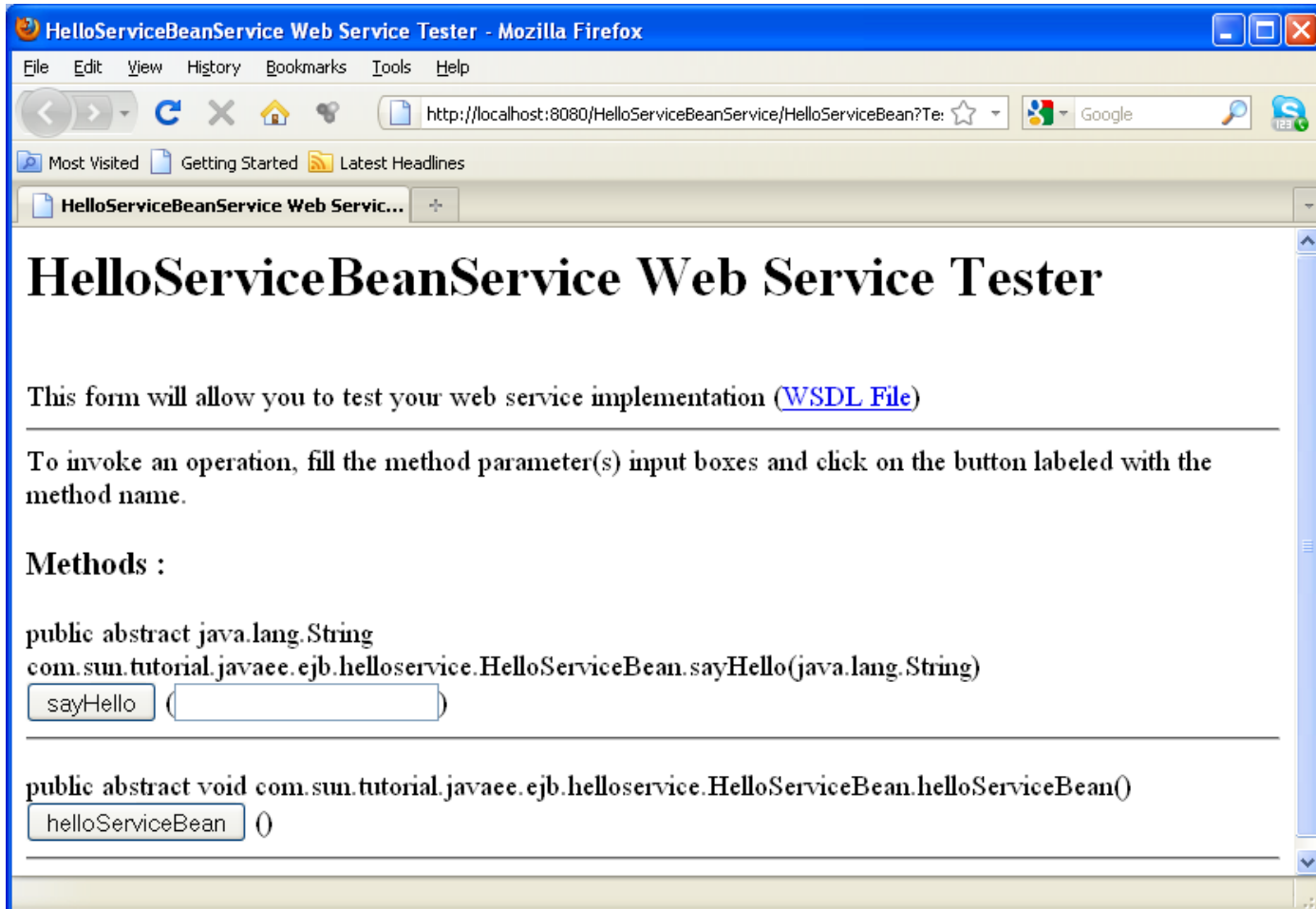
5. On the Web Service Endpoint Information page, click the Tester link:

[/HelloServiceBeanService/HelloServiceBean?Tester](#)

A Web Service Test Links page opens.

6. **On the Web Service Test Links page, click the non-secure link (the one that specifies port 8080).**

A HelloServiceBeanService Web Service Tester page opens.



7. **Under Methods, type a name as the parameter to the `sayHello` method.**



8. Click the `sayHello` button.

The `sayHello` Method invocation page opens. Under Method returned, you'll see the response from the endpoint.

Method invocation trace - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:8080/HelloServiceBeanService/HelloServiceBean?Te

Most Visited Getting Started Latest Headlines

Method invocation trace

sayHello Method invocation

Method parameter(s)

Type	Value
java.lang.String	MIREMENGJES

Method returned

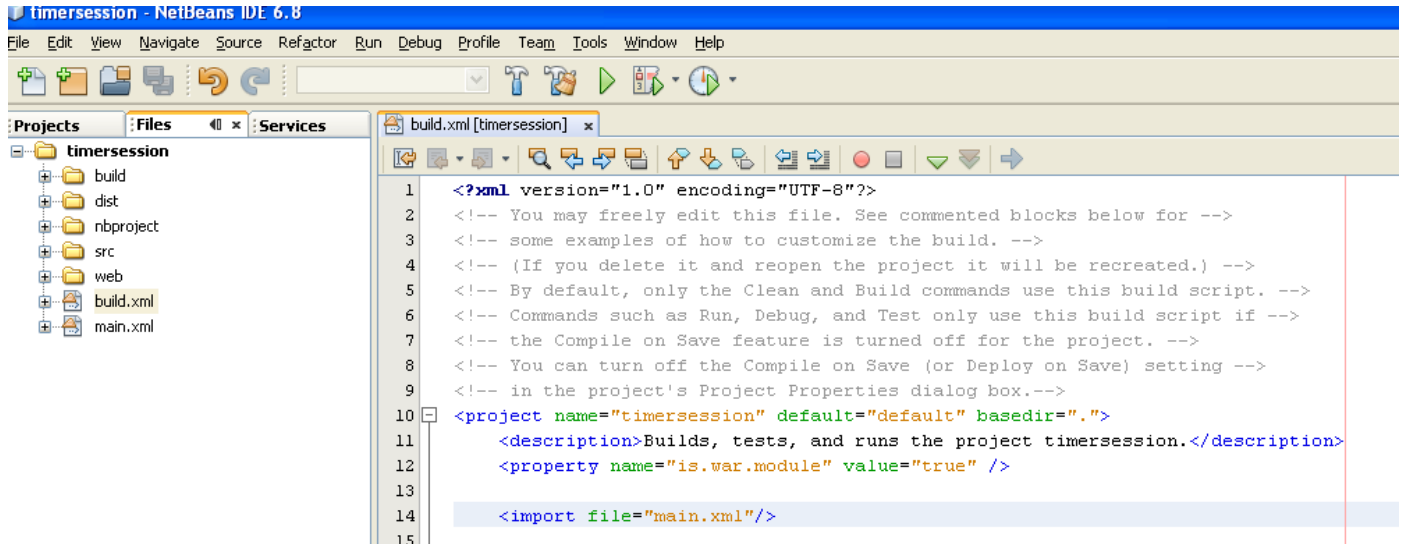
java.lang.String : "Hello, MIREMENGJES."

SOAP Request

Done

Running the timer application:

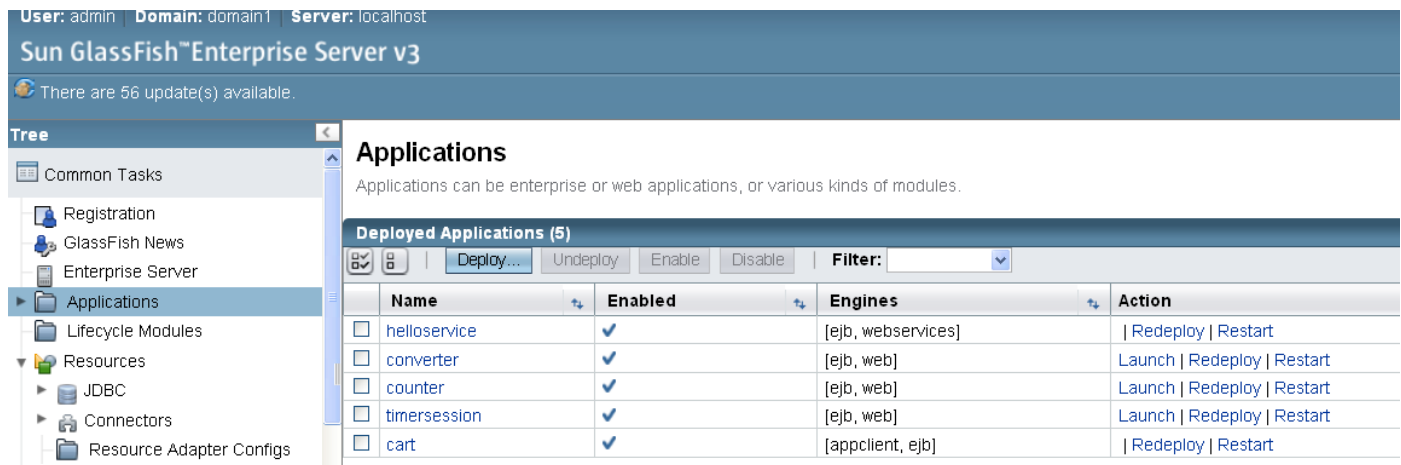
To properly build the application you need to edit the file build.xml:



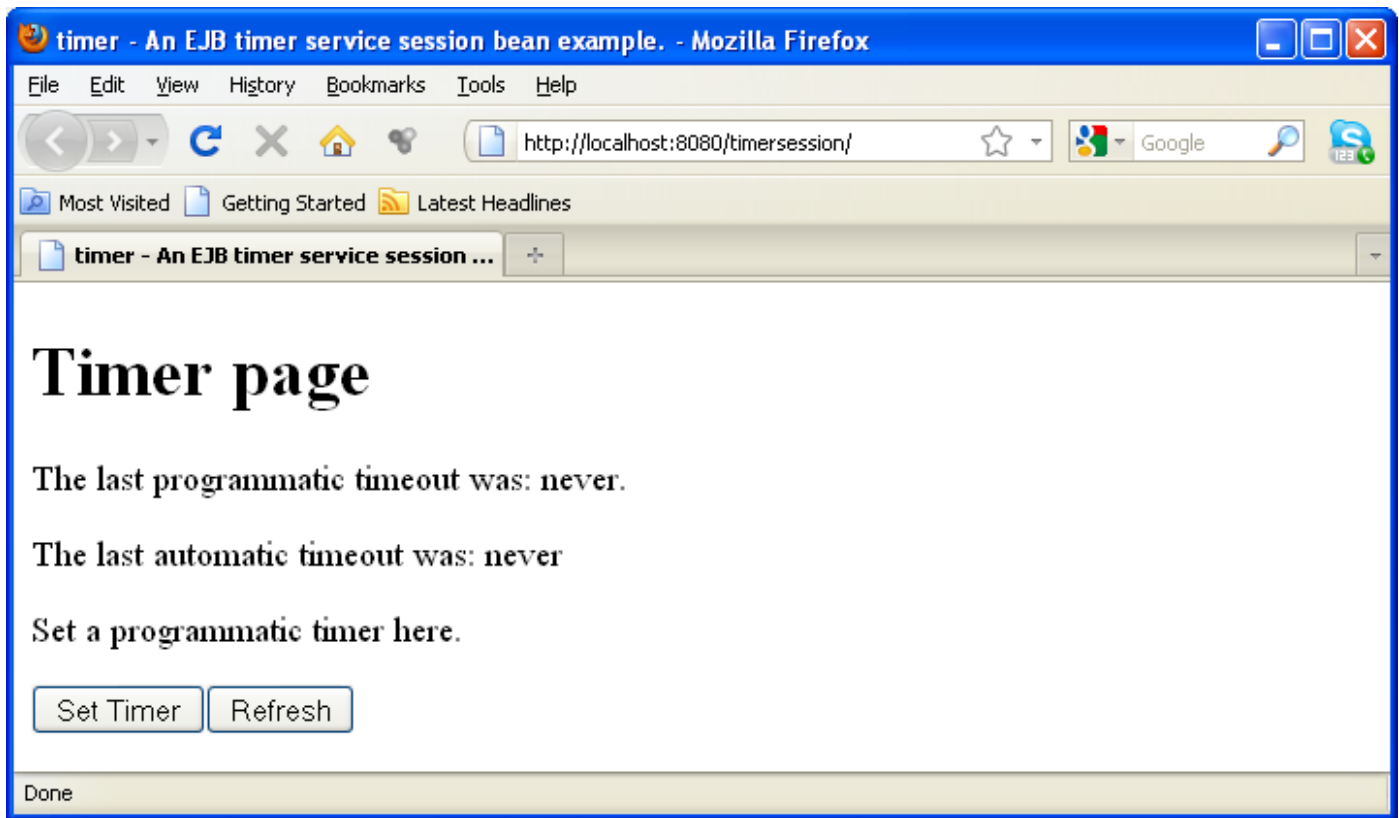
You should add to this file the following line again after copying the file main.xml:

```
<import file="main.xml"/>
```

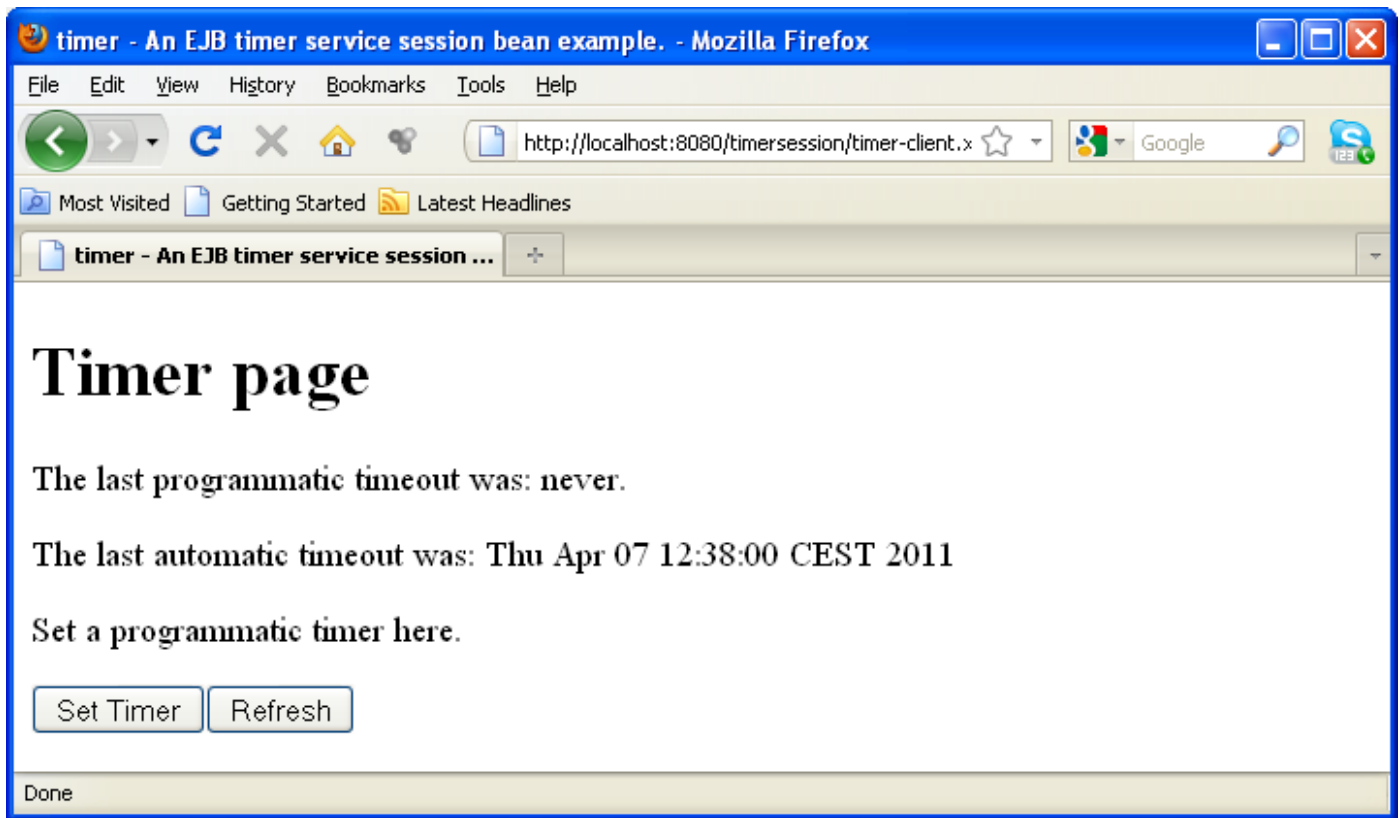
Once you Deploy the application you can check the status in the admin console:



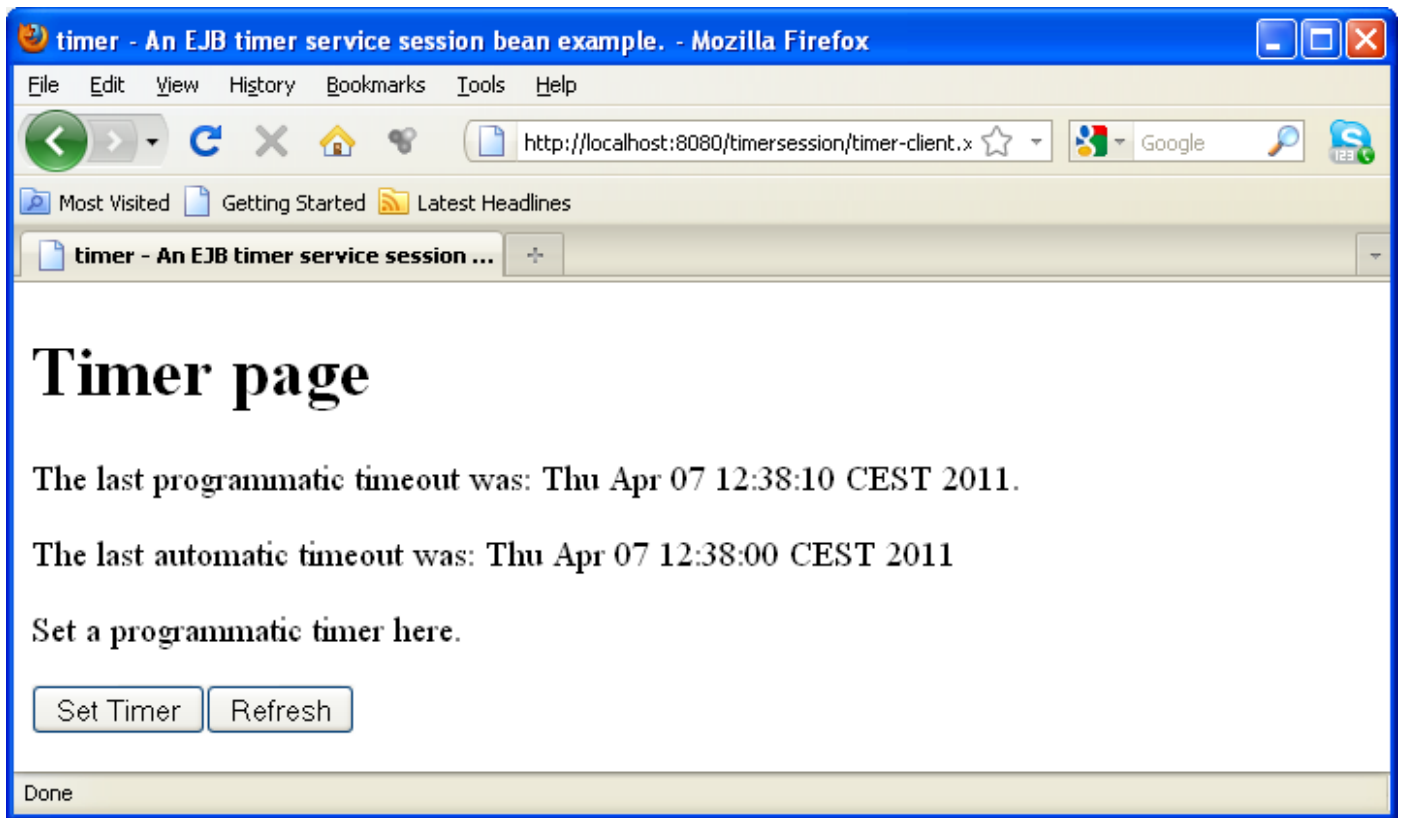
You can launch the application either by doing "Run" under Netbeans or "Launch" in the above window and you will see the following:



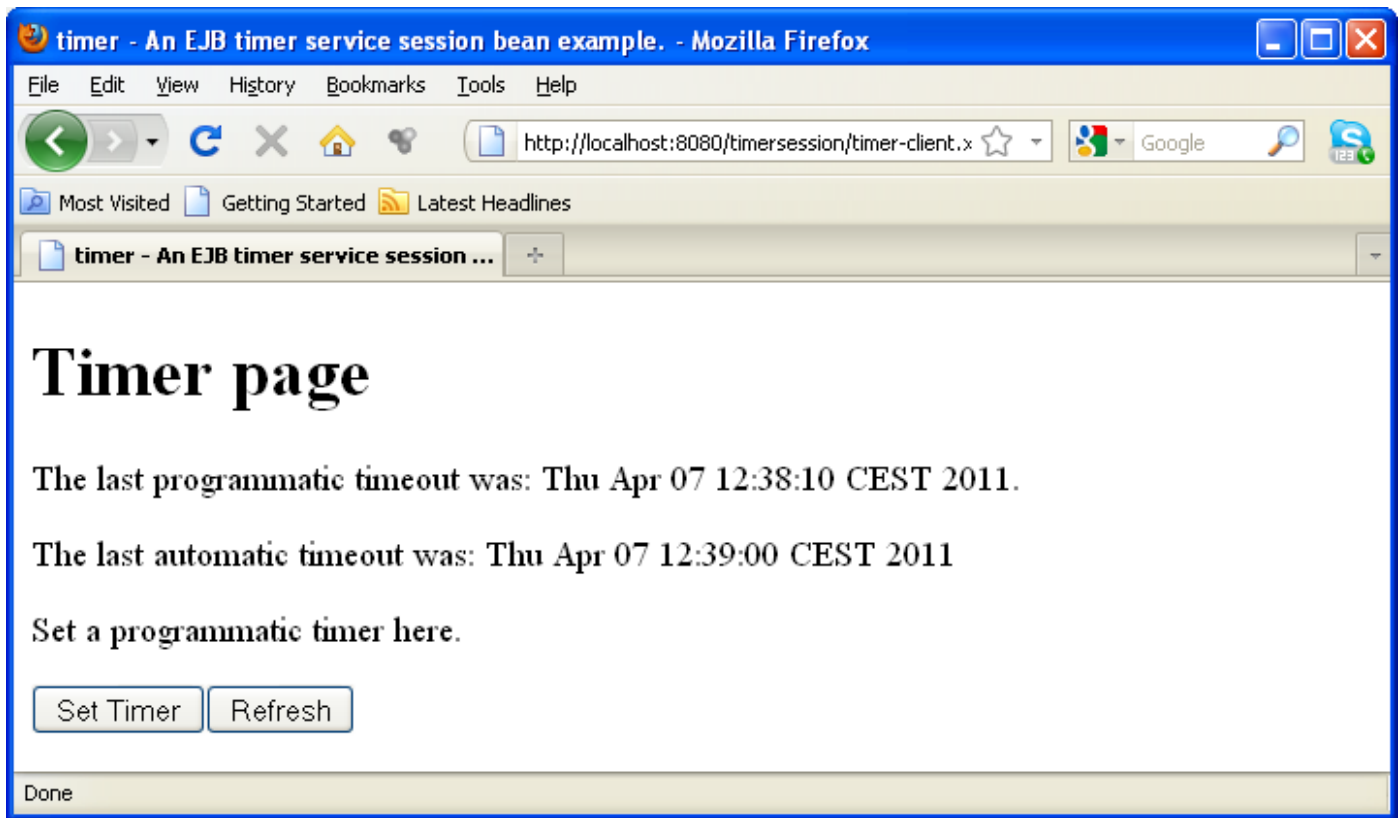
Press button Set Timer to set timer:



Press Refresh:



Press Refresh again:

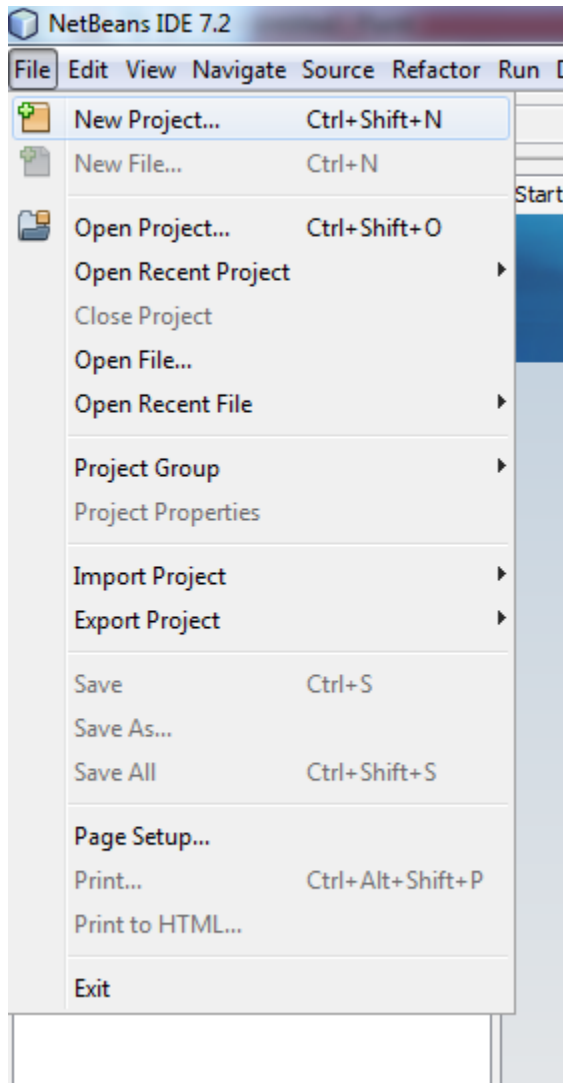


Automatic timeout has changed.

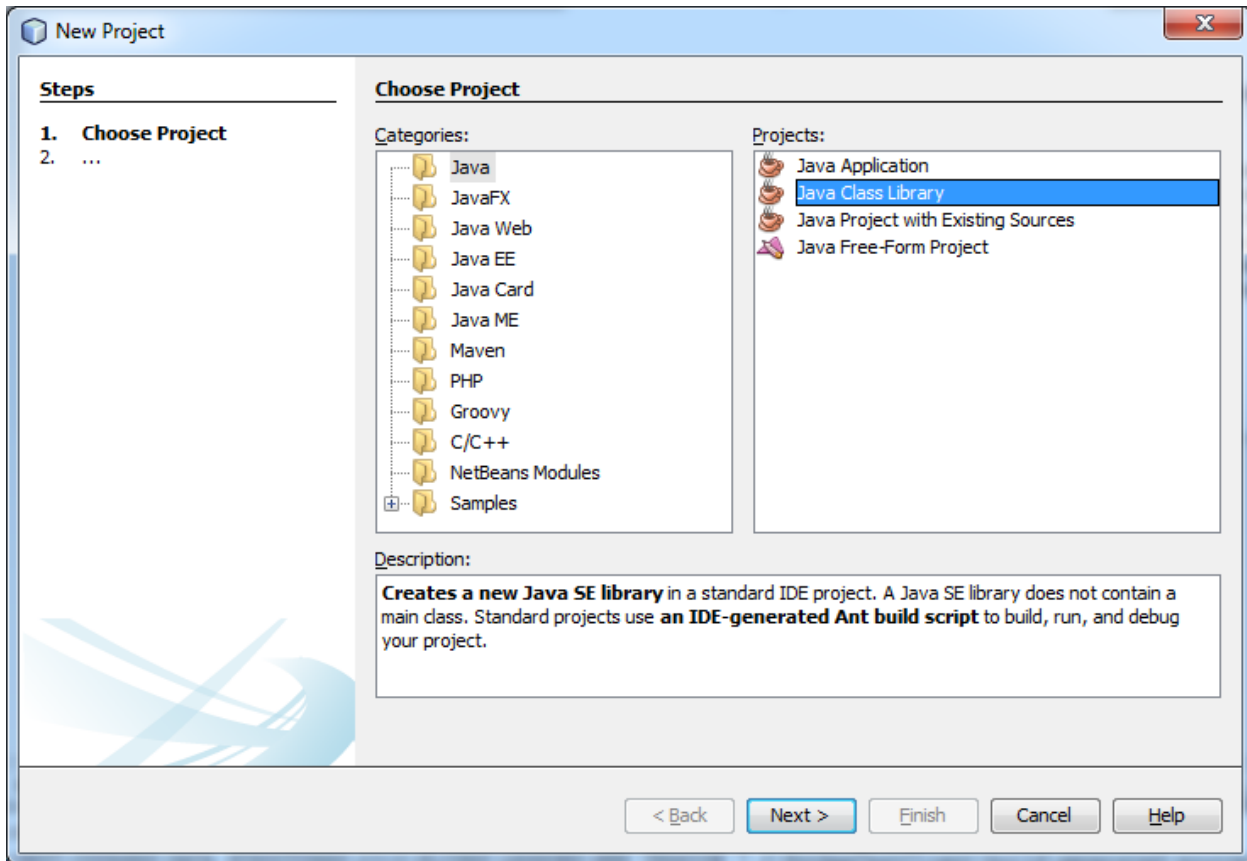
3. Developing a simple EJB application for bank managing

In order to create an EE Application perform the following steps in NetBeans:

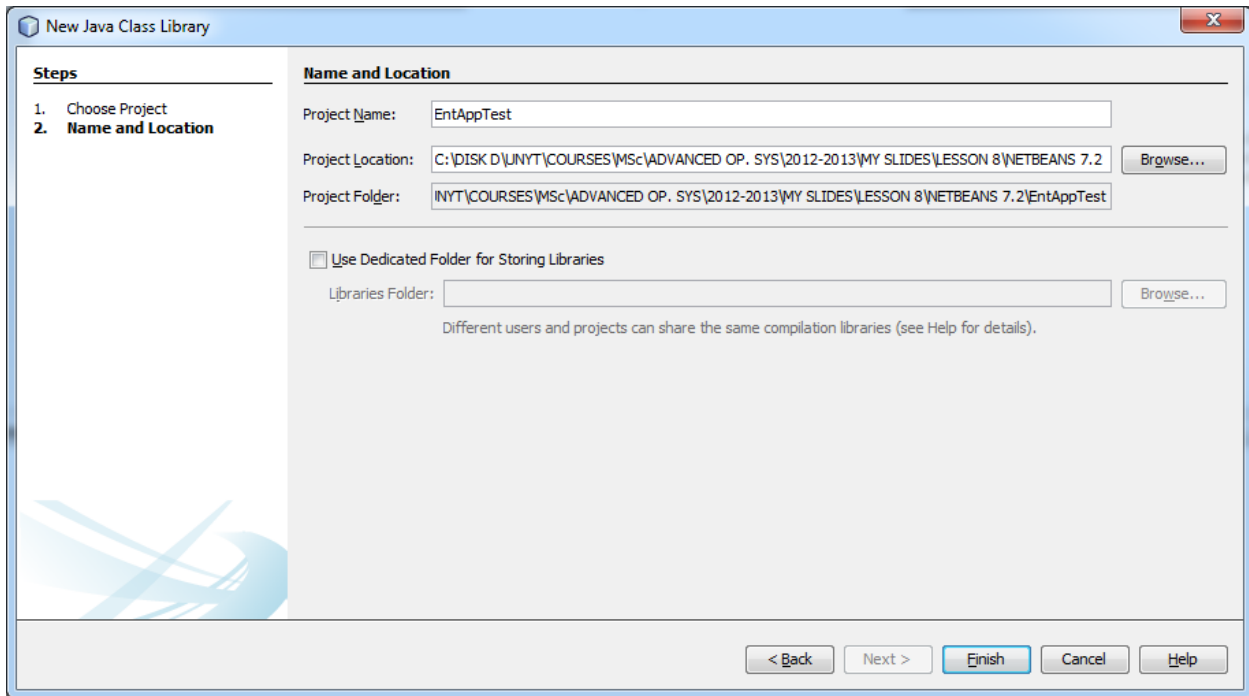
Go to new in Netbeans:



Select Java Class Library as follows:



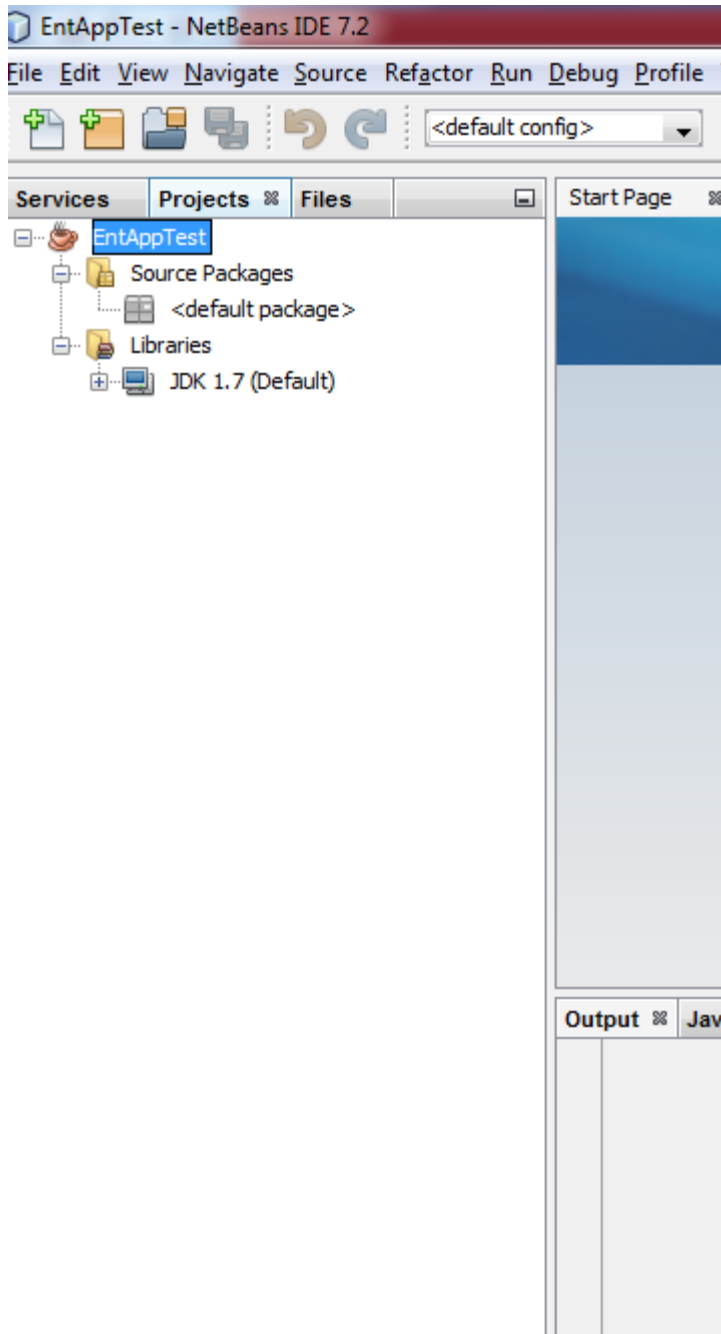
Select Java Class Library and then click Next:



Choose path of the project:

Click Finish.

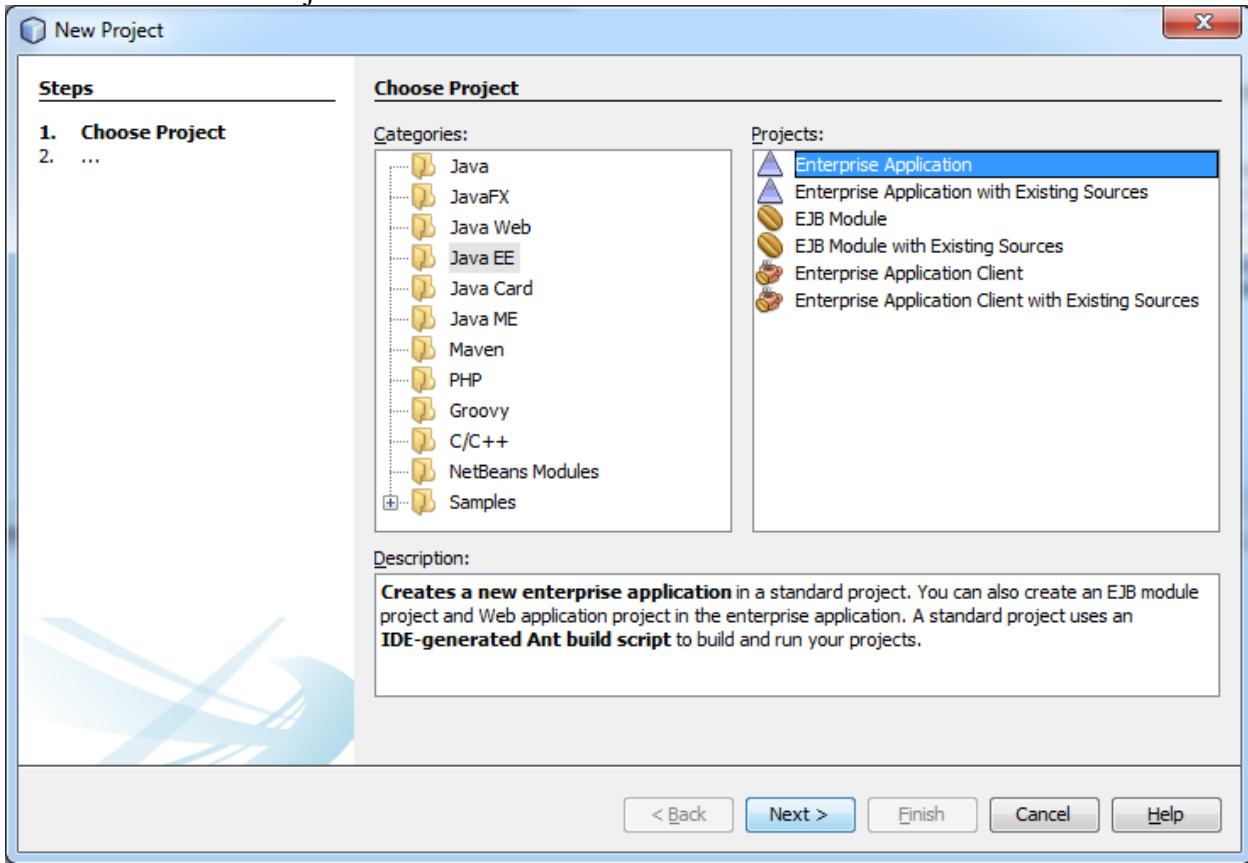
You will see the following window:



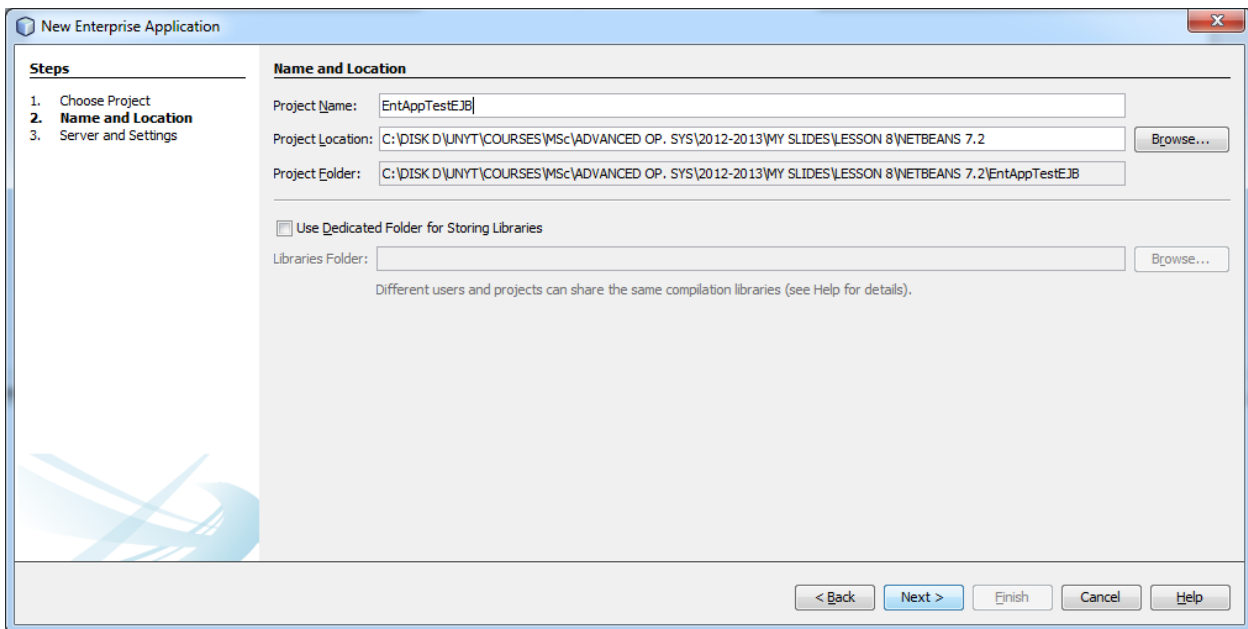
Creating the Enterprise Application

Now you need to create an enterprise application and an EJB module. When you create an EJB, the EJB should be created as part of an enterprise application and packaged as an EAR archive and deployed in to the server.

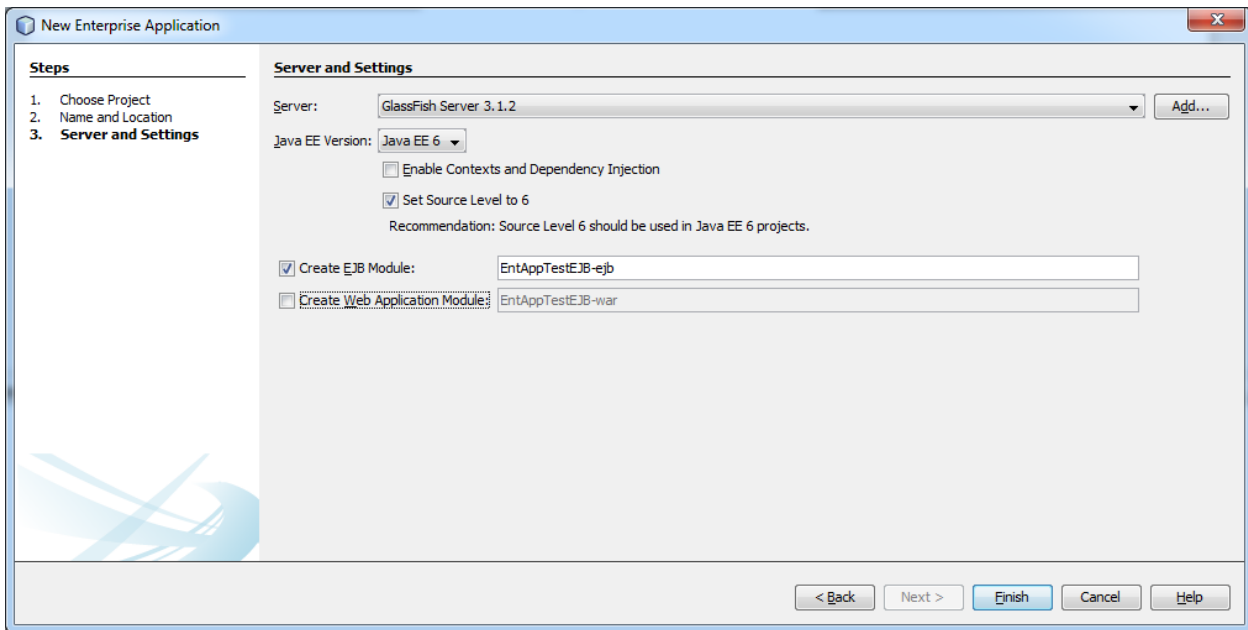
Choose File > New Project



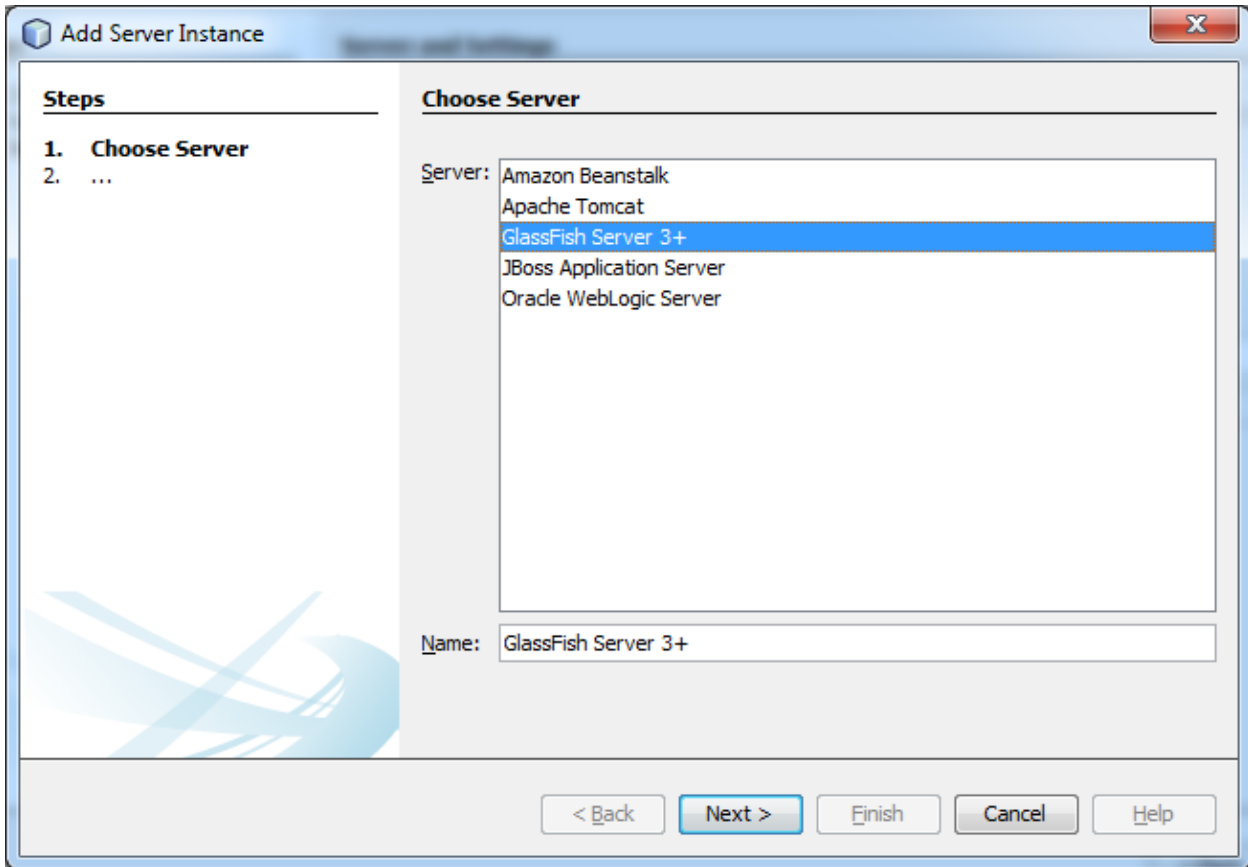
Select Enterprise Application in the Java EE category. Click Next.



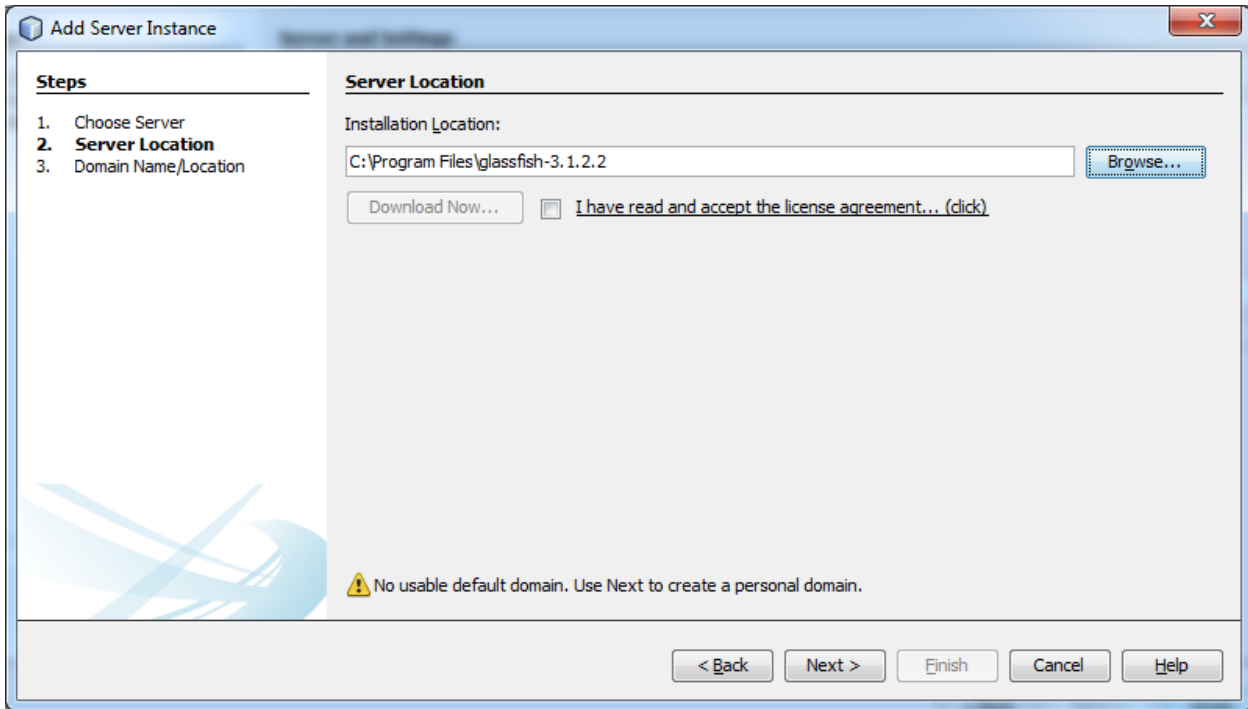
Then do the following:



If you do not have the Glassfish server selected you may do the following. Click Add...

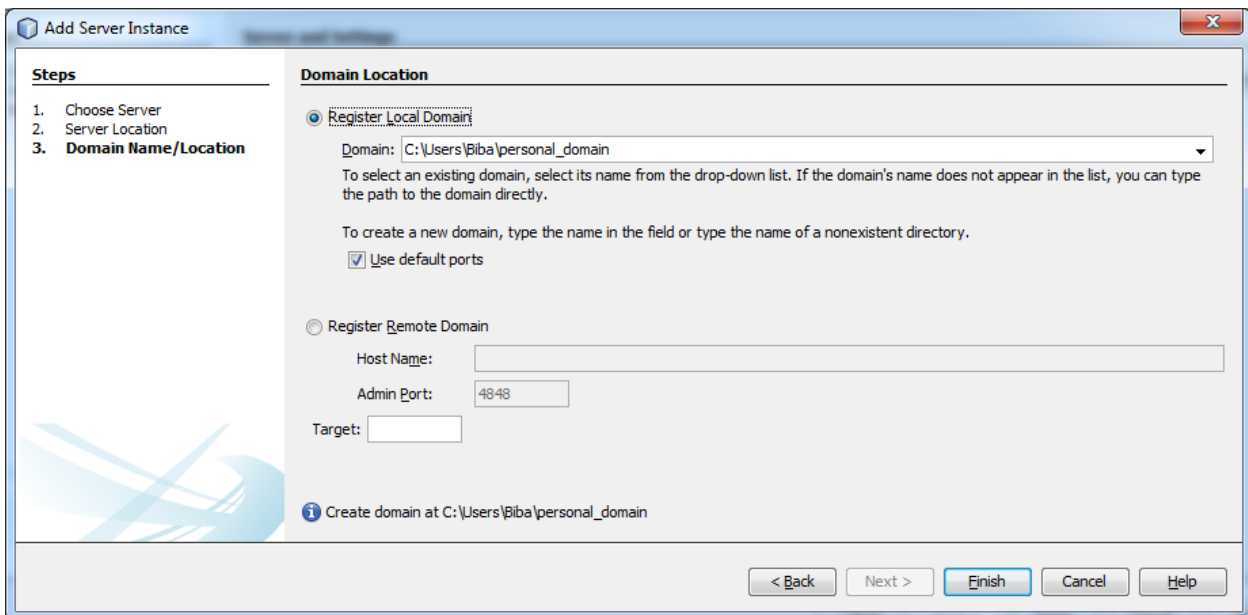


Click Next.

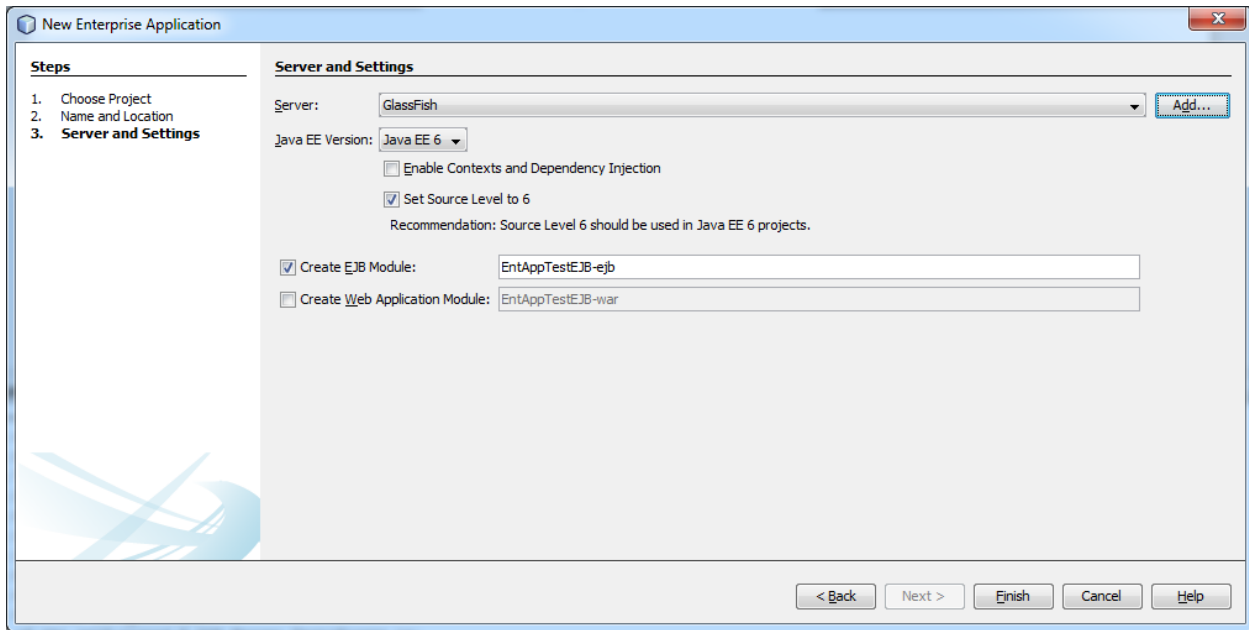


Choose in Browse the directory of Glassfish.

Click Next.

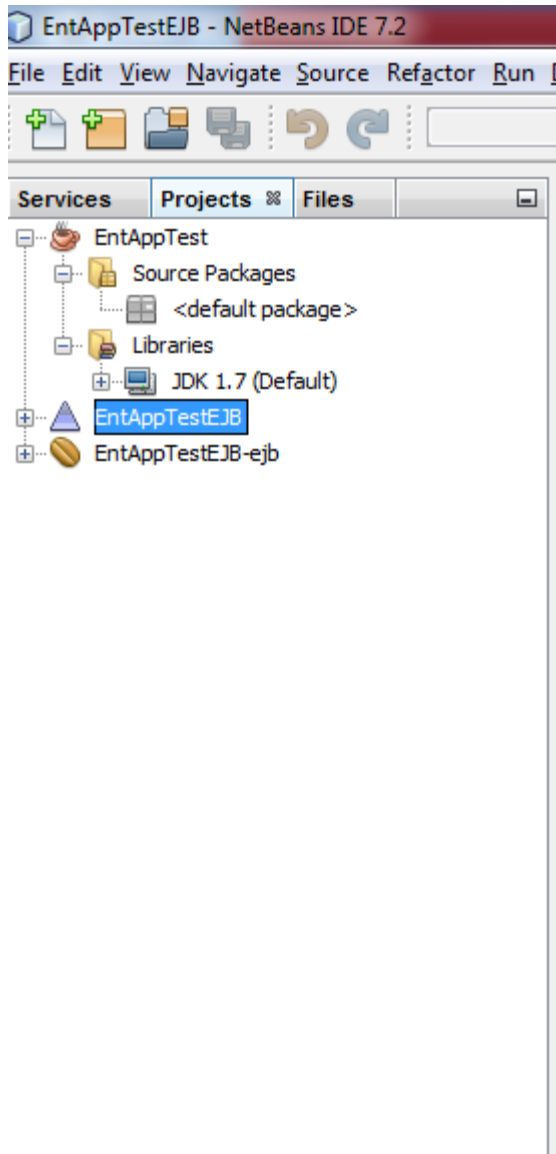


Click Finish. The following window will appear:



Click Finish. Netbeans starts creating the project. This operation may take a while depending on the resources of the machine.

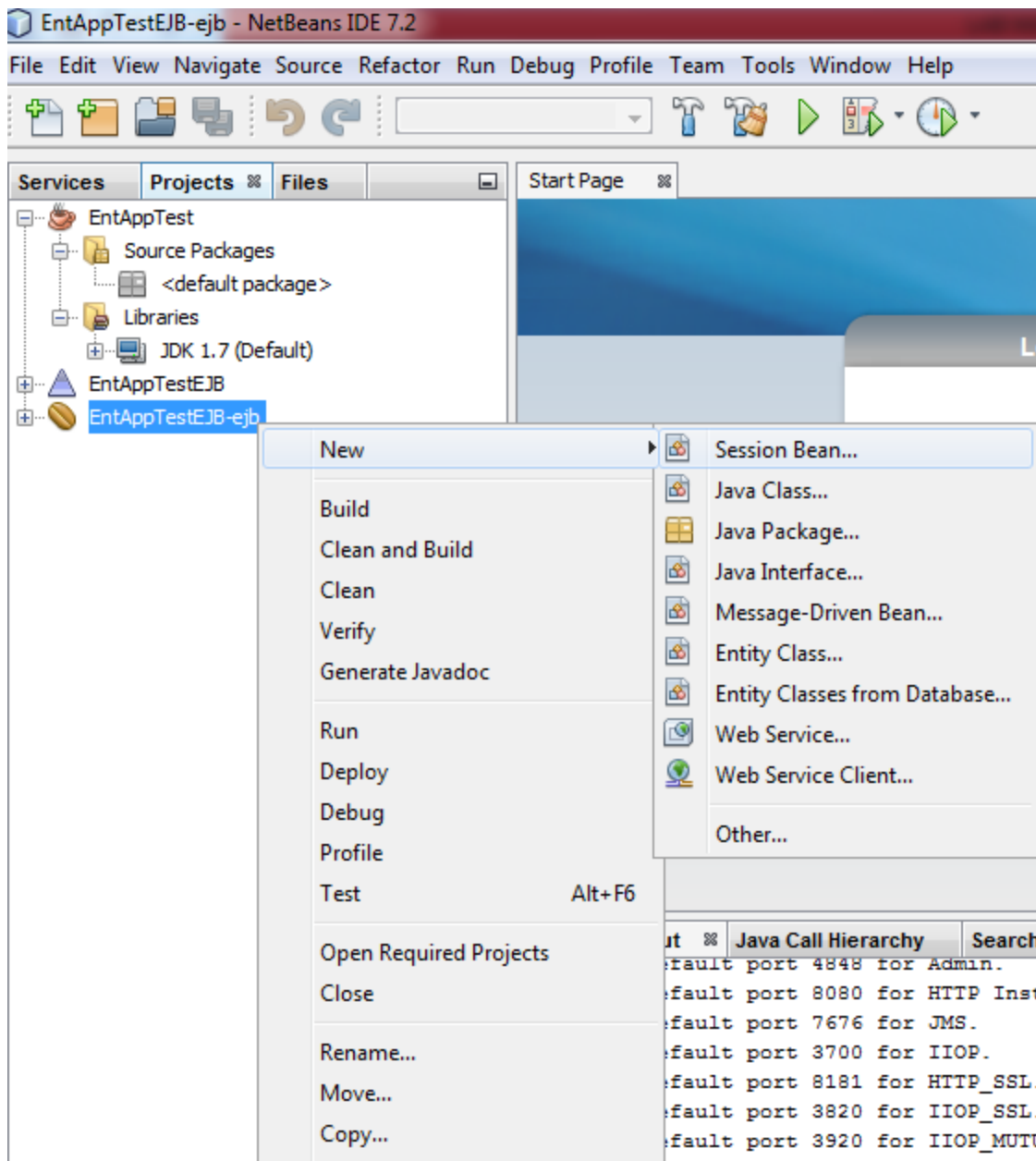
Once the project is correctly installed you will see the following under projects:



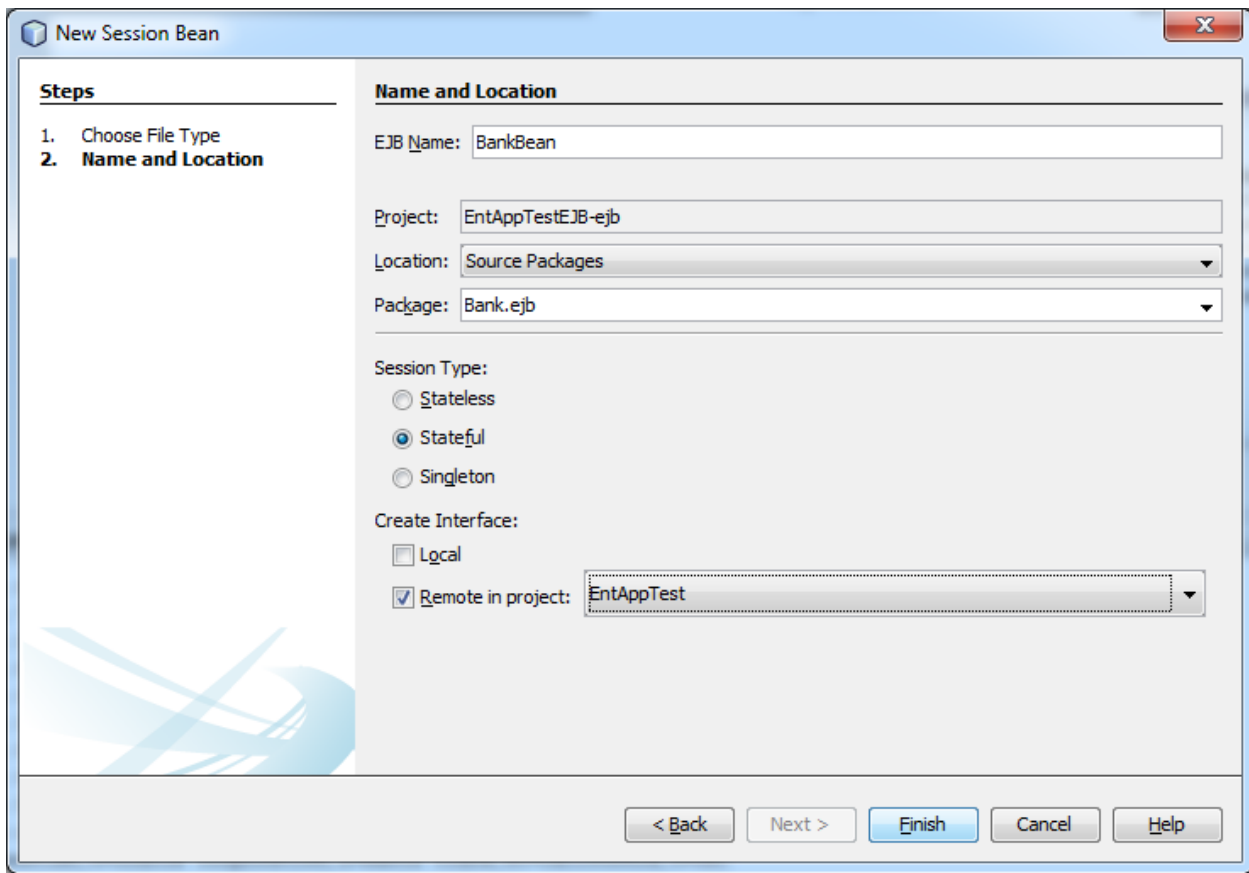
Creating the Session Bean

We will create now a session bean in the EJB module project. We will also create a remote interface for the session bean in the Class Library project. Do the following:

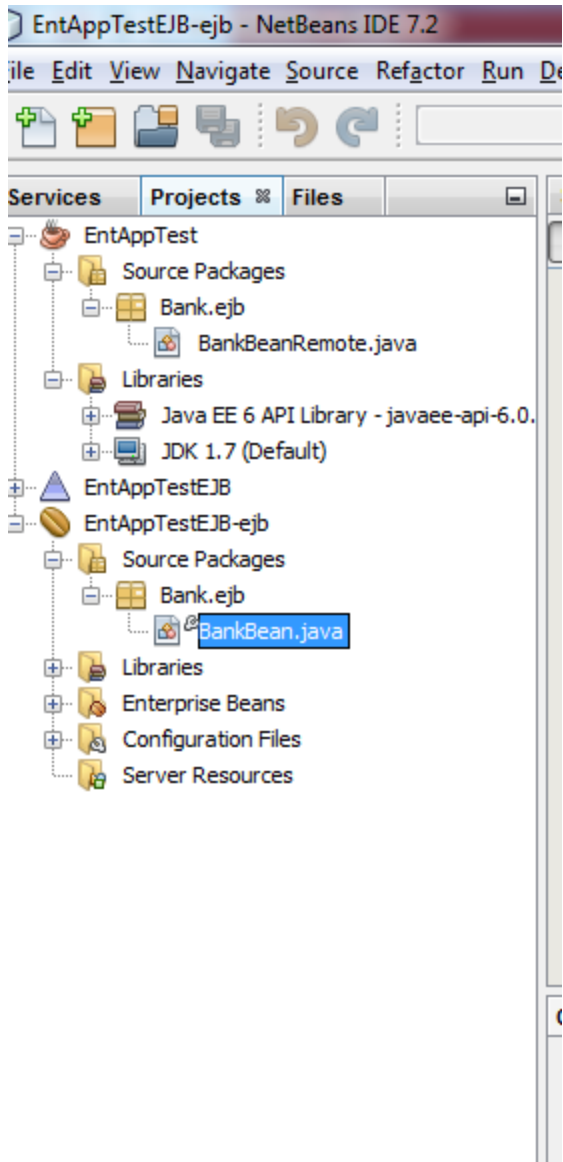
Right-click the EJB module project and choose New > Session Bean.



In the following window select the name and type of the session bean as follows:



Click Finish and you will see the following:



Adding the Methods

Now we need to add the methods in the session bean. Do the following:

Right-click in the editor of BankBean and choose Insert Code and select Add Business Method as follows:

EntAppTestEJB-ejb - NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

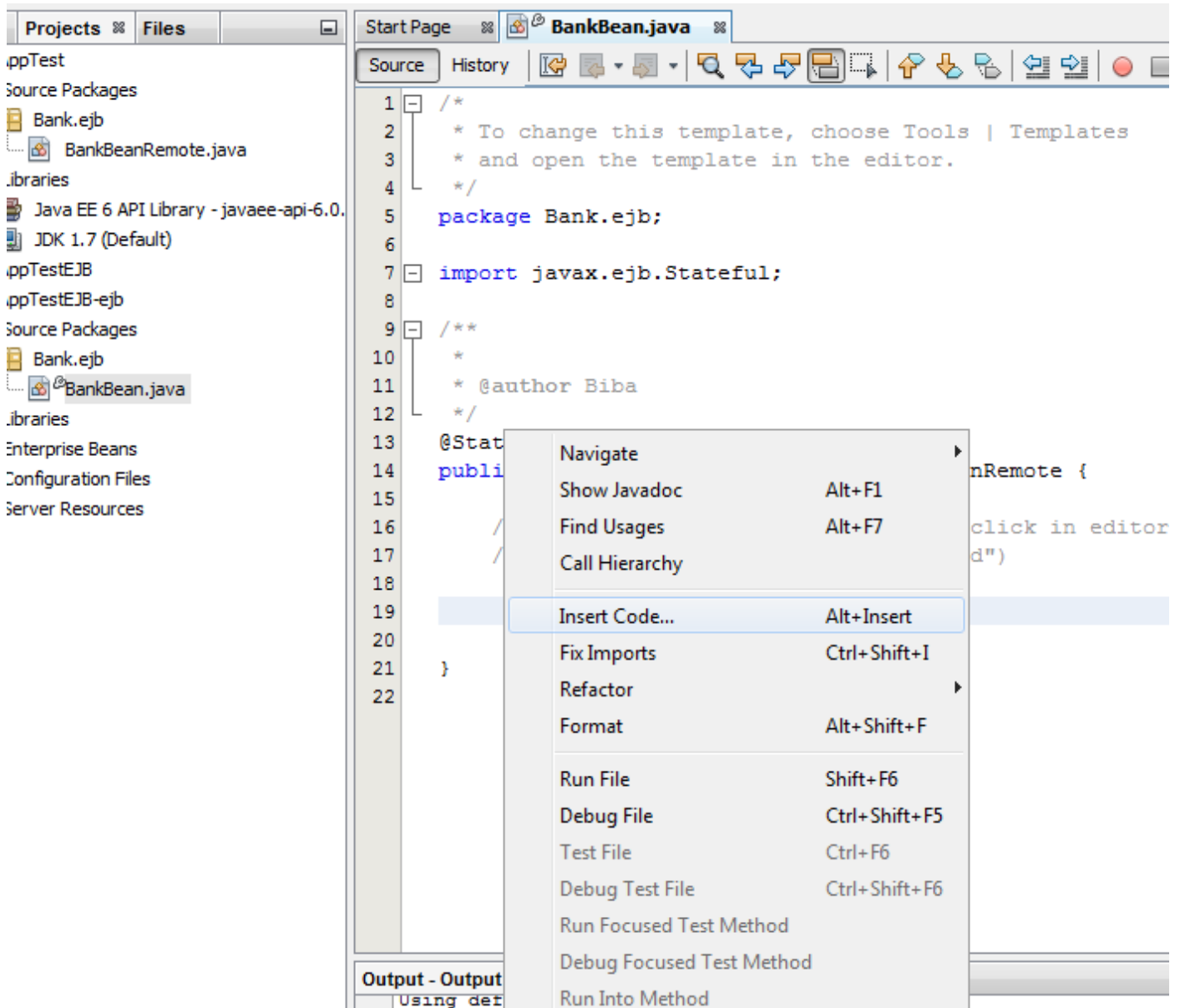
Services Projects Files Start Page BankBean.java

EntAppTest

- Source Packages
 - Bank.ejb
 - BankBeanRemote.java
- Libraries
 - Java EE 6 API Library - javaee-api-6.0.
 - JDK 1.7 (Default)
- EntAppTestEJB
- EntAppTestEJB-ejb
 - Source Packages
 - Bank.ejb
 - BankBean.java
 - Libraries
 - Enterprise Beans
 - Configuration Files
 - Server Resources

Source History

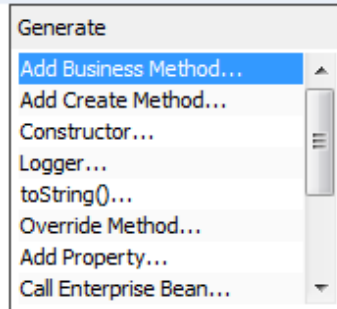
```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package Bank.ejb;
6
7  import javax.ejb.Stateful;
8
9  /**
10 *
11 * @author Biba
12 */
13 @Stateful
14 public class BankBean implements BankBeanRemote {
15
16     // Add business logic below. (Right-click in editor and choose
17     // "Insert Code > Add Business Method")
18
19
20
21 }
22
```



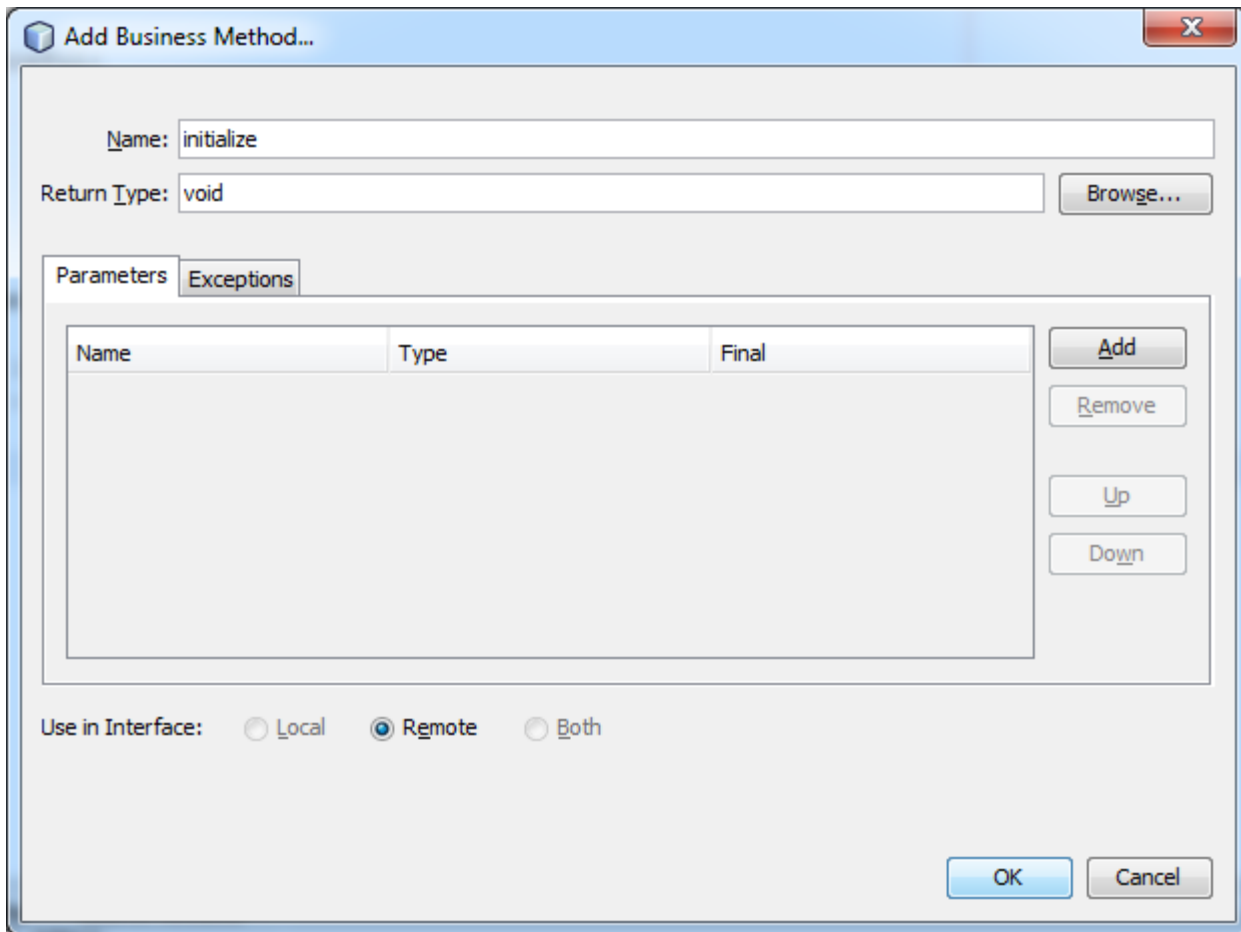
e.java

y - javaee-api-6.0.

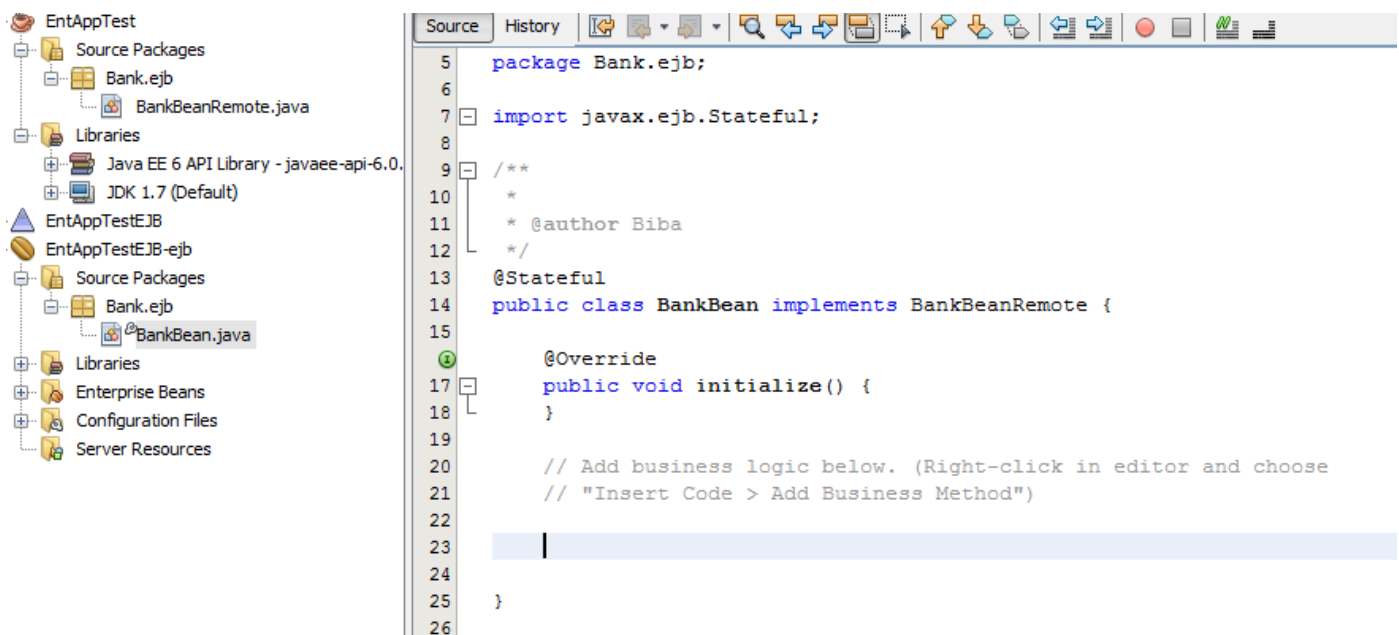
```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package Bank.ejb;
6
7  import javax.ejb.Stateful;
8
9  /**
10 *
11 * @author Biba
12 */
13 @Stateful
14 public class BankBean implements BankBeanRemote {
15
16     // Add business logic below. (Right-click in editor and choose
17     // "Insert Code > Add Business Method")
18
19
20
21 }
22
```



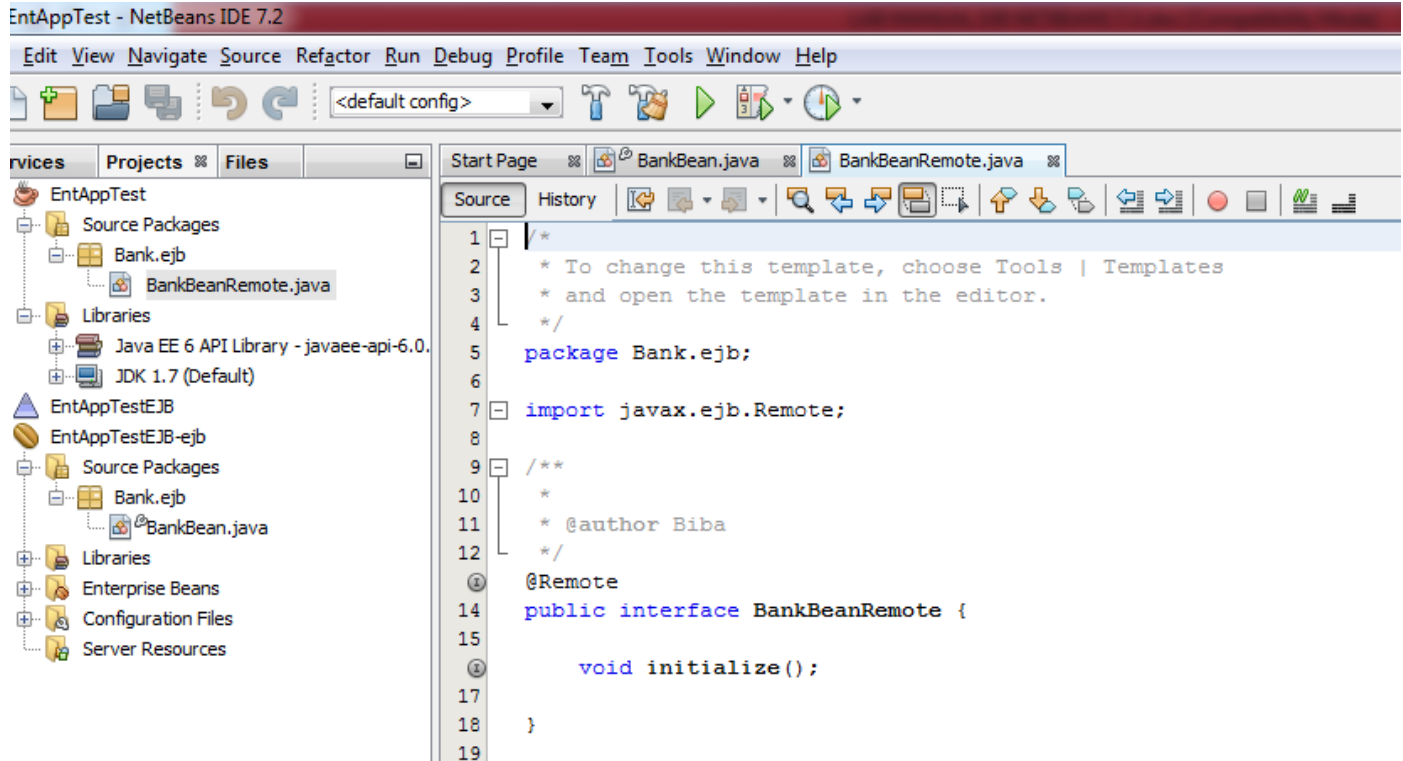
The following windows appears. Write the name of the method: “initialize” and return type void.



After you click Ok you will see:

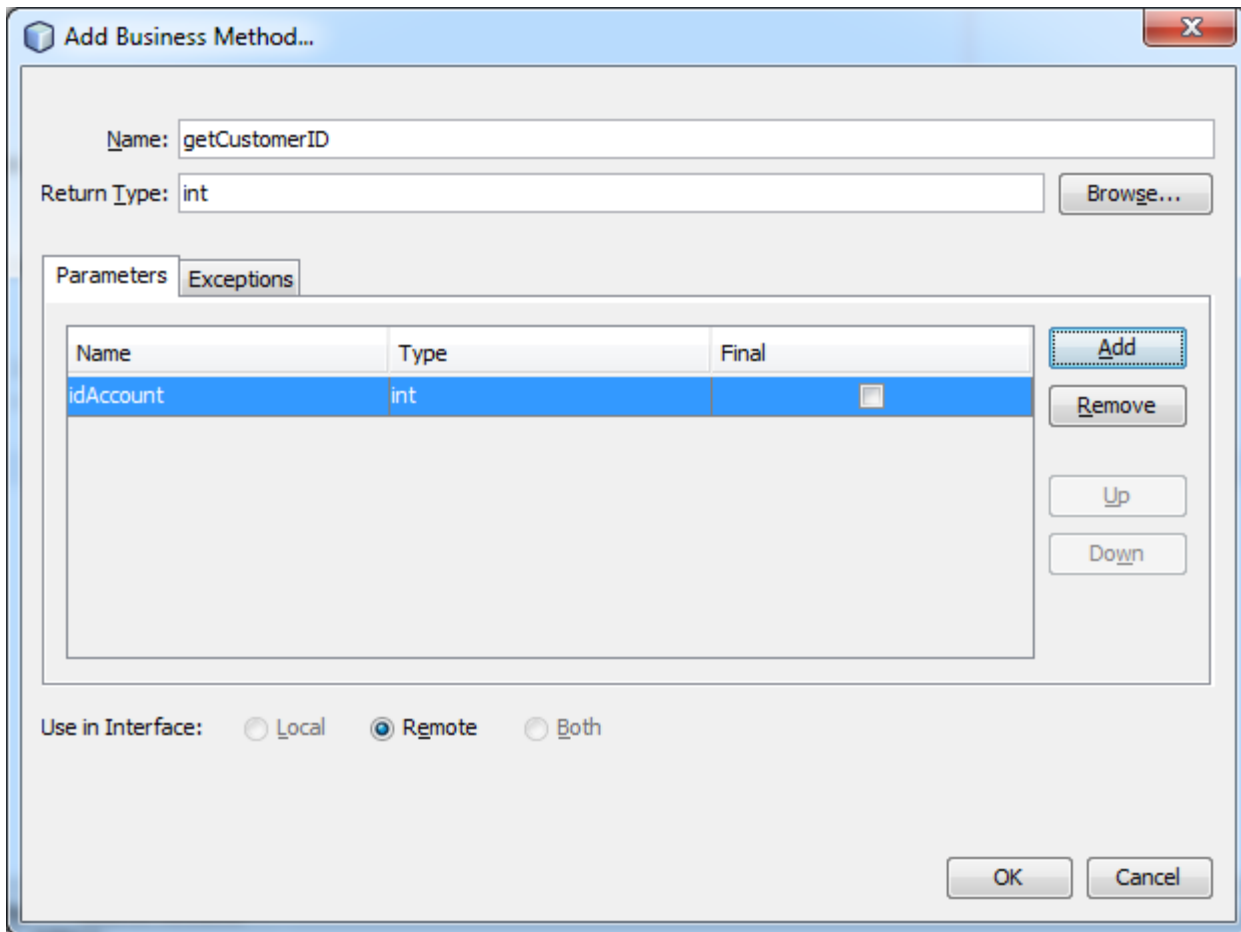


After you save all in Netbeans in the BankBeanRemote you will see:



```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package Bank.ejb;
6
7  import javax.ejb.Remote;
8
9  /**
10 *
11 * @author Biba
12 */
13 @Remote
14 public interface BankBeanRemote {
15
16     void initialize();
17
18 }
19
```

We do the same for the other method “getCustomerID”.



You will have the following:

The image shows an IDE window with two tabs: "BankBean.java" and "BankBeanRemote.java". The "BankBean.java" tab is active, displaying the following code:

```
7 import javax.ejb.Stateful;
8
9 /**
10  *
11  * @author Biba
12  */
13 @Stateful
14 public class BankBean implements BankBeanRemote {
15
16     @Override
17     public void initialize() {
18     }
19
20     // Add business logic below. (Right-click in editor and choose
21     // "Insert Code > Add Business Method")
22
23     @Override
24     public int getCustomerID(int idAccount) {
25         return 0;
26     }
27
28
29
30 }
31
```

Now write the code of the methods as follows:

```

4  L  */
5  package Bank.ejb;
6
7  [ ] import java.util.HashMap;
8  L  import javax.ejb.Stateful;
9
10 [ ] /**
11     *
12     * @author Biba
13     */
14 @Stateful
15 public class BankBean implements BankBeanRemote {
16
17     // Add business logic below. (Right-click in editor and choose
18     // "Insert Code > Add Business Method")
19
20     HashMap custAndAccounts = new HashMap();
21
22     @Override
23 [ ] public void initialize() {
24
25         // put some customers inside
26
27         custAndAccounts.put(1, 1);
28         custAndAccounts.put(2, 1);
29
30     }
31
32     @Override
33 [ ] public int getCustomerID(int idAccount) {
34     return (Integer)custAndAccounts.get(idAccount);
35     }
36
37
38 }
39

```

We now have an enterprise application with a simple EJB that is exposed through a remote interface. We also have an independent class library that contains the EJB interface that can be distributed to other developers. Developers can add the library to their projects if they want to communicate with the EJB that is exposed by the remote interface and do not need to have the sources for the EJB. When we modify the code for the EJB, we only need to distribute a JAR of the updated class library if any of the interfaces change.

When we use the Add Business Method dialog, the IDE automatically implements the method in the remote interface.

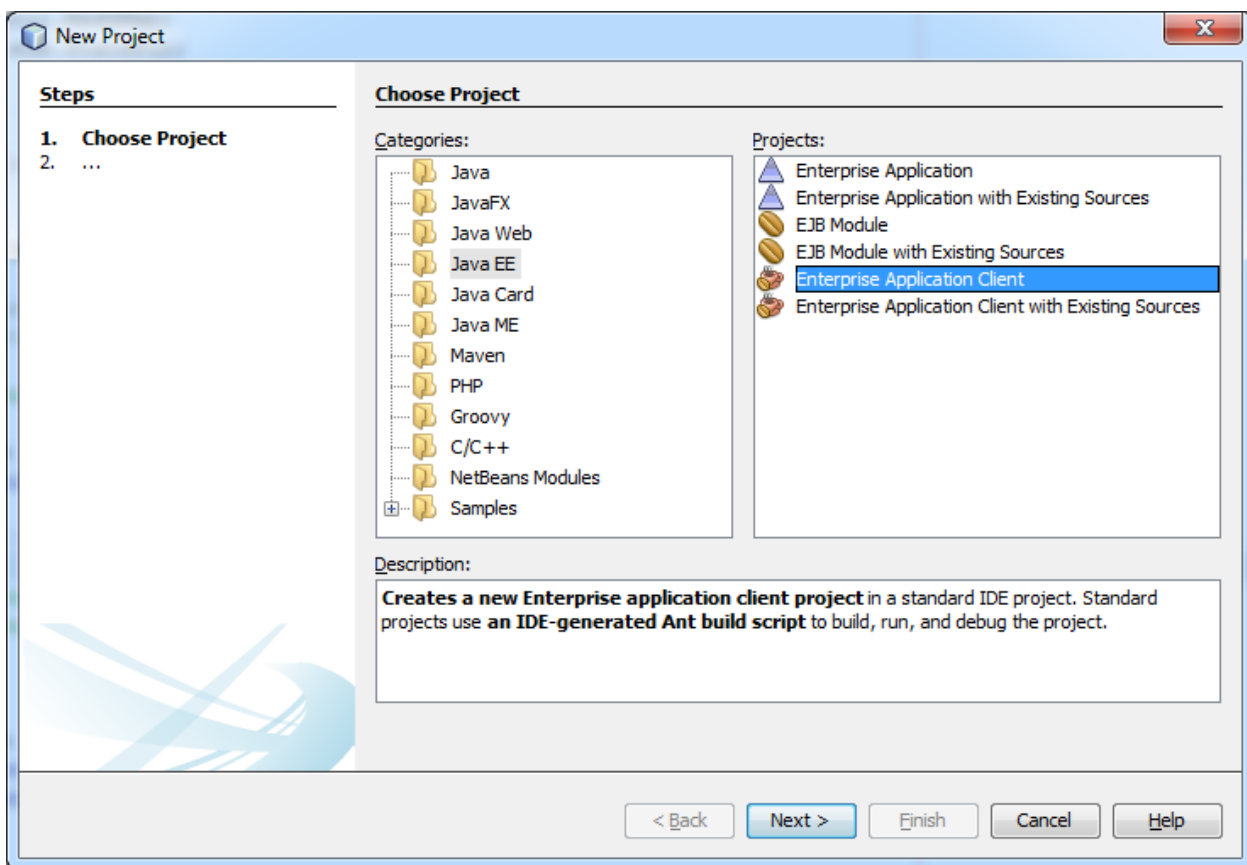
Creating the Application Client

We will now create an enterprise application client. When creating the application client, the project needs the Java class library as a library in order to reference the EJB.

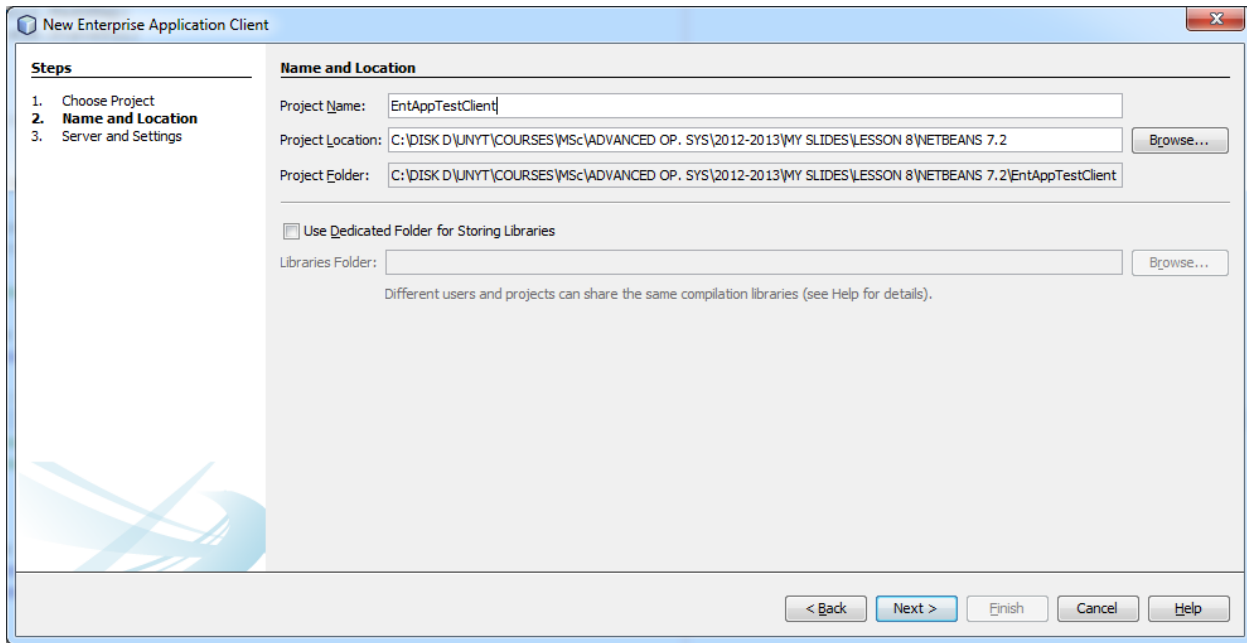
When you run the enterprise application, the IDE will package the application client and the Java class library JAR in the EAR archive. Library JARs must be packaged in an EAR with the application client if you want to access the JARs from the application client.

To create the application client do the following:

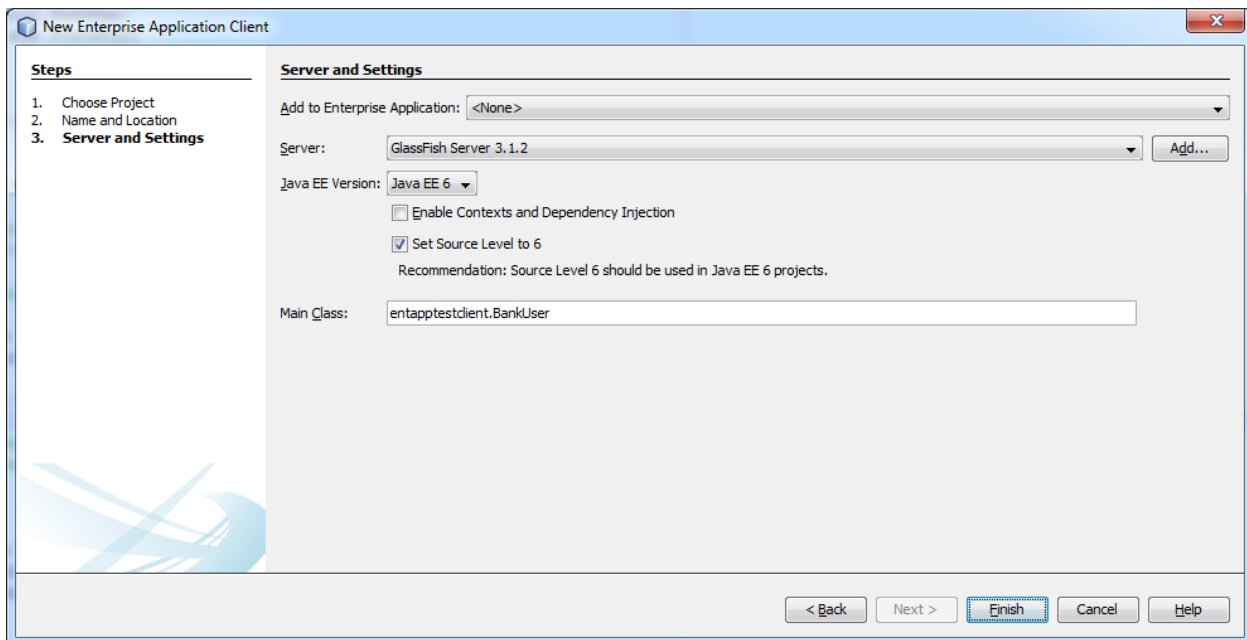
Choose File > New Project and select Enterprise Application Client in the Java EE category.



Click Next.



Click Next.



Click Finish.

Adding the Class Library

The class library that contains the remote interface now needs to be added to the classpath of the project to enable the application client to reference the EJB. The class library project is open, so you can use the Call Enterprise Bean dialog to help you generate the code to call the EJB.

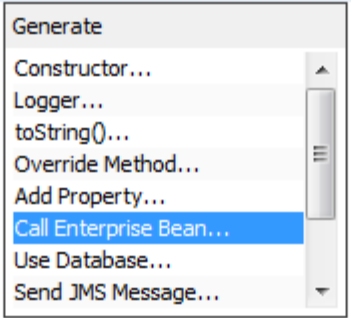
Go to BankUser and click on the editor:

```
3 | * and open the template in the editor.  
4 | */  
5 | package entapptestclient;  
6 |  
7 | /**  
8 | *  
9 | * @author Biba  
10 | */  
11 | public class BankUser {  
12 |  
13 |     /**  
14 |     * @param a  
15 |     */  
16 |     public stat  
17 |         // TODO  
18 |  
19 |  
20 |  
21 |     }  
22 | }  
23 |
```

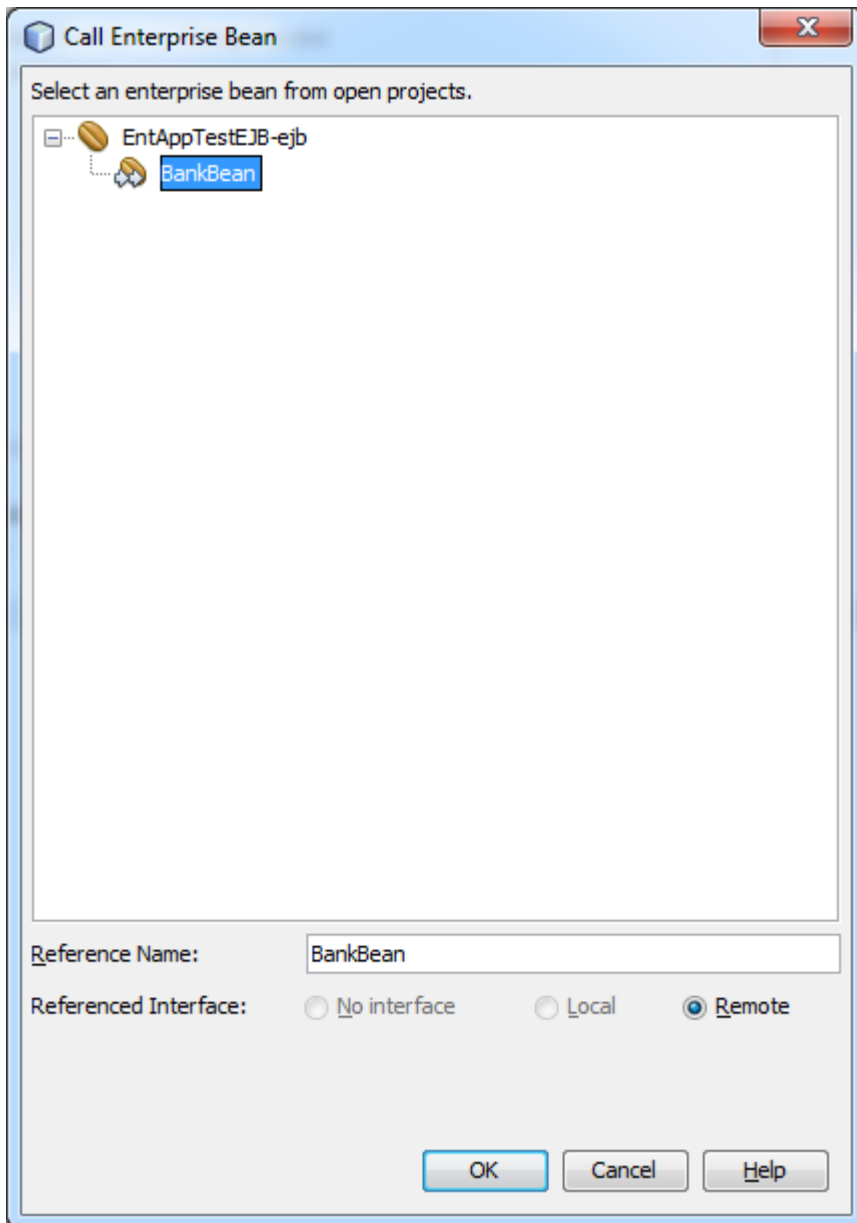
Navigate	
Show Javadoc	Alt+F1
Find Usages	Alt+F7
Call Hierarchy	
Insert Code...	Alt+Insert
Fix Imports	Ctrl+Shift+I
Refactor	
Format	Alt+Shift+F
Run File	Shift+F6
Debug File	Ctrl+Shift+F5
Test File	Ctrl+F6
Debug Test File	Ctrl+Shift+F6
Run Focused Test Method	
Debug Focused Test Method	
Run Into Method	

Select Insert Code and then Call Enterprise Bean:

```
10 | */
11 | public class BankUser {
12 |
13 | | /**
14 | |  * @param args the command line arguments
15 | |  */
16 | | public static void main(String[] args) {
17 | |     // TODO code application logic here
18 | |
19 | | _____
20 | |
21 | | }
22 | }
23 |
```

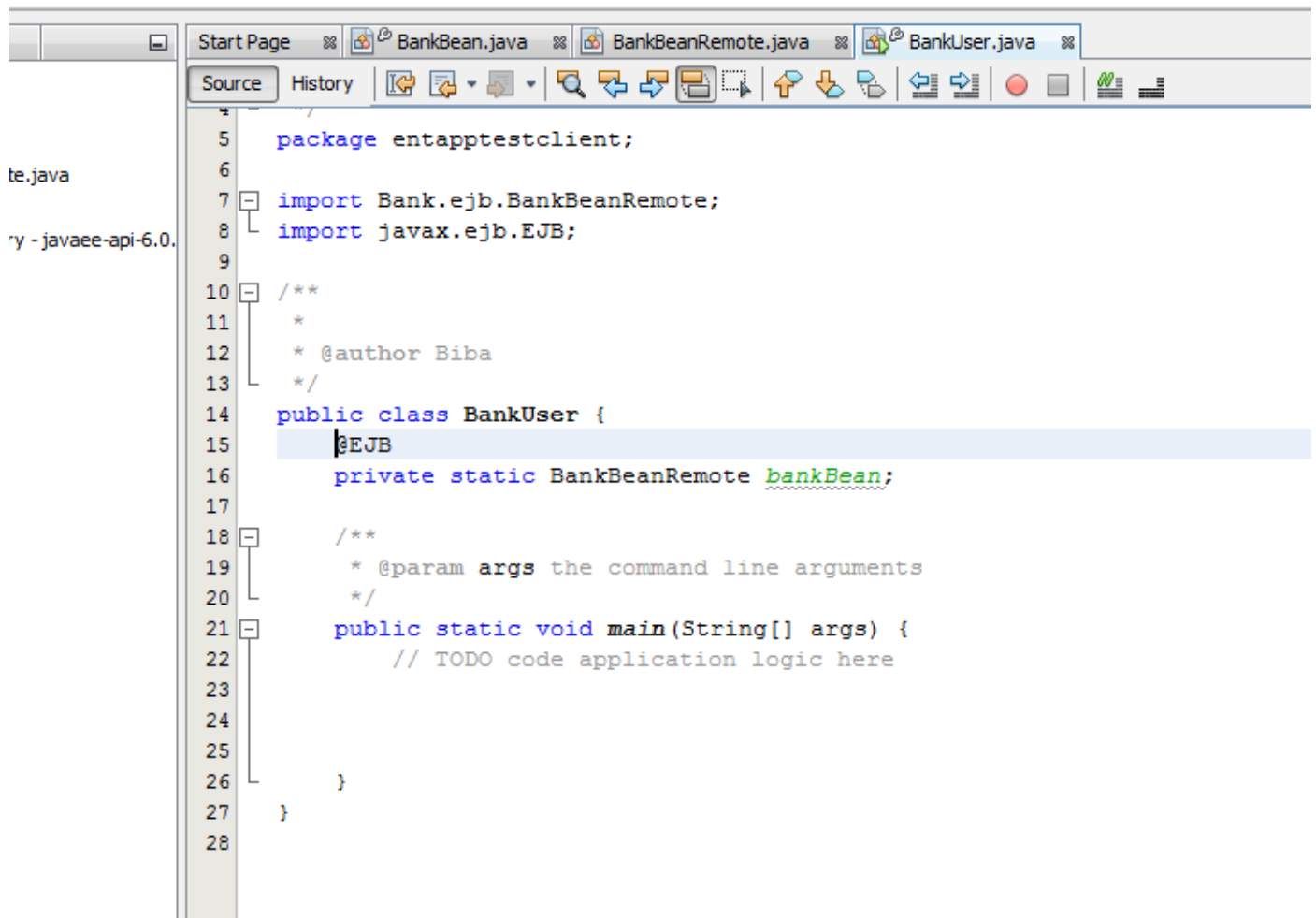


The following appears:



Click Ok.

You will see the following:



```
4  
5 package entapptestclient;  
6  
7 import Bank.ejb.BankBeanRemote;  
8 import javax.ejb.EJB;  
9  
10 /**  
11  *  
12  * @author Biba  
13  */  
14 public class BankUser {  
15     @EJB  
16     private static BankBeanRemote bankBean;  
17  
18     /**  
19     * @param args the command line arguments  
20     */  
21     public static void main(String[] args) {  
22         // TODO code application logic here  
23  
24  
25  
26     }  
27 }  
28
```

Now you may add the following code to the methods:


```

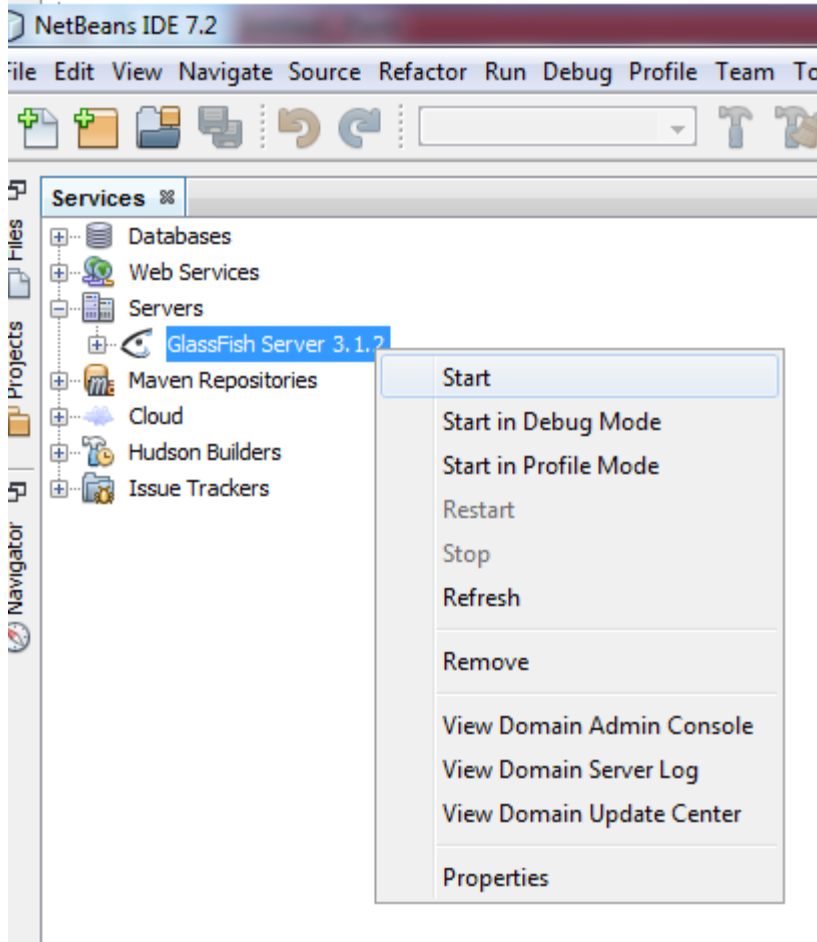
5  package entapptestclient;
6
7  import Bank.ejb.BankBeanRemote;
8  import javax.ejb.EJB;
9
10 /**
11  *
12  * @author Biba
13  */
14 public class BankUser {
15     @EJB
16     private static BankBeanRemote bankBean;
17
18     public BankUser(String[] args) {
19     }
20
21     /**
22     * @param args the command line arguments
23     */
24     public static void main(String[] args) {
25         BankUser client = new BankUser(args);
26         client.start();
27     }
28
29     public void start() {
30         bankBean.initialize();
31         int account = 1;
32         System.out.println("-----");
33         System.out.println("Bank User Started.");
34         System.out.println("Executing the query to the bean");
35         System.out.println("The Customer with account " + account + " is: "
36             + bankBean.getCustomerID(account));
37         System.out.println("-----");
38     }
39 }

```

Save all in Netbeans.

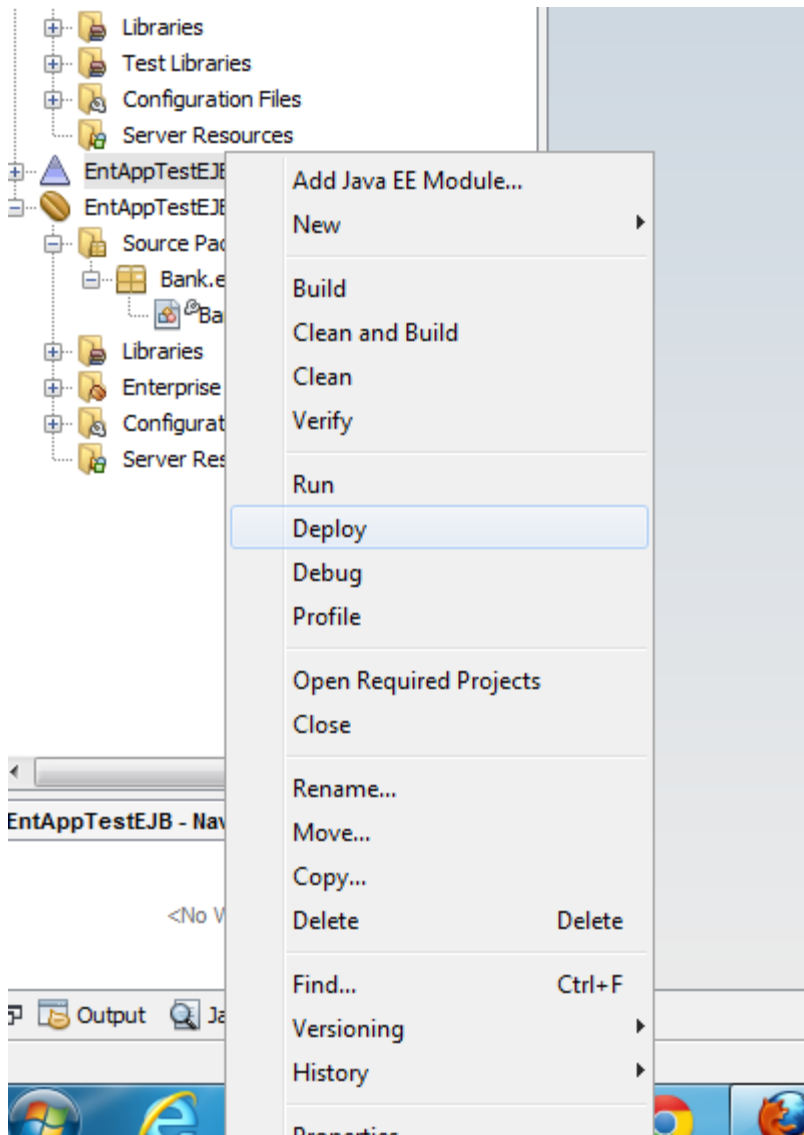
Running the application

You first need to start the Glassfish Server as follows:



In the above click Start

You first need to deploy the Enterprise Application. Right-click the EntAppTestEJB enterprise application and choose Deploy.



If you check the administration panel of the server you will see the following:

Applications

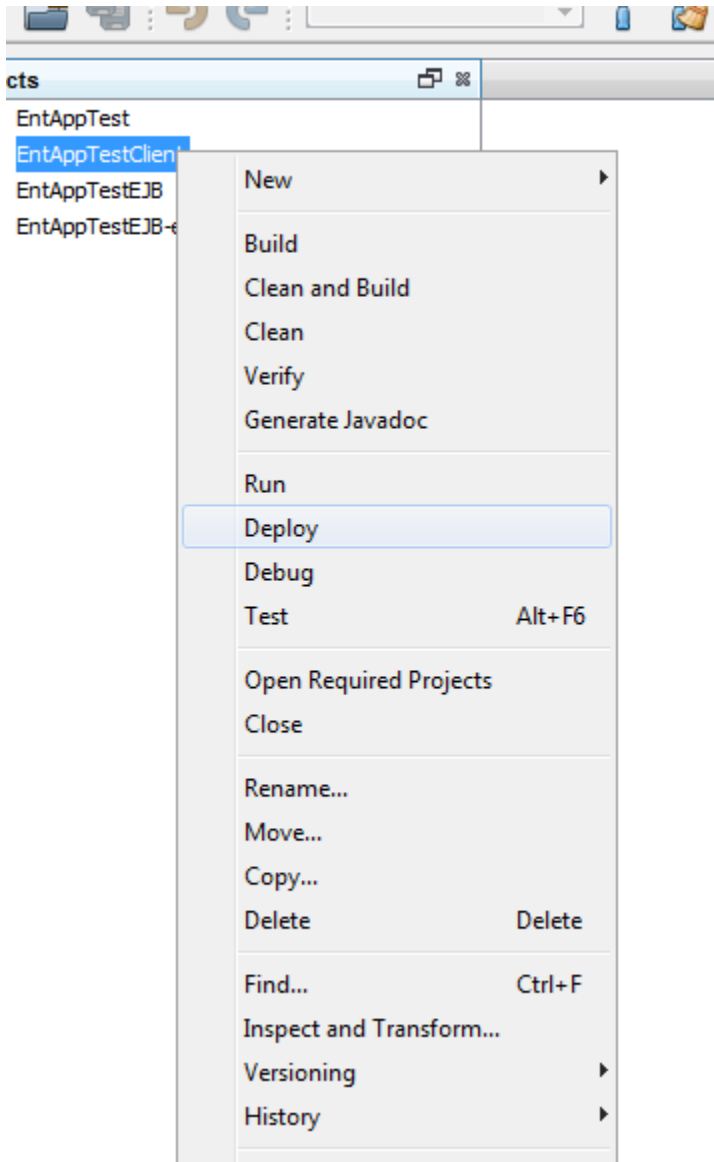
Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link.

Deployed Applications (5)

| [Deploy...](#) [Undeploy](#) [Enable](#) [Disable](#) | Filter:

	Name	Enabled	Engines
<input type="checkbox"/>	EntAppTestEJB	✓	ear, ejb
<input type="checkbox"/>	converter	✓	ejb, web
<input type="checkbox"/>	counter	✓	ejb, web
<input type="checkbox"/>	helloservice	✓	ejb, webservices
<input type="checkbox"/>	timersession	✓	ejb, web

Now you can now deploy the application client by clicking with the right of the mouse and clicking on Deploy:



Now you can now deploy the application client by clicking with the right of the mouse and clicking on Deploy:

You will see in Glassfish:

GlassFish™ Server Open Source Edition

Applications

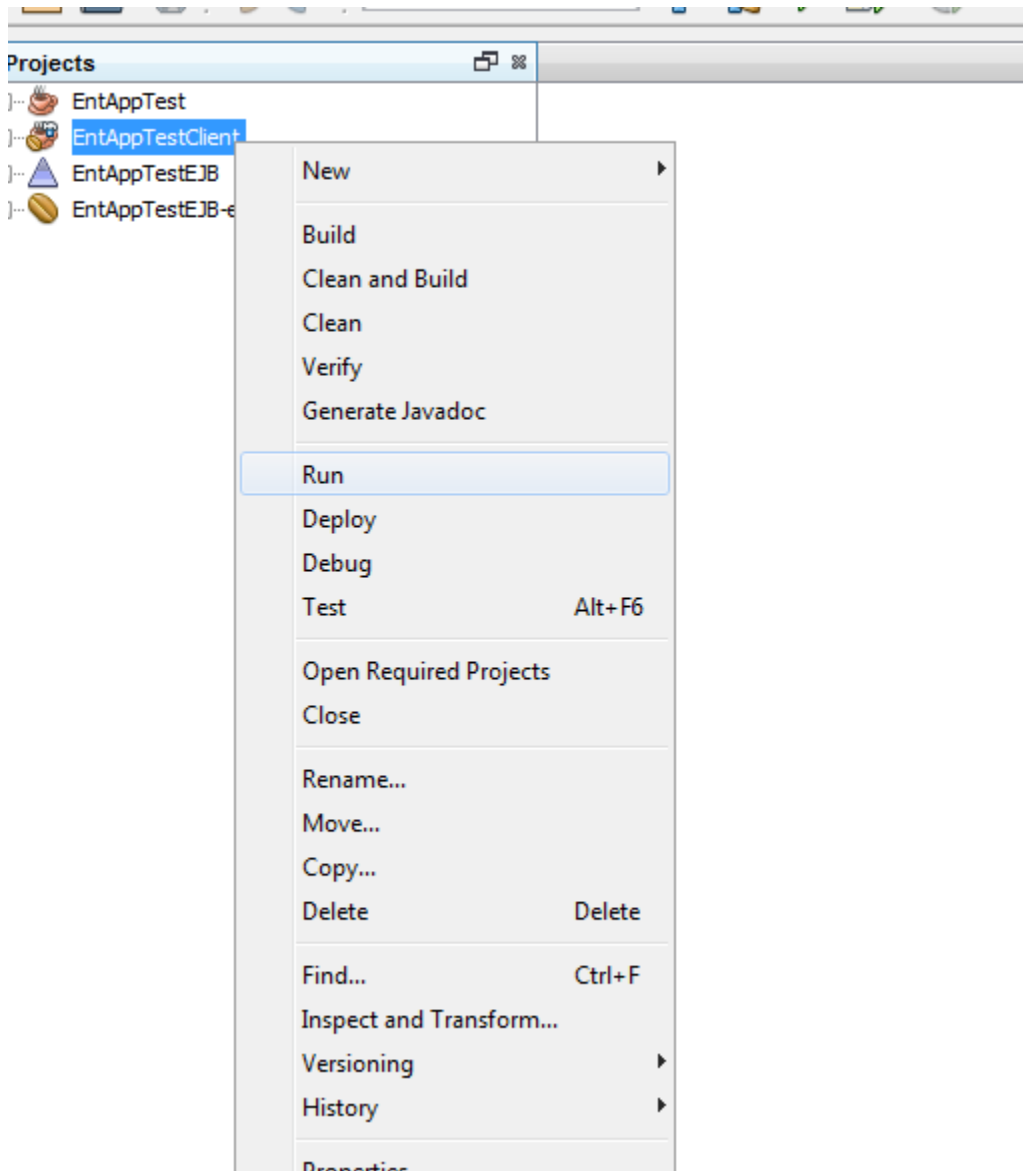
Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking

Deployed Applications (6)

| [Deploy...](#) [Undeploy](#) [Enable](#) [Disable](#) | Filter:

	Name	Enabled	Engines
<input type="checkbox"/>	EntAppTestClient	✓	appclient
<input type="checkbox"/>	EntAppTestEJB	✓	ear, ejb
<input type="checkbox"/>	converter	✓	ejb, web
<input type="checkbox"/>	counter	✓	ejb, web
<input type="checkbox"/>	helloservice	✓	ejb, webservices
<input type="checkbox"/>	timersession	✓	ejb, web

Now you can now run the application client by clicking with the right of the mouse and clicking on Run:



You will see the following execution:

Output

Java DB Database Process GlassFish Server 3.1.2 EntAppTestClient (run)

▶ Redeploying C:\DISK D\UNYT\COURSES\MSc\ADVANCED OP. SYS\2012-2013\MY SLIDES\LESSON 8\NETBEANS 7.2\
▶ Initializing...
■ post-run-deploy:
🔍 run-deploy:
Copying 1 file to C:\DISK D\UNYT\COURSES\MSc\ADVANCED OP. SYS\2012-2013\MY SLIDES\LESSON 8\NETBEAN
Copying 2 files to C:\DISK D\UNYT\COURSES\MSc\ADVANCED OP. SYS\2012-2013\MY SLIDES\LESSON 8\NETBEA
Warning: C:\DISK D\UNYT\COURSES\MSc\ADVANCED OP. SYS\2012-2013\MY SLIDES\LESSON 8\NETBEANS 7.2\Ent

Bank User Started.
Executing the query to the bean
The Customer with account 1 is: 1

run:
BUILD SUCCESSFUL (total time: 14 seconds)
|