

IBM® Feature Pack for Web Services– LAB EXERCISE

Feature Pack for Web Services

Top-down development of a JAX-WS service

What this exercise is about	1
Lab requirements	1
What you should be able to do	2
Introduction	2
Development models	2
Exercise instructions	3
Part 1: Create a dynamic Web project.....	5
Part 2: Import the WSDL file	12
Part 3: Creating a Web service from the WSDL file.....	18
Part 4: Install and deploy the application	27
Part 5: Creating the client code.....	41
Summary.....	56

What this exercise is about

The objective of this lab is to show you how to develop a complete Web services application using the Feature Pack for Web Services for WebSphere Application Server V6.1.

You will take a top down approach beginning with only a WSDL file. It will then show you how to develop a Web Service, package your code and artifacts into an EAR file, install the EAR file as an application for WebSphere Application Server version 6.1, start the application, perform a simple test to confirm the application is deployed and started correctly, create a Web services client, and then do an end-to-end test using the client.

Lab requirements

The system on which you are installing WebSphere Application Server must be running a supported distributed operating system. For more information on supported operating systems, see the WebSphere Application Server V6.1 information center. In addition, the system must have the following resources available:

- An installation of IBM Web Services Feature Pack for WebSphere Application Server V6.1
- An installation of IBM Application Server Toolkit V6.1

What you should be able to do

At the end of this lab you should be able to:

- Perform an end-to-end test from the client to the Web Service and back.

Use a Top Down approach to develop a Web Service

- Take an existing WSDL file and, using the IBM WebSphere Application Server Toolkit Version 6.1, create a Web Service for the IBM Feature Pack for Web Services for WebSphere Application Server V6.1
- Package the Web Service into an EAR file
- Deploy the EAR file onto WebSphere Application Server V6.1
- Start the Web Service application
- Perform a simple test to confirm that the application is installed and listening at the expected URL
- Develop a client for the Web Service application
- Perform an end-to-end test from the client to the Web Service and back.

Introduction

The Feature Pack for Web Services builds on Web Services technologies available in WebSphere Application Server V6.1. A few of the major enhancements introduced in this Feature Pack are:

- A new JAX-WS programming model (with support for JAX-B, annotations, SOAP 1.2 and more)
- Support for WS-Reliable Messaging and WS-Secure Conversation
- Improved administration of Web Services configuration through the use of Policy Sets

Development models

Web services can be created using one of two approaches: top-down development and bottom-up development. Top-down Web services development involves creating a Web service from a WSDL file. Bottom-up Web services development involves creating a Web service from a Java file that contains code that will be the basis for implementing the Web service.

When creating a Web service using a top-down approach, first you design the implementation of the Web service by creating a WSDL file. You can do this using the WSDL Editor. You can then use the Web services wizard to create the Web service and skeleton Java™ classes to which you can add the required code.

Although bottom-up Web service development may be faster and easier, especially if you are new to Web services, the top-down approach is the recommended way of creating a Web service – especially if interoperability is a concern. By creating the WSDL file first you will ultimately have more control over the Web service, and can eliminate interoperability issues that may arise when creating a Web service using the bottom-up method.

Lab Exercise –Top down development

Part 1: Create a dynamic Web project

In this portion of the lab you will create a dynamic Web project in the AST.

Part 2: Import the WSDL File

This part of the lab will show you how to import the WSDL file into the dynamic Web project you created in Part 1.

Part 3: Creating a Web service from the WSDL file

This section will show how to create a Web service from the WSDL file and how to export the EAR file containing that Web service.

Part 4: Install and deploy the application

This section will show how to install the EAR file from Part 3, how to start the Web Service application from the administrative console, and how to do some simple verification.

Part 5: Creating the client code

This section will show you how to develop a Web Services client for the Web Service that has been created. Then shows how to run the client and interact with the Web Service that was created and installed.

Exercise instructions

Instructions and subsequent documentation use symbolic references to directories which are listed as follows:

Reference Variable	Windows Location	Linux UNIX Location
<WAS_HOME>	C:\WebSphere\AppServer	Linux /opt/WebSphere/AppServer AIX /usr/WebSphere/AppServer
<TEMP>	C:\temp	/tmp
<hostname>	Host name or host address for the machine where the profiles are being created	Host name or host address for the machine where the profiles are being created

Note:

This Lab exercise assumes that the following software is already installed on your system. Use the documentation associated with the software to identify prerequisite software and hardware and for installation and verification instructions:

- This Lab exercise assumes that the following software is already installed on your system. Use the documentation associated with the software to identify prerequisite software and hardware and for installation and verification instructions: IBM WebSphere Application Server Version 6.1.0.9

- IBM WebSphere Application Server Version 6.1 Feature Pack for Web Services
- IBM WebSphere Application Server Toolkit Version 6.1.1.3 and the add-on for the Feature Pack for Web Services.

Part 1: Create a dynamic Web project

This section will describe the process of creating a Dynamic Web Project – an environment that will be used to build a Web Service application using the IBM WebSphere Application Server Toolkit version 6.1 (AST). The AST comes with IBM WebSphere Application Server V6.1 and this release of AST has specific support for the IBM WebSphere Application Server V6.1 Feature Pack for Web Services (WSFP).

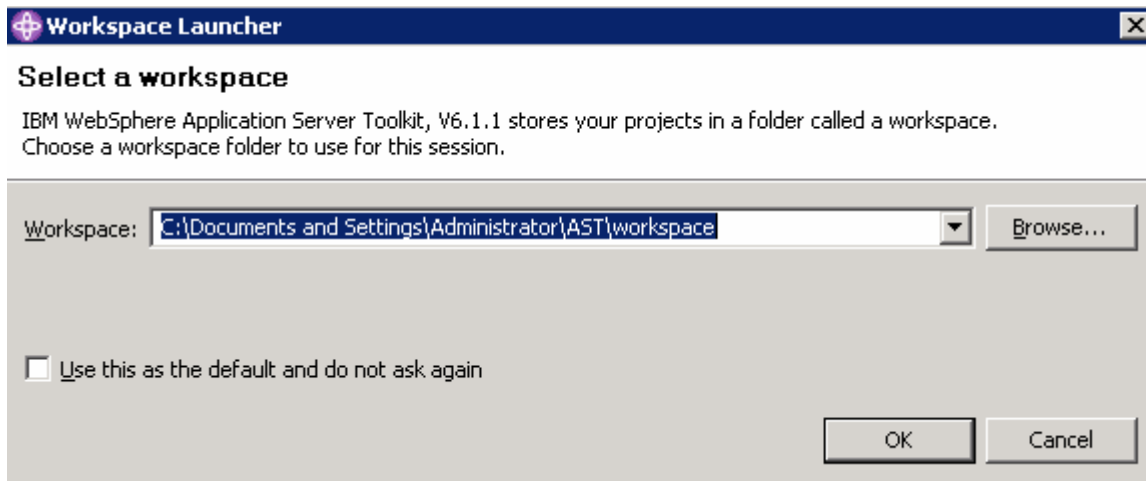
- ___ 1. Start the AST workspace.

To start the AST:

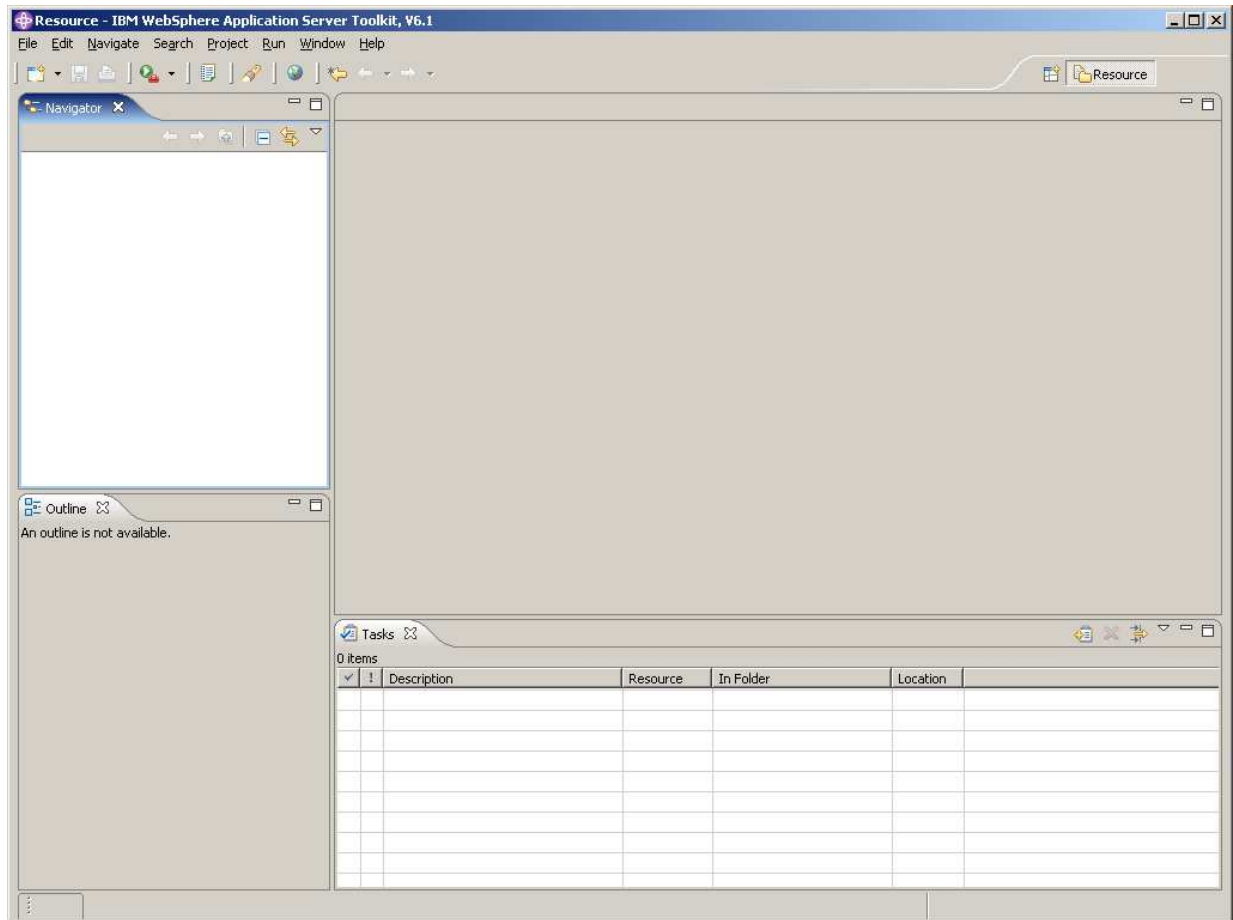
Using the Start Menu:

Start-> All Programs -> IBM WebSphere -> Application Server Toolkit V6.1-> Application Server Toolkit

- ___ 2. When prompted to select a workspace, keep the default workspace of **C:\Documents and Settings\Administrator\AST\workspace** and click **OK**.

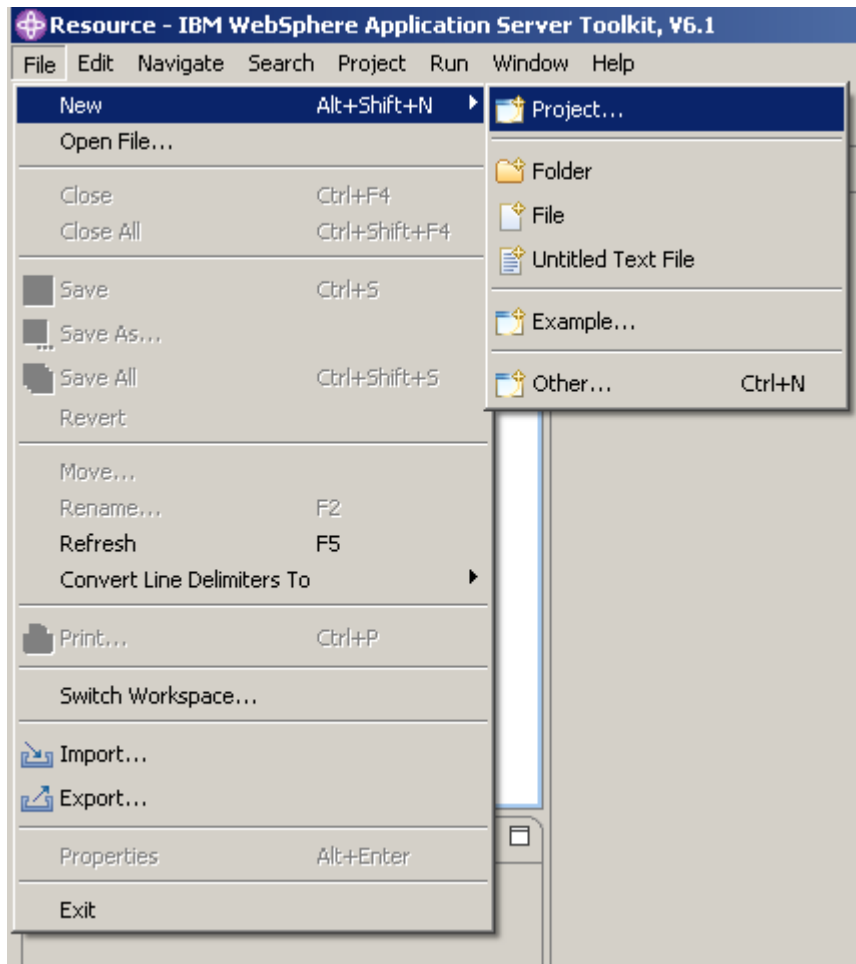


3. This will open the AST. Close the welcome screen by clicking the **X** on the welcome tab and you will see the screen shown below.

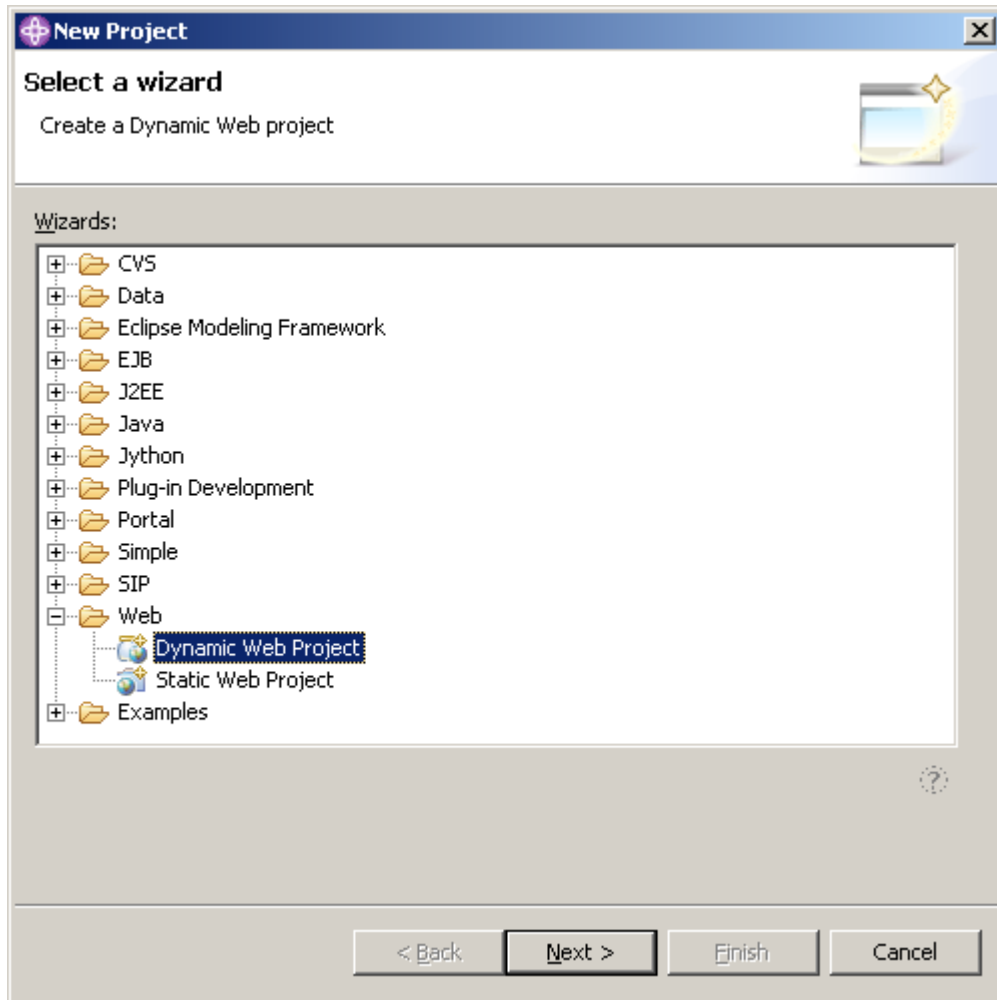


___ 4. Creating a new Dynamic Web Project.

___ a. Select **File** from the navigation bar. Select **New -> Project** to create a new project.



__ b. Expand the Web folder and select **Dynamic Web Project**. Click **Next**.



- c. For the Project Name enter **WSFPLab1**. Check the **Add project to an EAR** checkbox. Enter **WSFPLab1EAR** for the EAR Project Name. Click **Next**.

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project contents:
 Use default

Directory:

Target Runtime

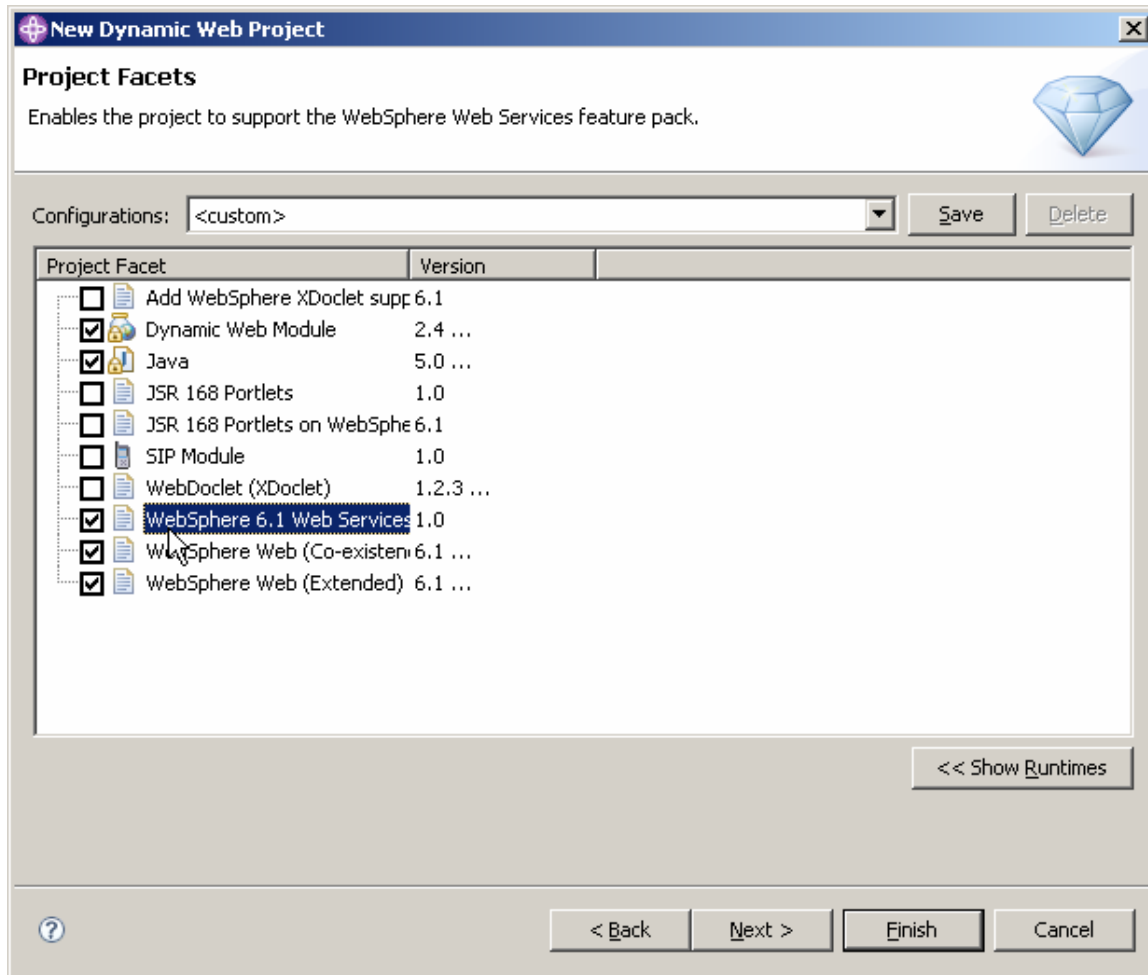
Configurations

Hint: Get started quickly by selecting one of the pre-defined project configurations.

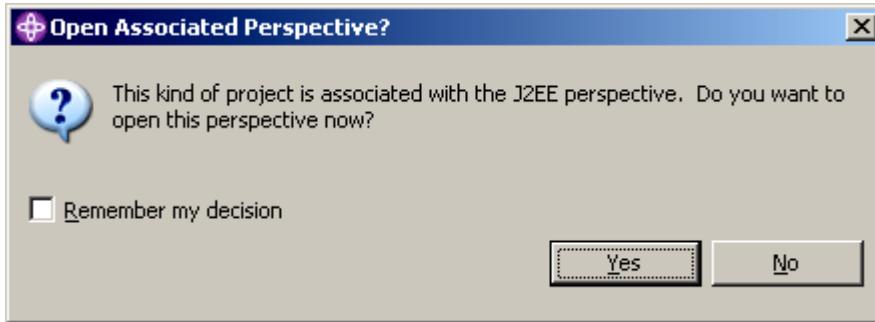
EAR Membership
 Add project to an EAR

EAR Project Name:

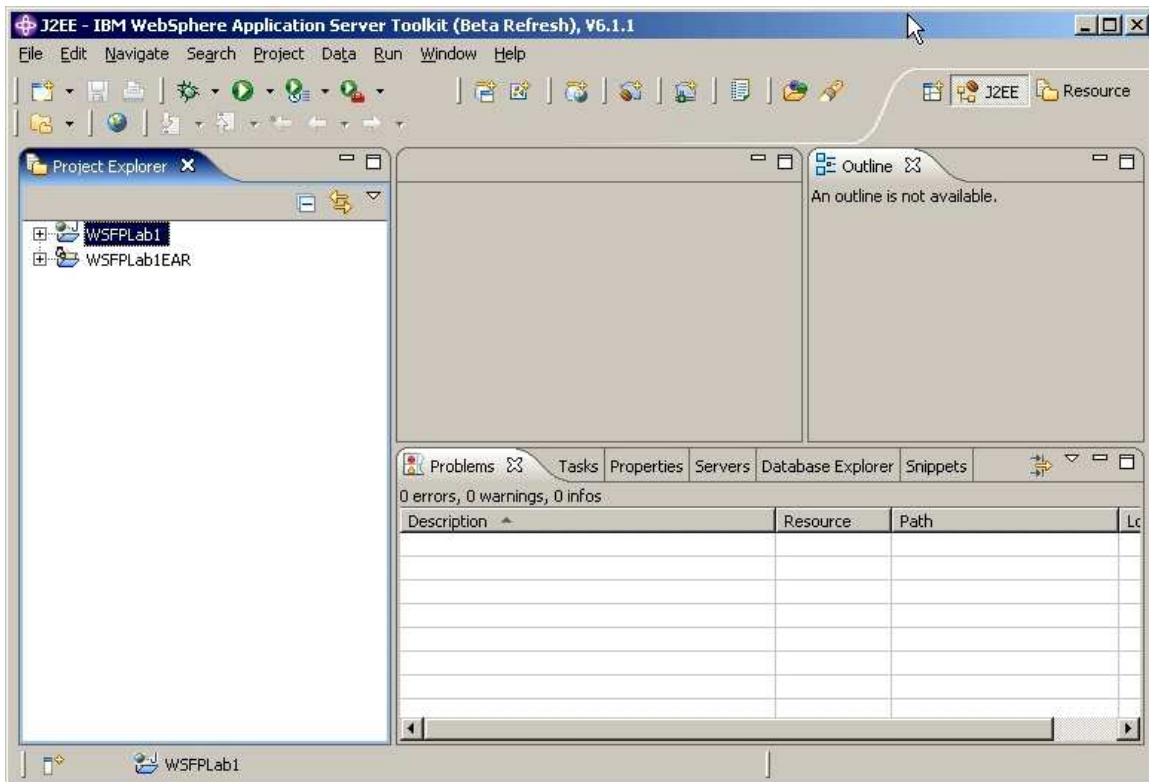
___ d. This will open the **Project Facets** panel. Click on the check box for **WebSphere 6.1 Web Services Feature Pack** and then click **Finish**.



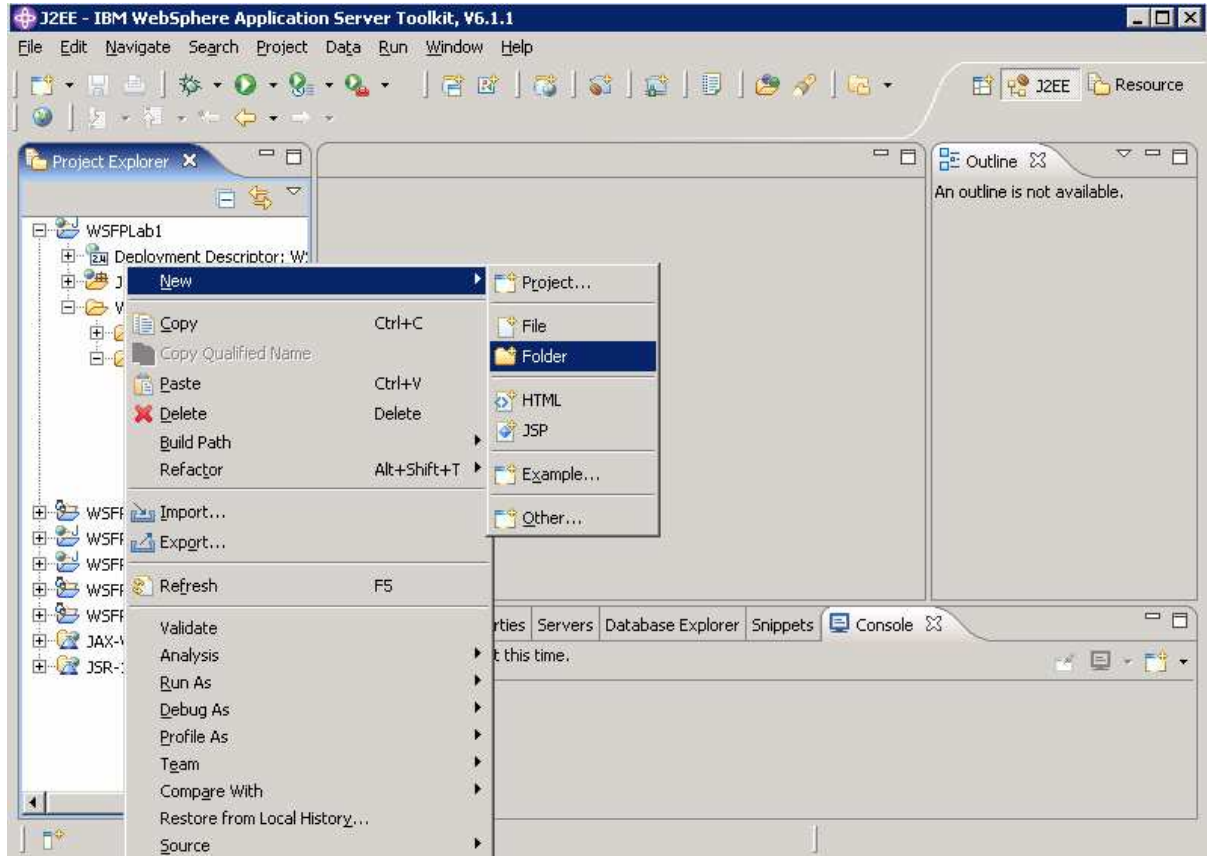
__ e. If you see a prompt asking to open the J2EE perspective, click **Yes**.



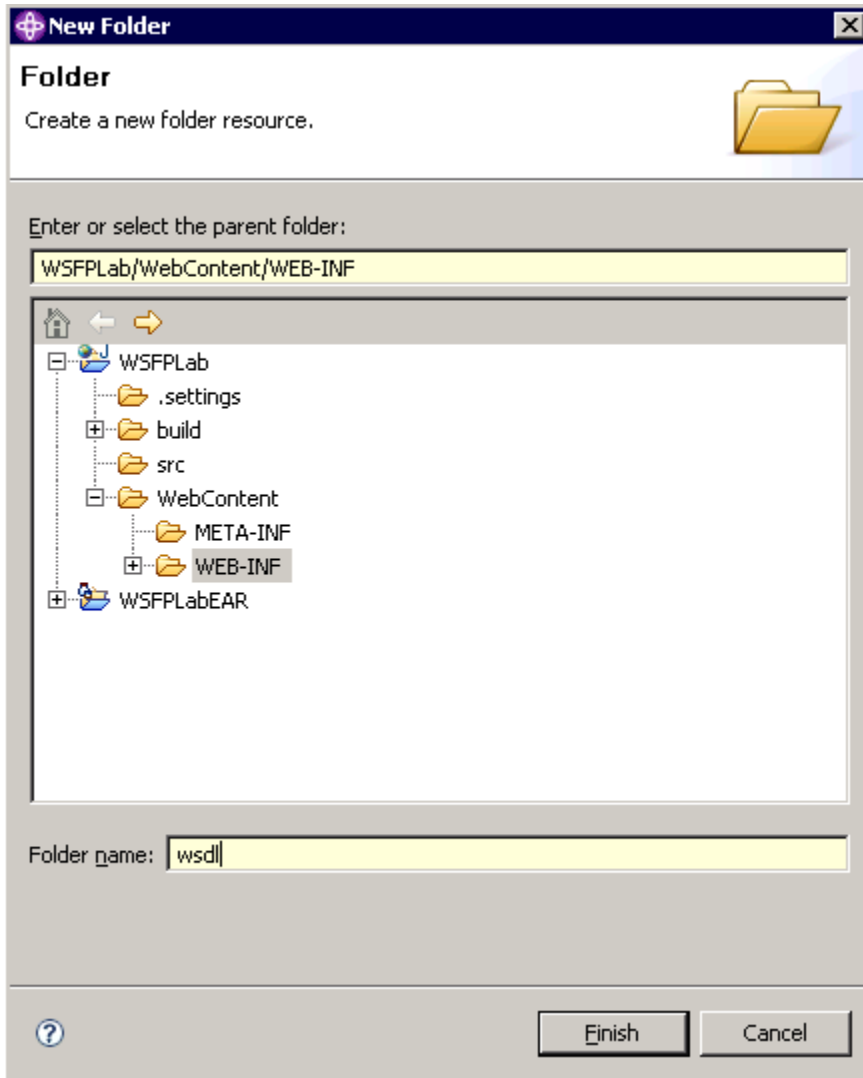
__ f. The Dynamic Web Project should now be visible in the Project Explorer panel of the AST.



- ___ d. Right click on **WEB-INF**, select **New** and **Folder** to create a new folder to hold the WSDL document.

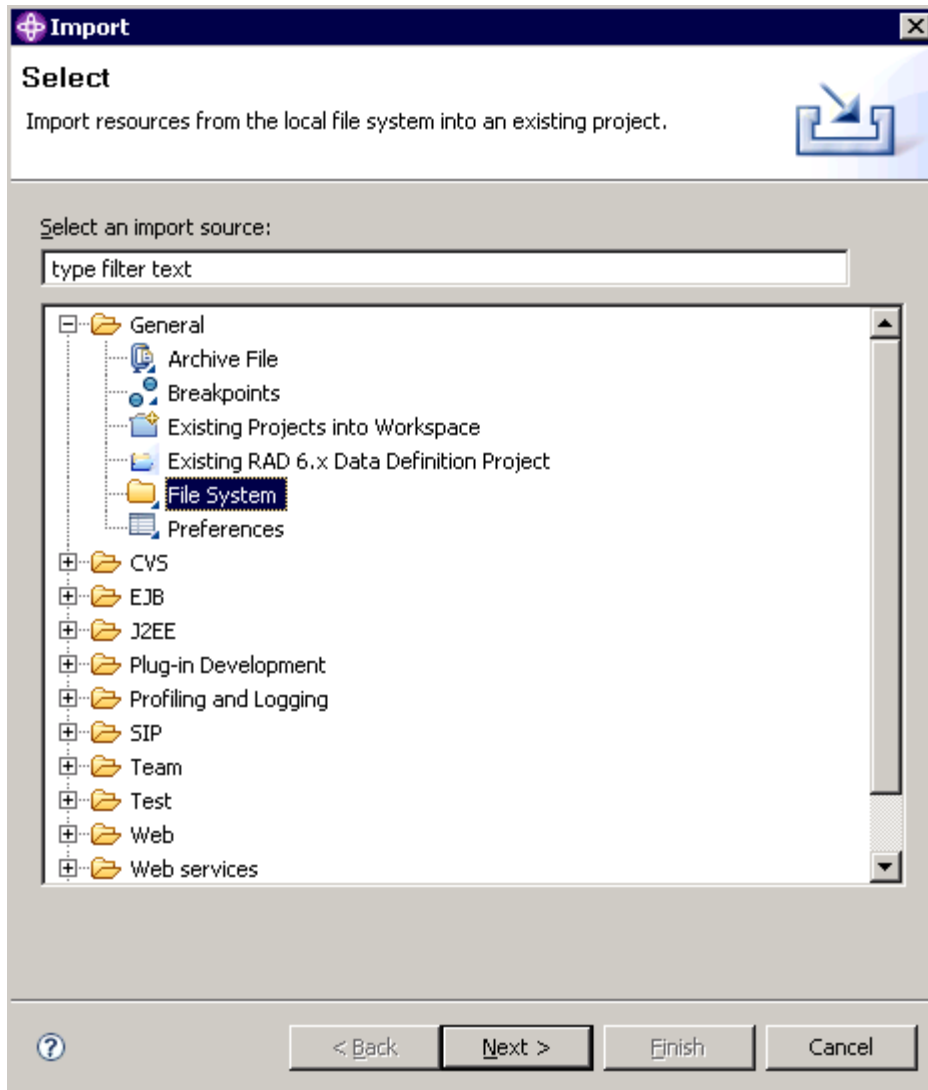


__ e. Enter a folder name of **wsdl** and click **Finish**.

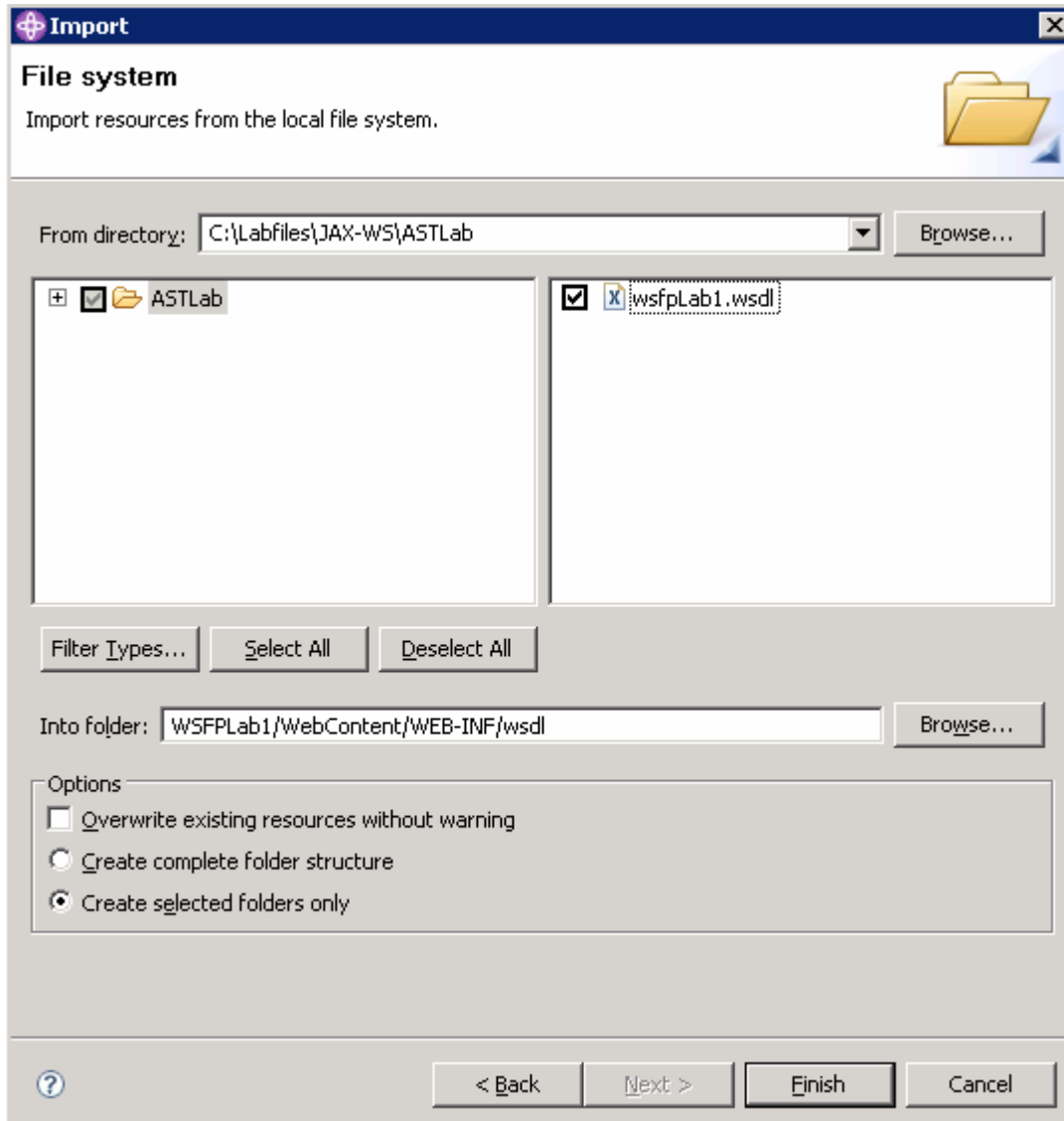


___ f. Right click on the **wSDL** folder you just created and select **Import**.

___ g. From the **Import** panel select **File system** and click **Next**.



- ___ h. On the **File system** panel select **Browse**. Navigate to the **C:\Labfiles\JAX-WS\ASTLab** directory.
- ___ i. Select the checkbox on the left next to **ASTLab**. Then on the right make sure to select the checkbox for the **wsfpLab1.wsdl**. Click **Finish**.



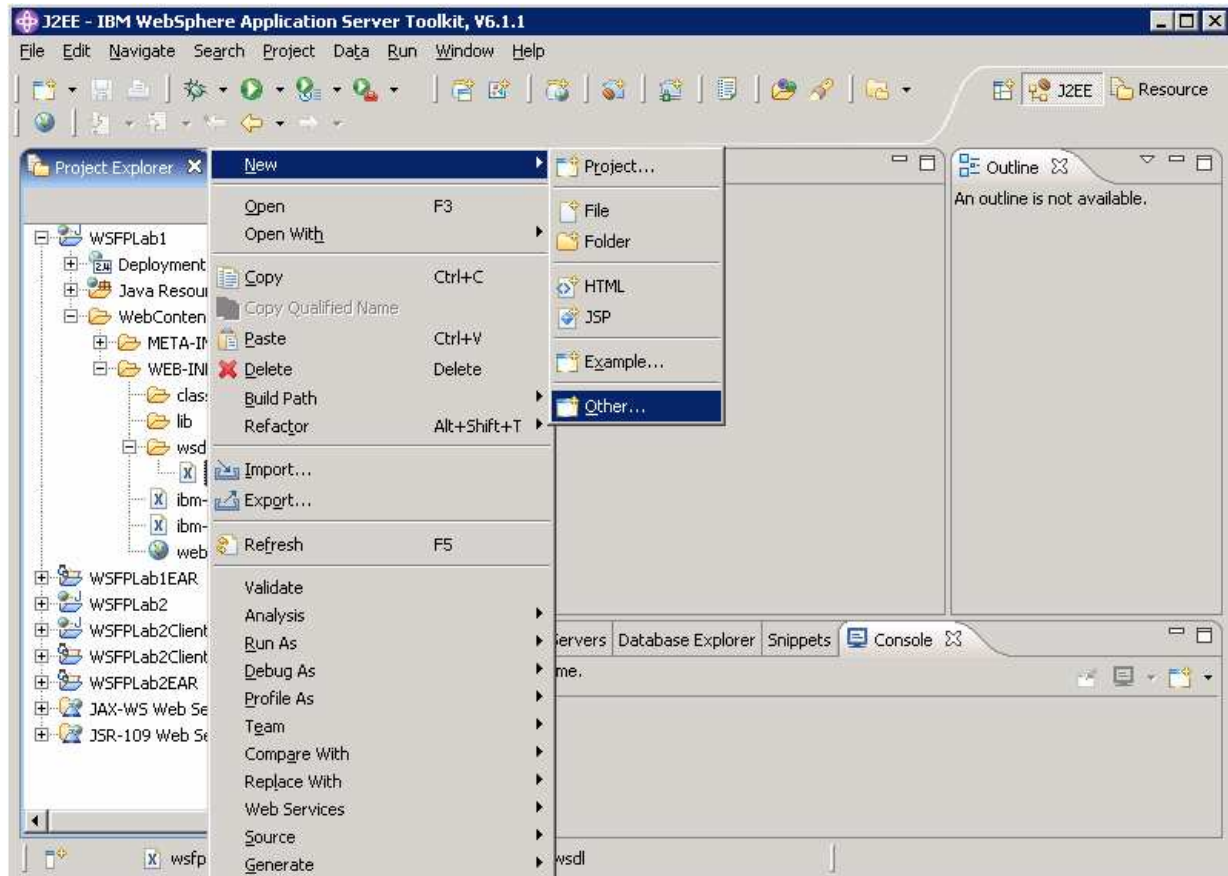
___ j. The WSDL document has now been added to the project. This is the WSDL that describes the Web Service that will be created, as well as how to access it. Some of the main WSDL elements are:

- 1) The <service> element contains the name of the Web Service and its address.
- 2) The <portType> element describes a Web Service. It specifies the operations that can be performed as well as the input and output messages for each operation.
- 3) The <message> element describes the data that are used by an operation. Each message can consist of one or more parts. The parts are similar to parameters in a function call.
- 4) The <types> element defines the data types that are used by the Web Service.
- 5) The <binding> element defines the message format and protocol details for each port.

Part 3: Creating a Web service from the WSDL file

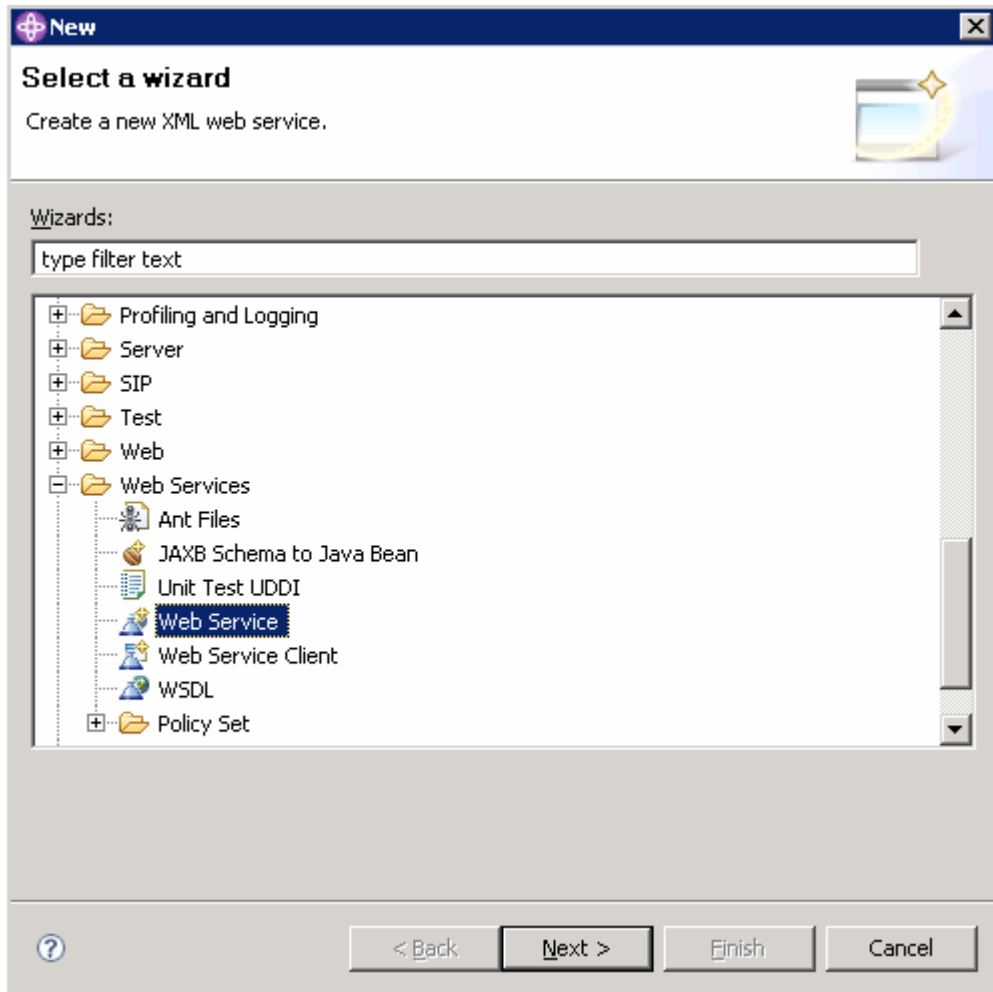
This section will describe the process creating a Web Service from the WSDL file that was imported in the previous step.

- ___ 1. Use the WSDL to create a Web Service.
 - ___ a. Expand the **wSDL** folder in the project explorer.
 - ___ b. Right click on the wsfpLab1.wsdl document and select **New** and then select **Other**.

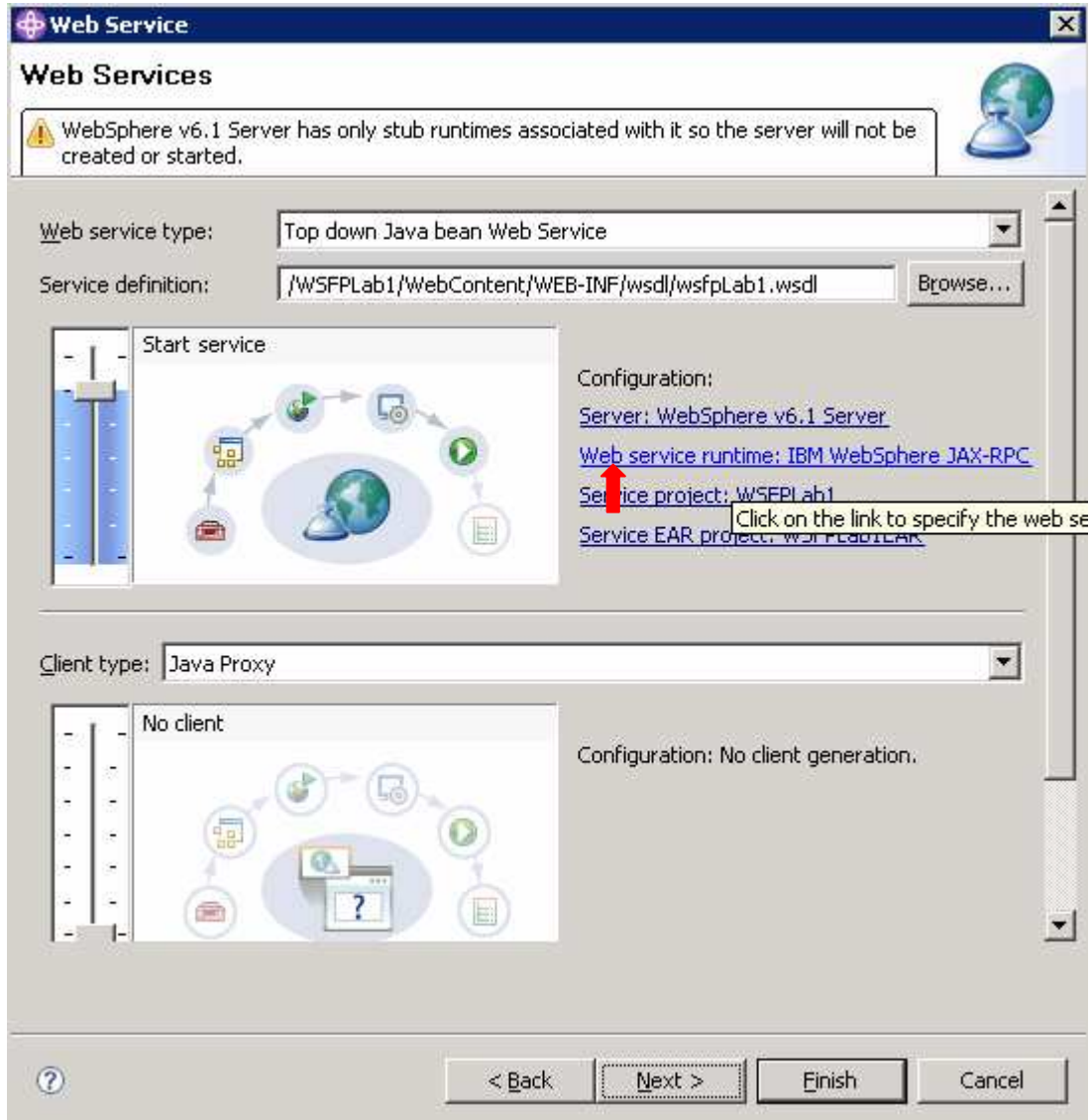


__ c. In the New window expand the Web Services folder and select **Web Service**.

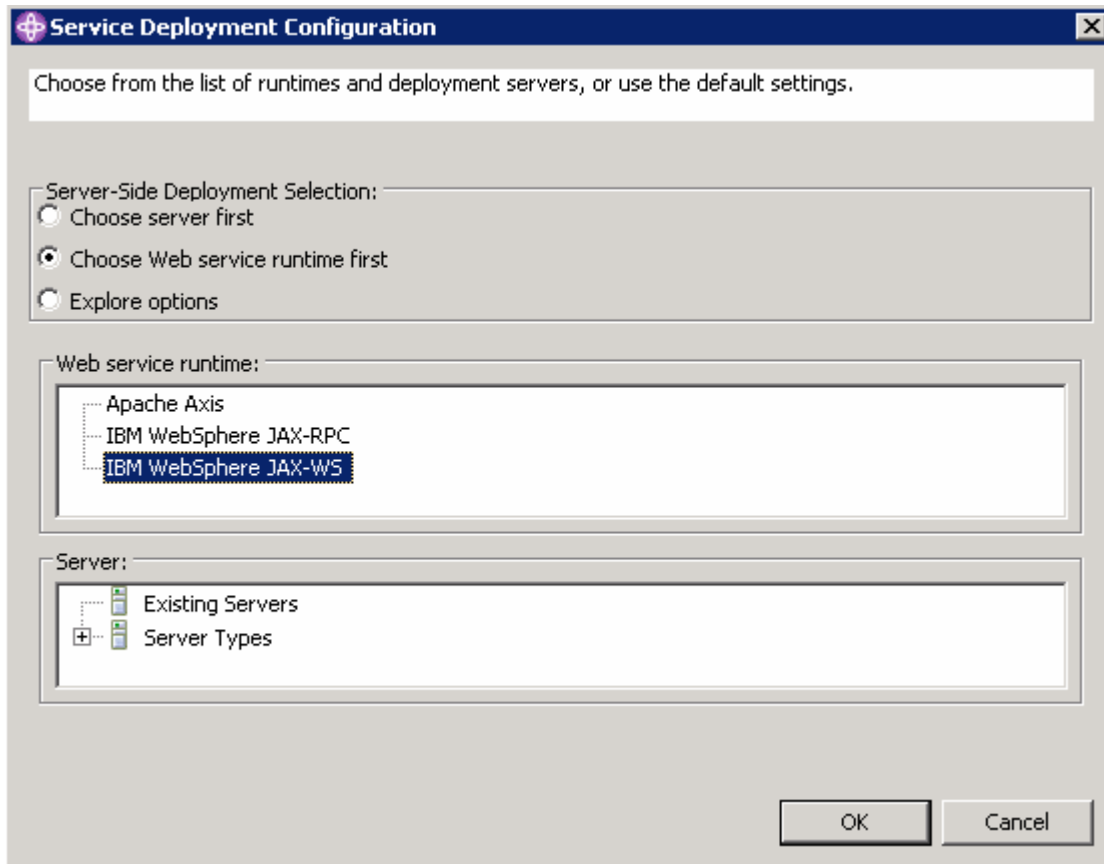
__ d. Select **Next**.



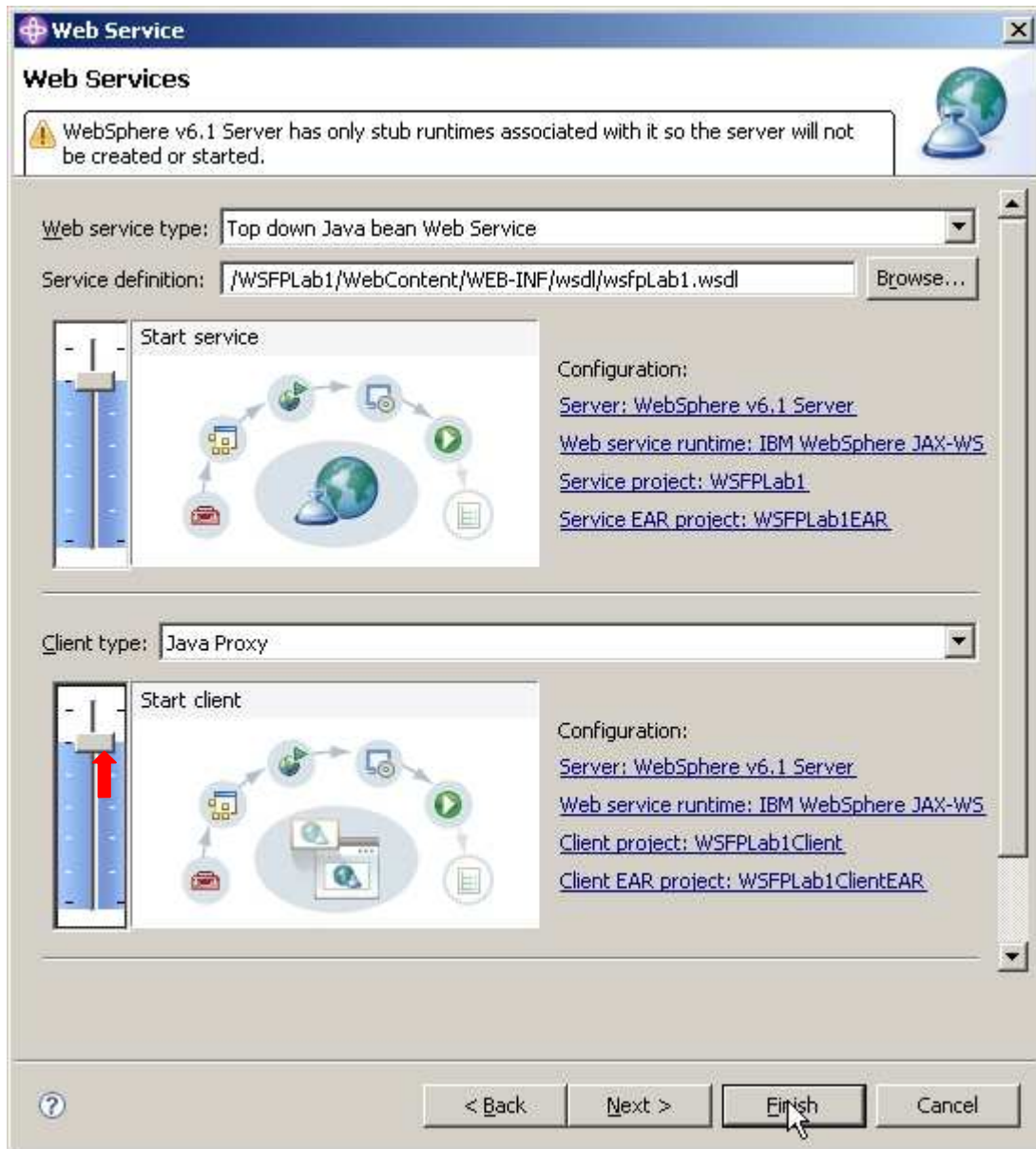
- e. Change the **Web service runtime**. Click on **Web service runtime: IBM WebSphere JAX-RPC** to change the choice of runtimes.



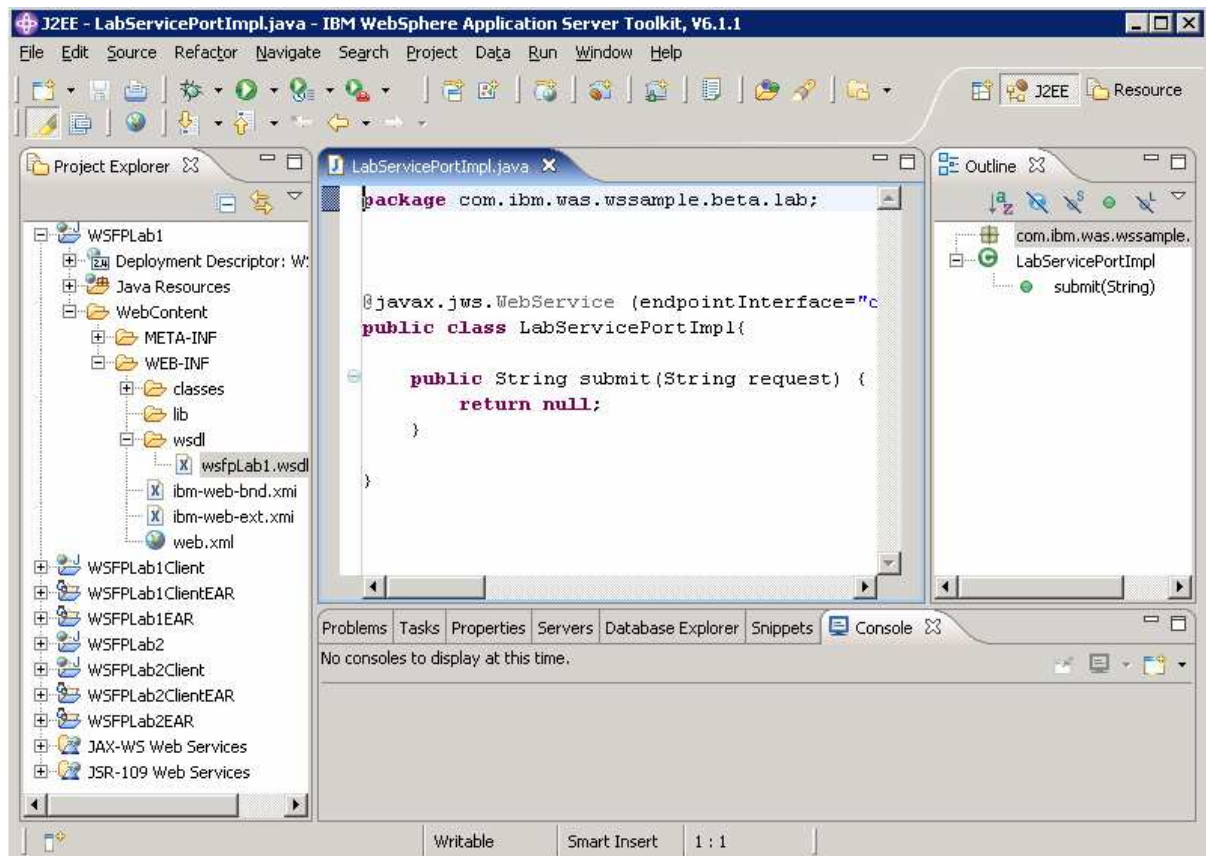
- ___ f. Select the **IBM WebSphere JAX-WS** runtime option, and click **OK**. This will insure that the Web Service created will comply with the JAX-WS specification.



___ g. In the same panel, position the slider bar for the client so that it shows “Start client”.



- ___ h. Select **Finish**. The AST will now create the implementation classes for the service described by the WSDL document. It will open the implementation class for the Web Service, which in this case is named **LabServicePortImpl.java**.



NOTE: Annotations are required for **JAX-WS** services, they have already been added by the tools; these can be found in the code at the `@javax.jws.WebService` line. The annotations identify each service, including its implementation class and the location of its WSDL document. They optionally identify any **Quality of Service** modules (e.g. **WS-Addressing**) to be engaged on the server. It is very important that these annotations be added. If the `@WebService` is not added, the Web Service will not deploy as a **JAX-WS** service. Additionally, the `serviceName` attribute defines what the service is called.

__ i. Change the following line to the code in LabServicePortImpl.java.

1) Change:

```
public class LabServicePortTypeImpl{  
    public String submit(String request) {  
        return null;  
    }  
}
```

2) To:

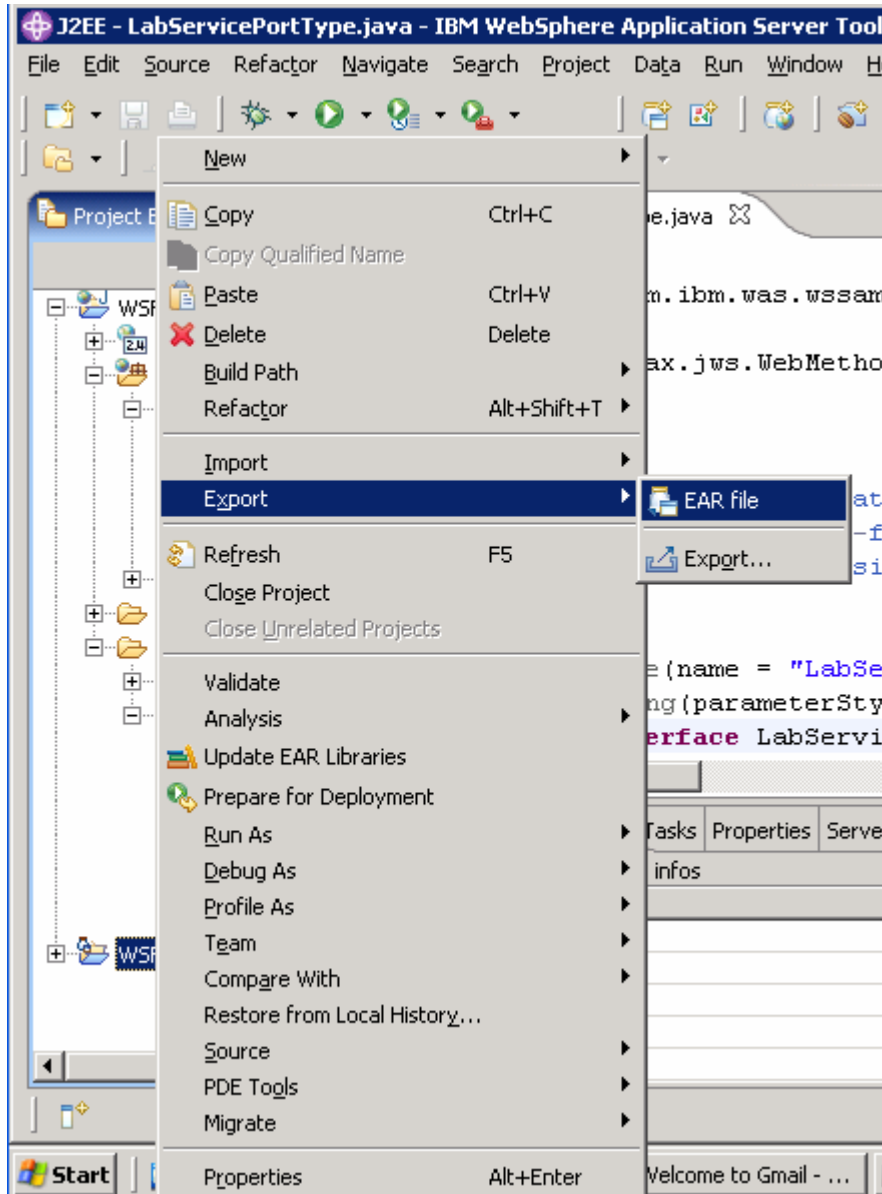
```
public class LabServicePortTypeImpl{  
    public String submit(String request) {  
        return ("Hello " + request);  
    }  
}
```

__ j. Save your changes with **CTRL + S**.

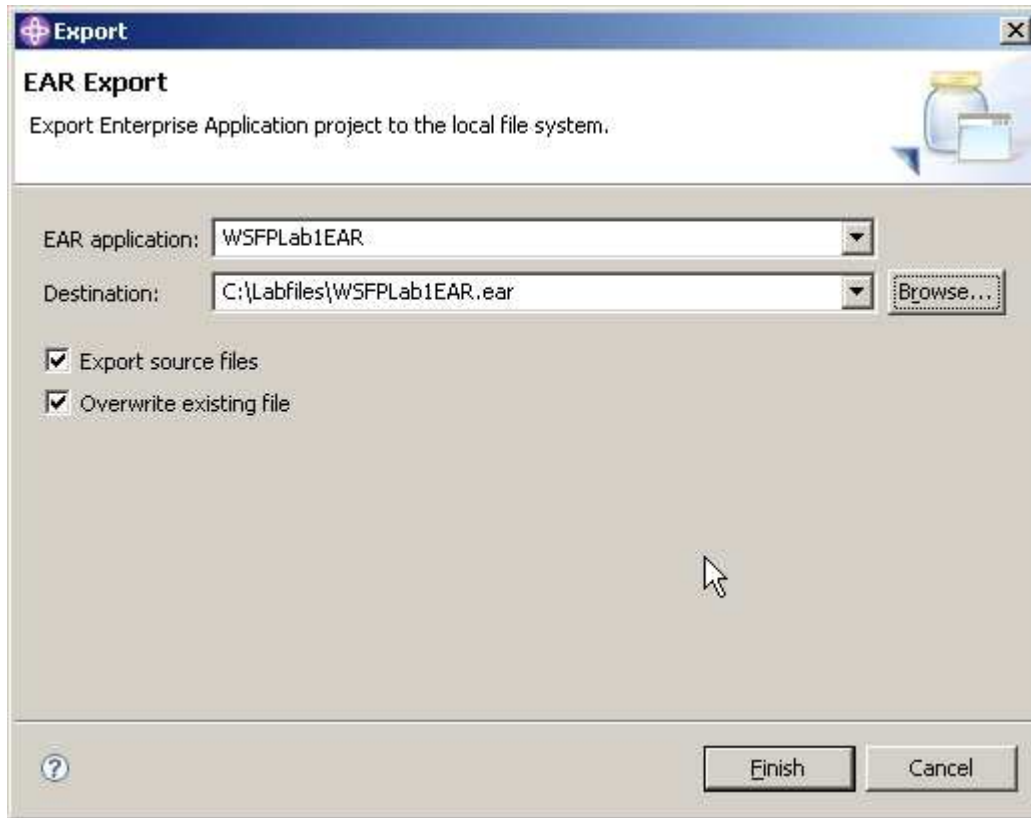
___ 5. Exporting the EAR file.

___ a. In the Project Explorer **right-click** on the WSFPLab1EAR folder to open the menu.

___ b. Select **Export** and the **EAR file**.



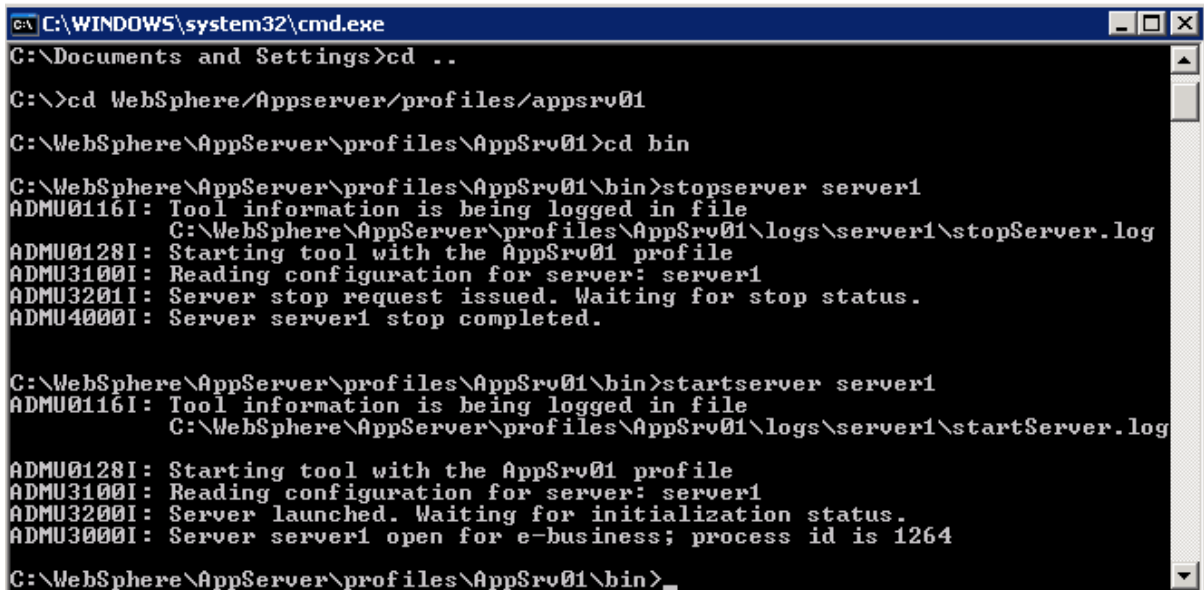
- ___ c. Select **WSFPLab1EAR** for the EAR application. For the Destination select **C:\LabFiles**. Check the boxes to **Export source files the source** code and **Overwrite existing file**. Click **Finish**.



Part 4: Install and deploy the application

This section describes how to install and deploy the EAR file that has been created to an installation of WebSphere Application Server Version 6.1 with the Feature Pack for Web Services installed. This will use the profile that was created in Part 1 of this lab.

- ___ 1. **Start the application server.**
 - ___ a. Open a command prompt and change to **C:\WebSphere\AppServer\profiles\AppSrv01\bin**.
 - ___ b. Run the command **startserver server1**.



```

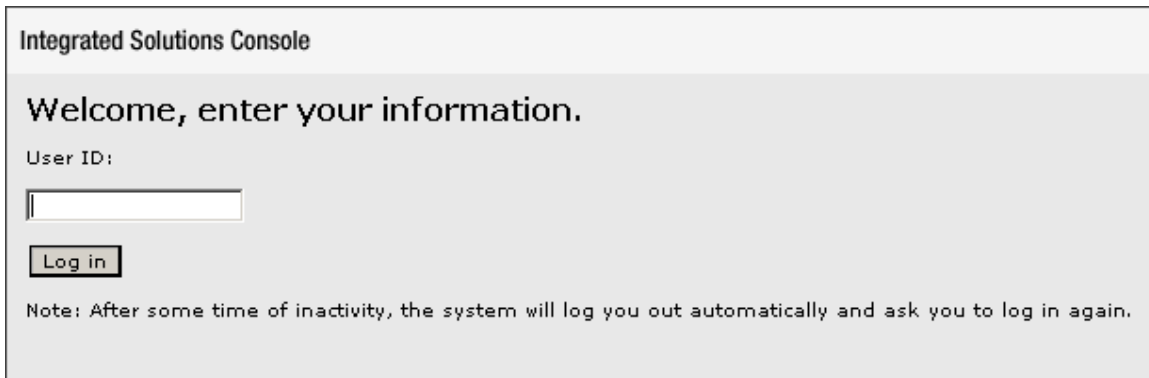
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings>cd ..
C:\>cd WebSphere/Appserver/profiles/appsrv01
C:\WebSphere\AppServer\profiles\AppSrv01>cd bin
C:\WebSphere\AppServer\profiles\AppSrv01\bin>stopserver server1
ADMU0116I: Tool information is being logged in file
           C:\WebSphere\AppServer\profiles\AppSrv01\logs\server1\stopServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

C:\WebSphere\AppServer\profiles\AppSrv01\bin>startserver server1
ADMU0116I: Tool information is being logged in file
           C:\WebSphere\AppServer\profiles\AppSrv01\logs\server1\startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 1264
C:\WebSphere\AppServer\profiles\AppSrv01\bin>
  
```

- ___ c. Wait for the server to start.
- ___ 2. **Start the Integrated Solution Console (ISC).** There are two ways to start the ISC.
 - ___ a. Using the Start Menu **Windows** :
 Click on Start->Programs->IBM WebSphere->Application Server v6.1->Profiles->wslab->Administrative Console
 - ___ b. Using a browser. Start the browser and enter the following in the address window::

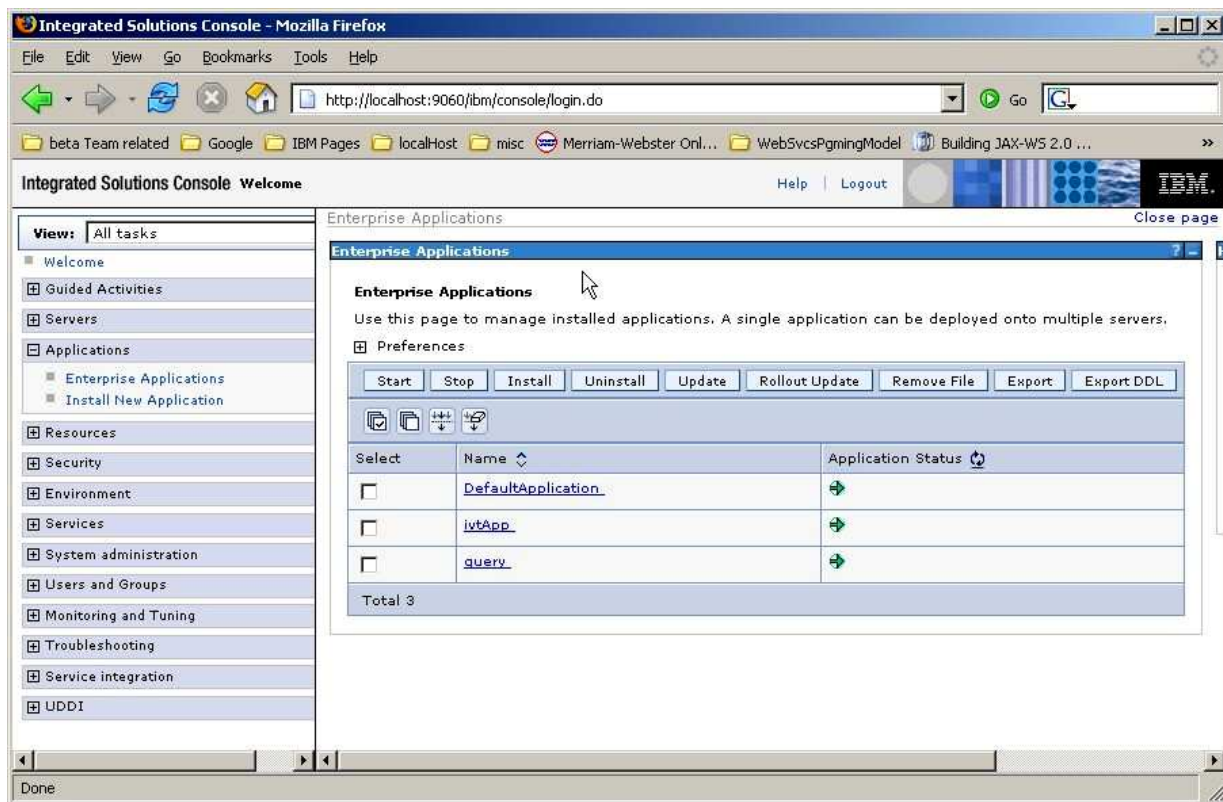
<http://localhost:9060/ibm/console>

___ 3. Log in to the ISC as wsdemo.

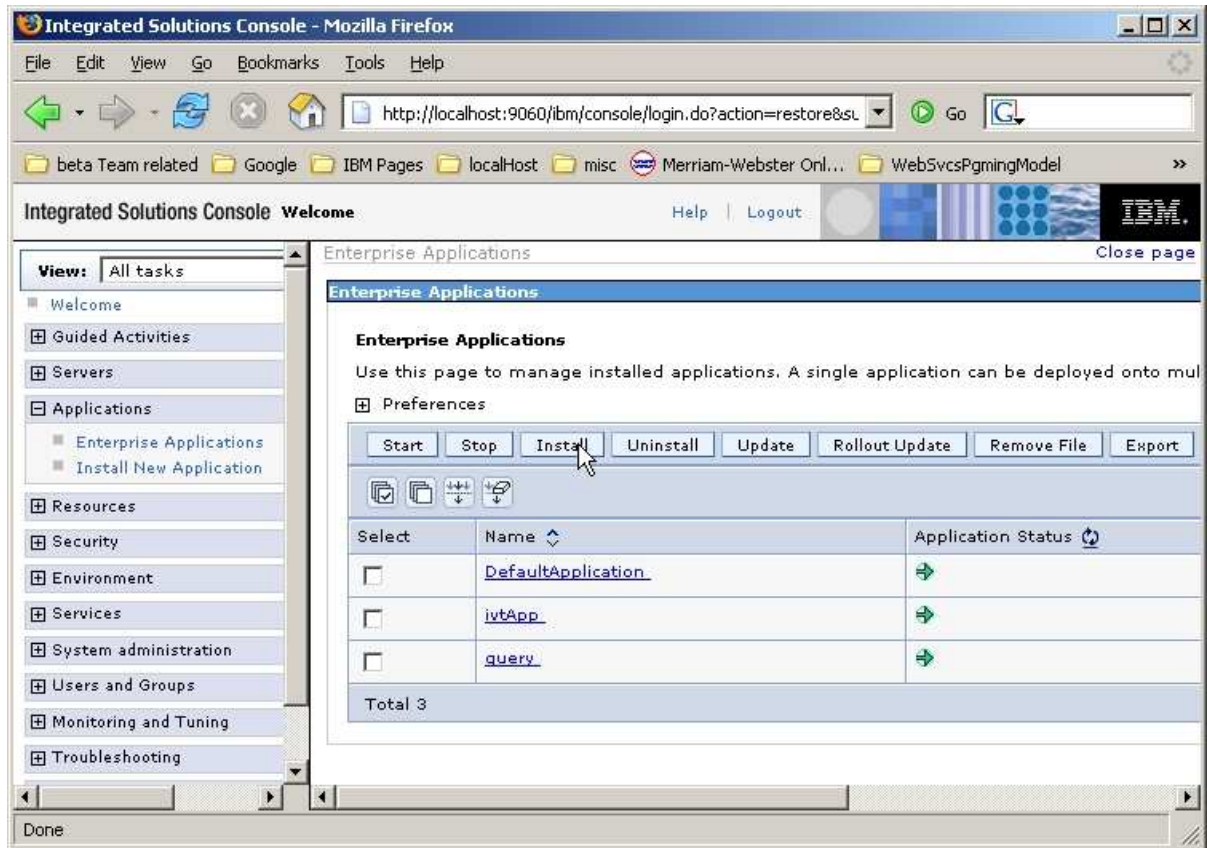


___ 4. Install theWSFPLab1EAR file.

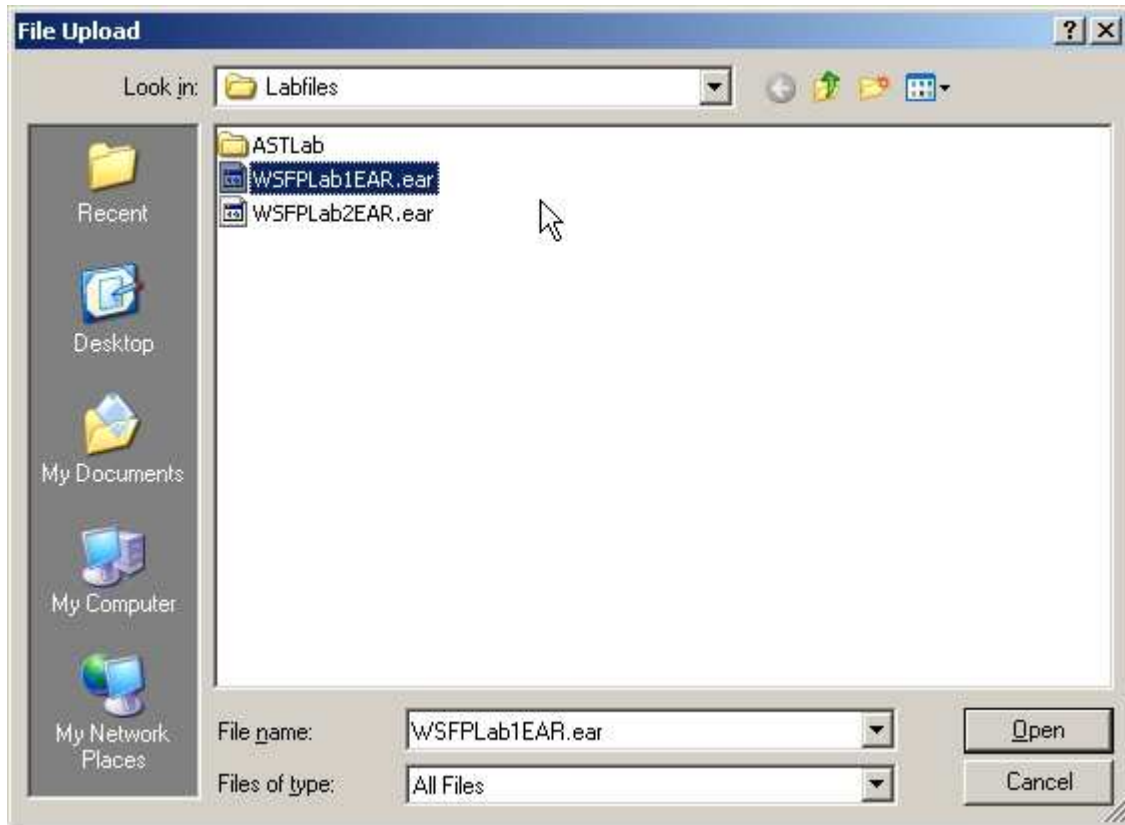
__ a. Expand **Applications** and click on **Enterprise Applications**. The ISC will display a screen similar to the one shown below.



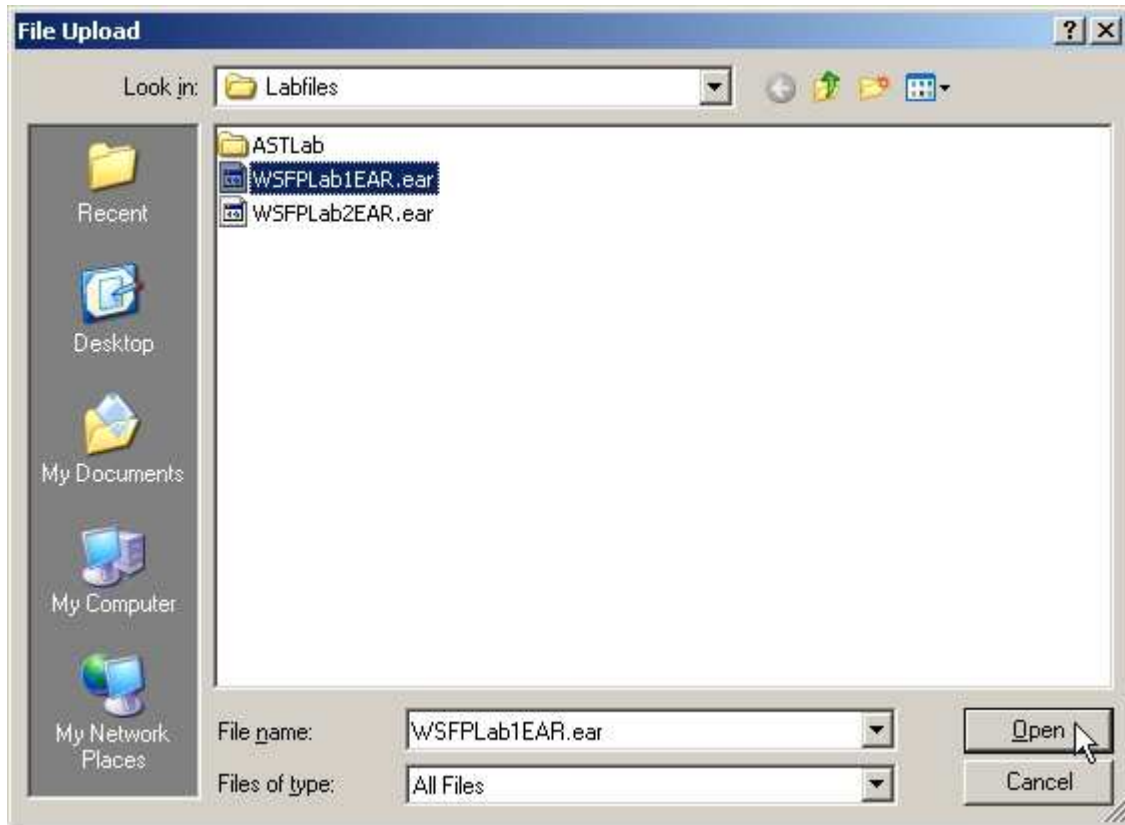
__ b. In the Enterprise Applications panel, select the **Install** button.



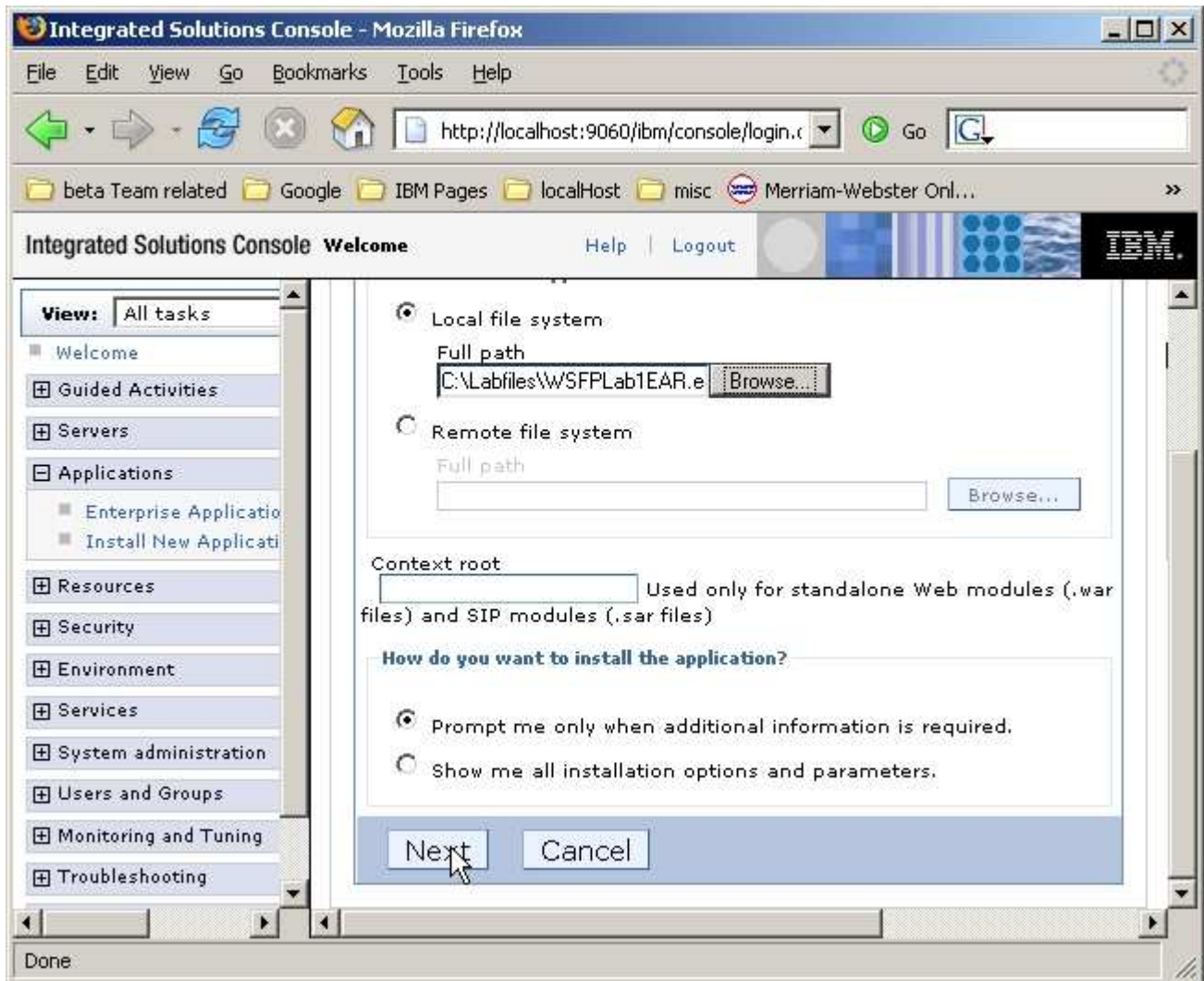
- c. Click **Browse** and browse to **C:\LabFiles** where you previously exported the EAR for this service.



__ d. Select **WSFPLab1EAR.ear** and Click **Open**.



__ e. Click **Next**.



__ f. On the **Select installation options** panel click **Next**.

__ g. On the **Map modules to servers** panel click **Next**.

__ h. On the **Summary** panel click **Finish**.

Application WSFPLab1EAR installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

To work with installed applications, click the "Manage Applications" button.

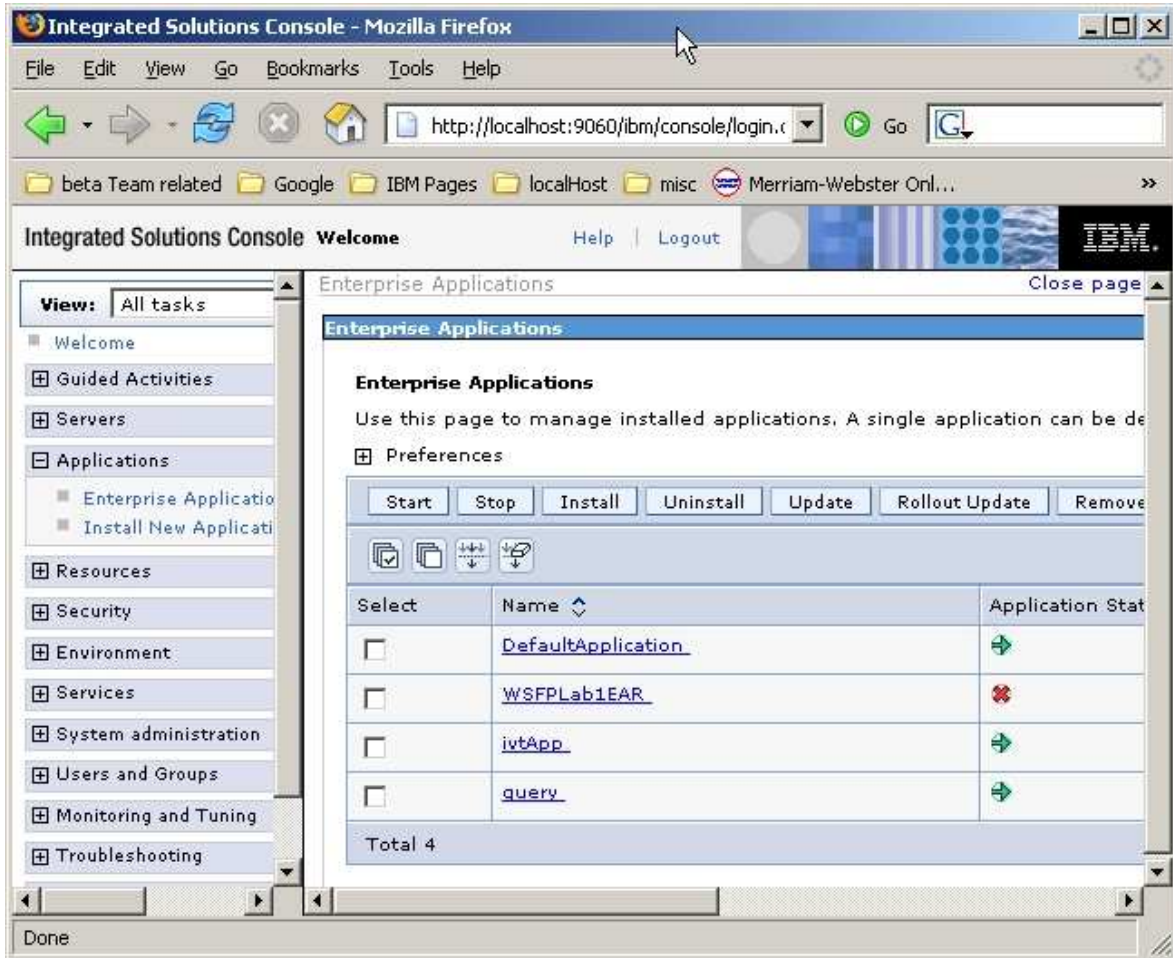
[Manage Applications](#)

__ i. Wait to see that the application installs successfully. The messages shown above will be displayed. Click **Save** to save the changes to the master configuration.

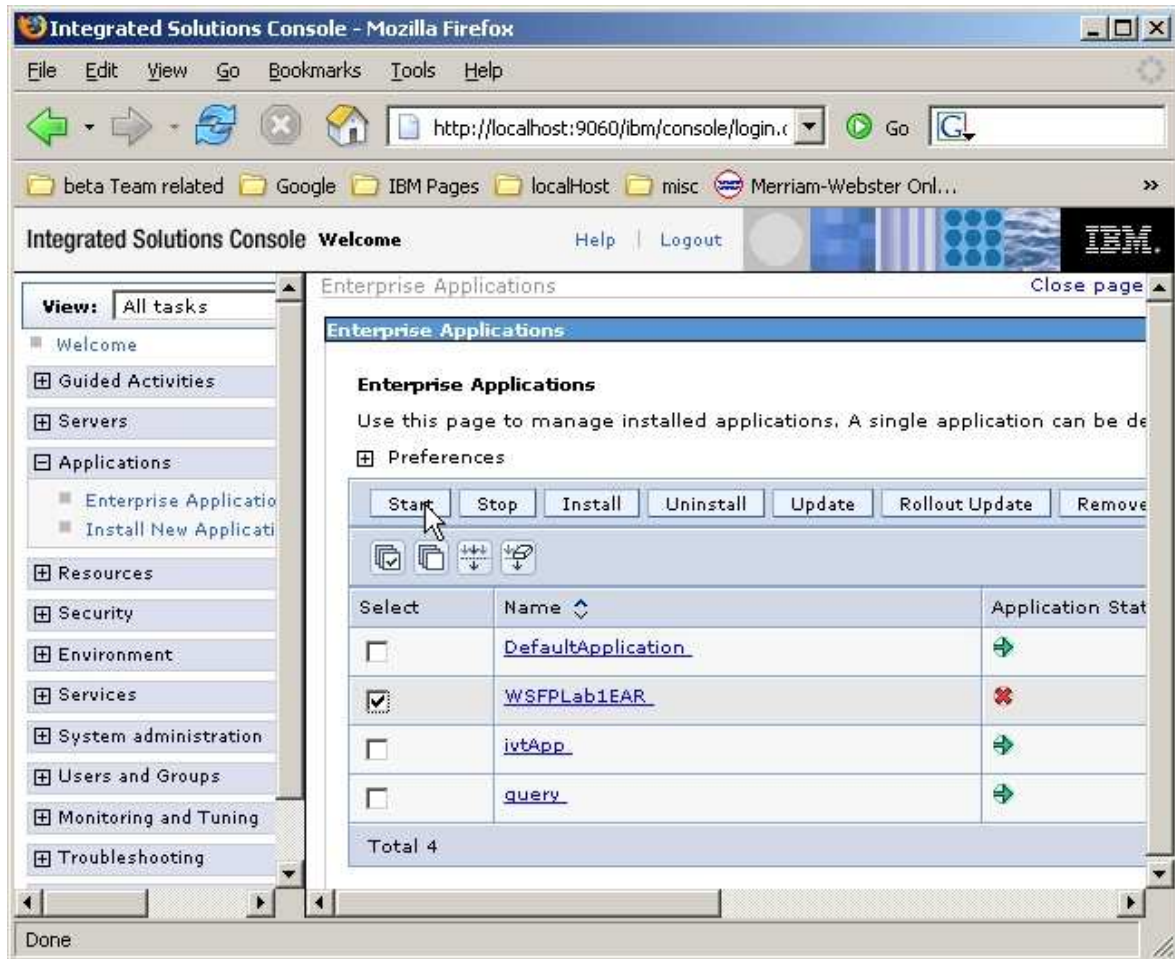
__ j. The WSFPLab1EAR application has now been deployed.

5. Starting the application.

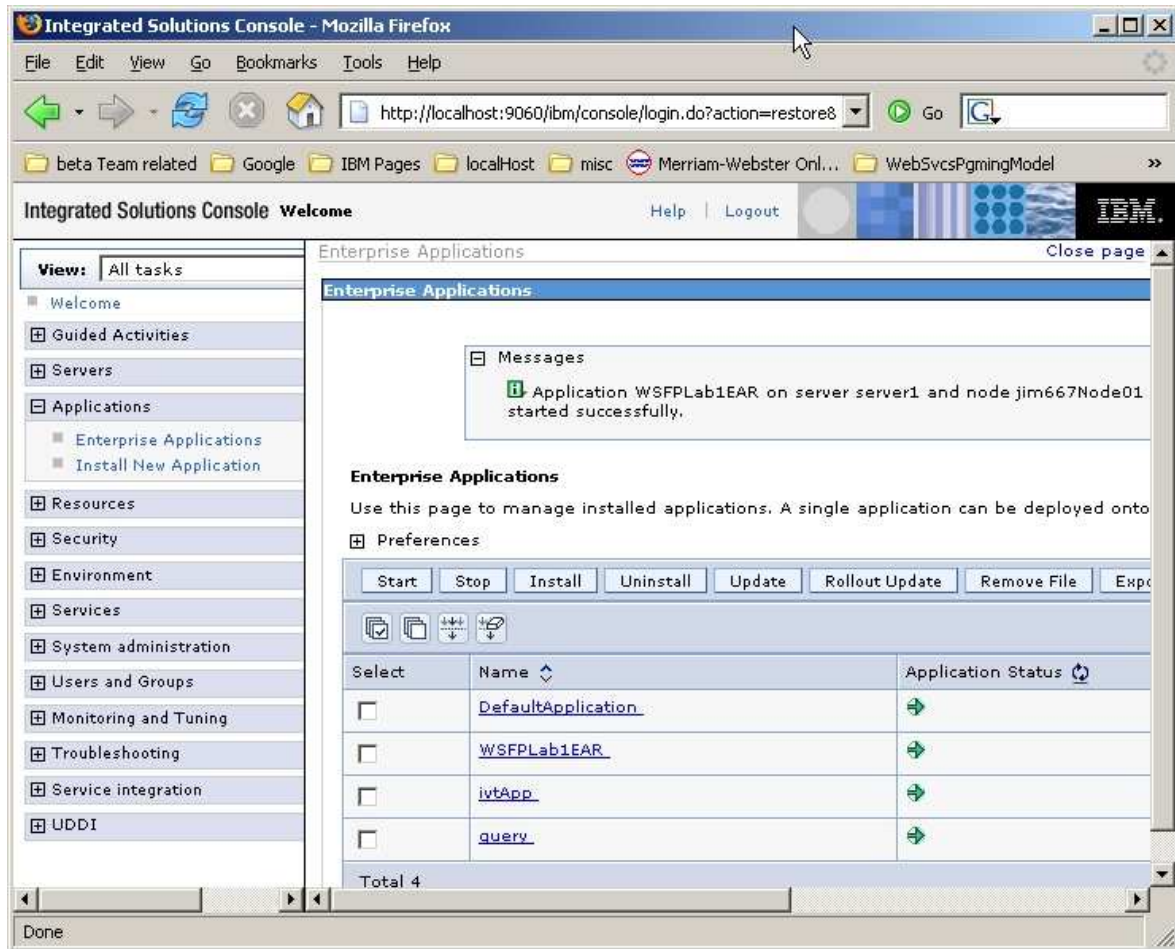
a. Return to the **Enterprise Applications** panel. The ISC will display a screen similar to the one shown below.



__ b. Check the box next to **WSFPLab1EAR** and click **Start**.



__ c. Wait for the application to start. The following page will be displayed.



___ 6. Test the Web Service.

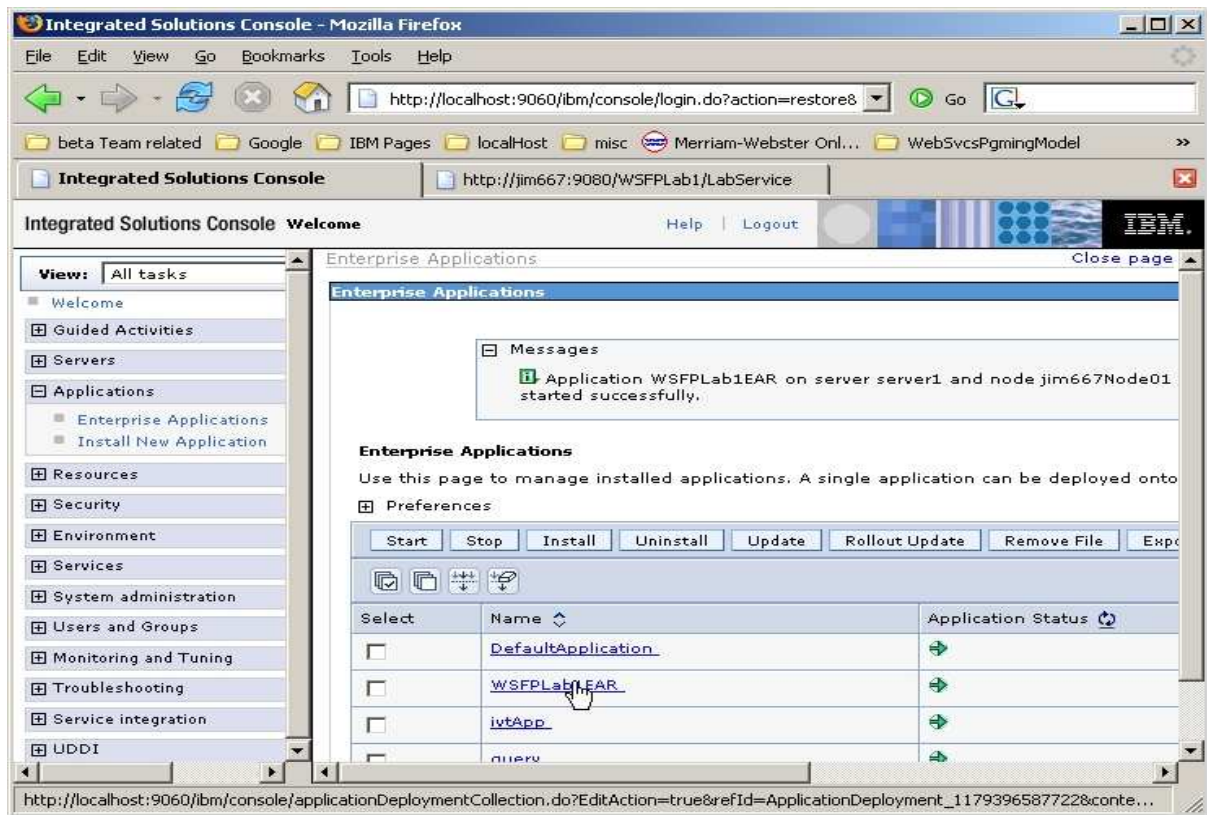
- ___ a. To verify that the Web service was deployed and started properly, a browser will be given the URL for the Web Service and it should display a message from the application server indicating that this is a Web service.

The default URL for the Web service can be built by concatenating the following:

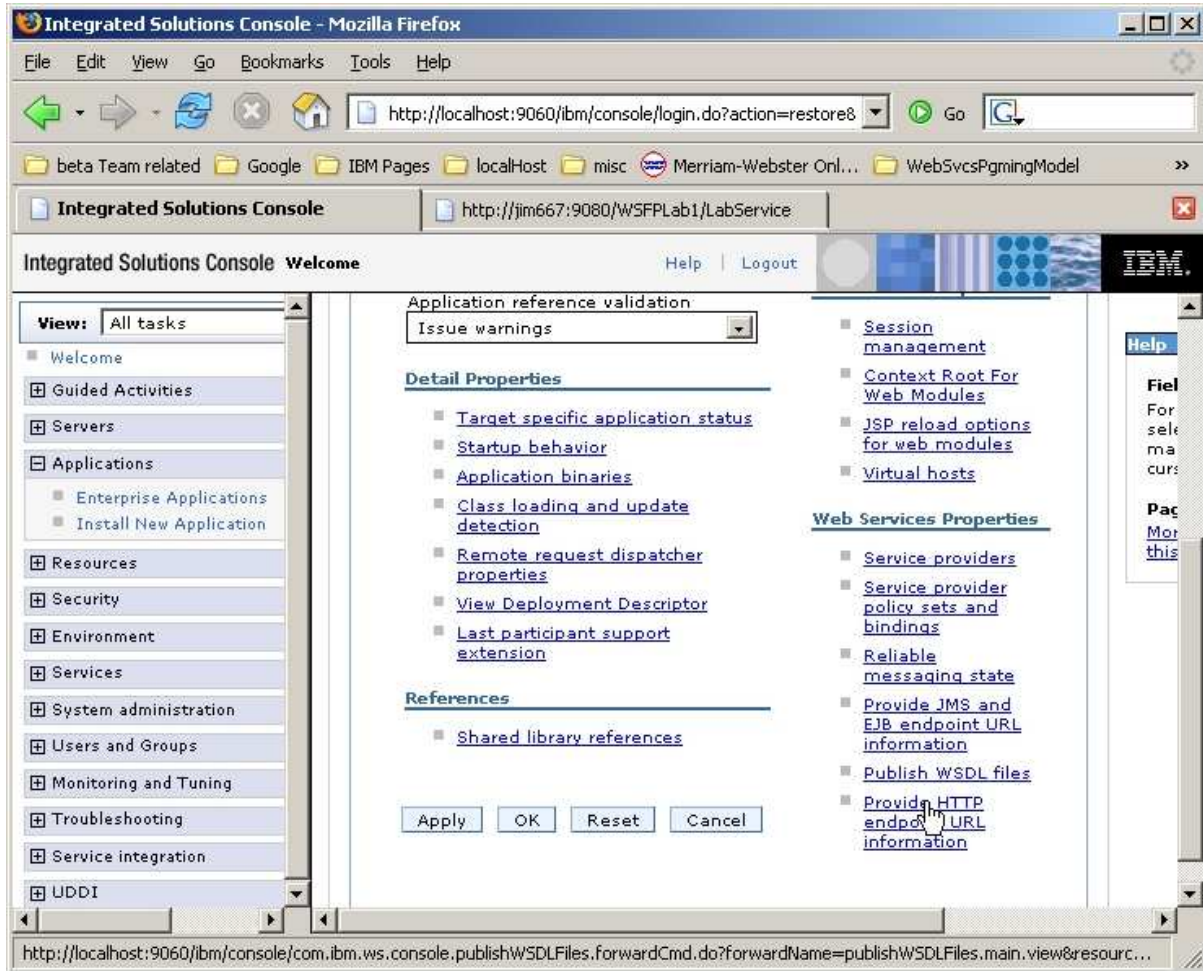
<HTTP URL Prefix>/WSFPLab1/LabService

The next steps show how to find this information and create this URL.

- ___ b. The next steps show how to find this information and create this URL.to find the <HTTP URL Prefix>, in the **ISC** navigate to the **Enterprise Applications** panel. Select **WSFPLab1EAR**.



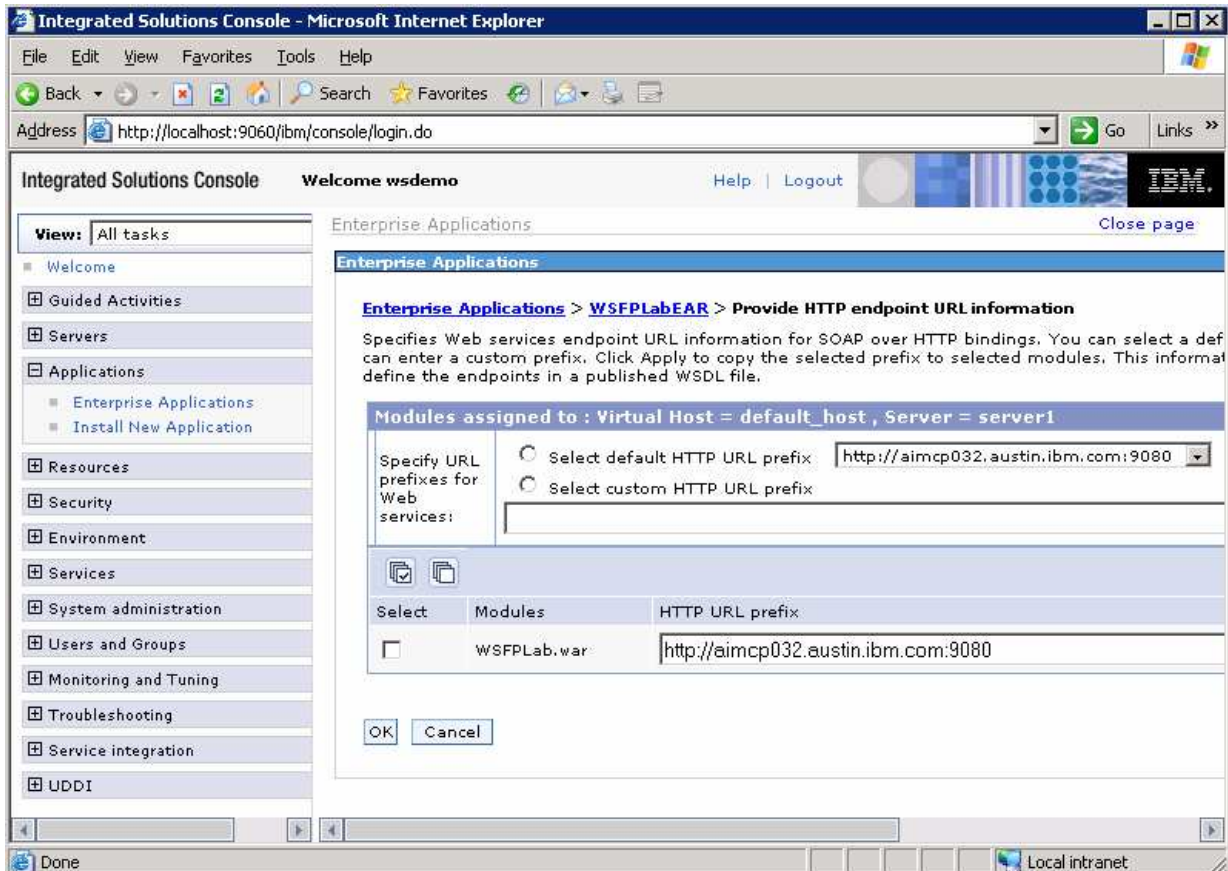
__ c. Select Provide HTTP endpoint URL information.



- d. The value displayed in the window labeled **HTTP URL prefix** is what is needed for **<HTTP URL Prefix>**. In this case it is:

http://aimcp032.austin.ibm.com:9080

This URL will be different on other systems!



- ___ e. Putting the URL prefix together with **WSFPLab1/LabService** results in the URL for the Web Service. So with the above prefix the test URL would be:

http://aimcp032.austin.ibm.com:9080/WSFPLab1/LabService

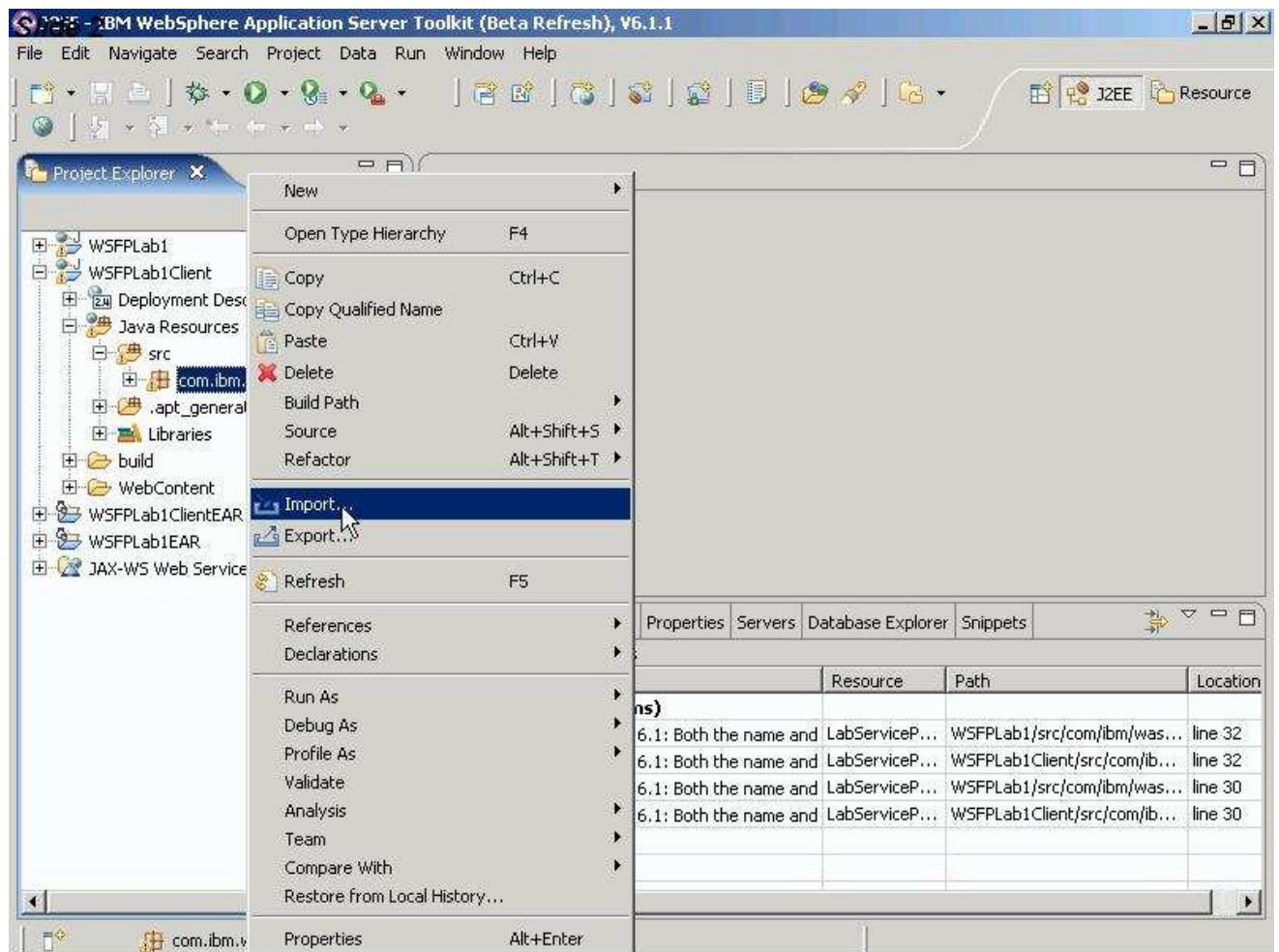
- ___ f. Open a new browser window and paste in the URL for your system. You should see a screen like that below.



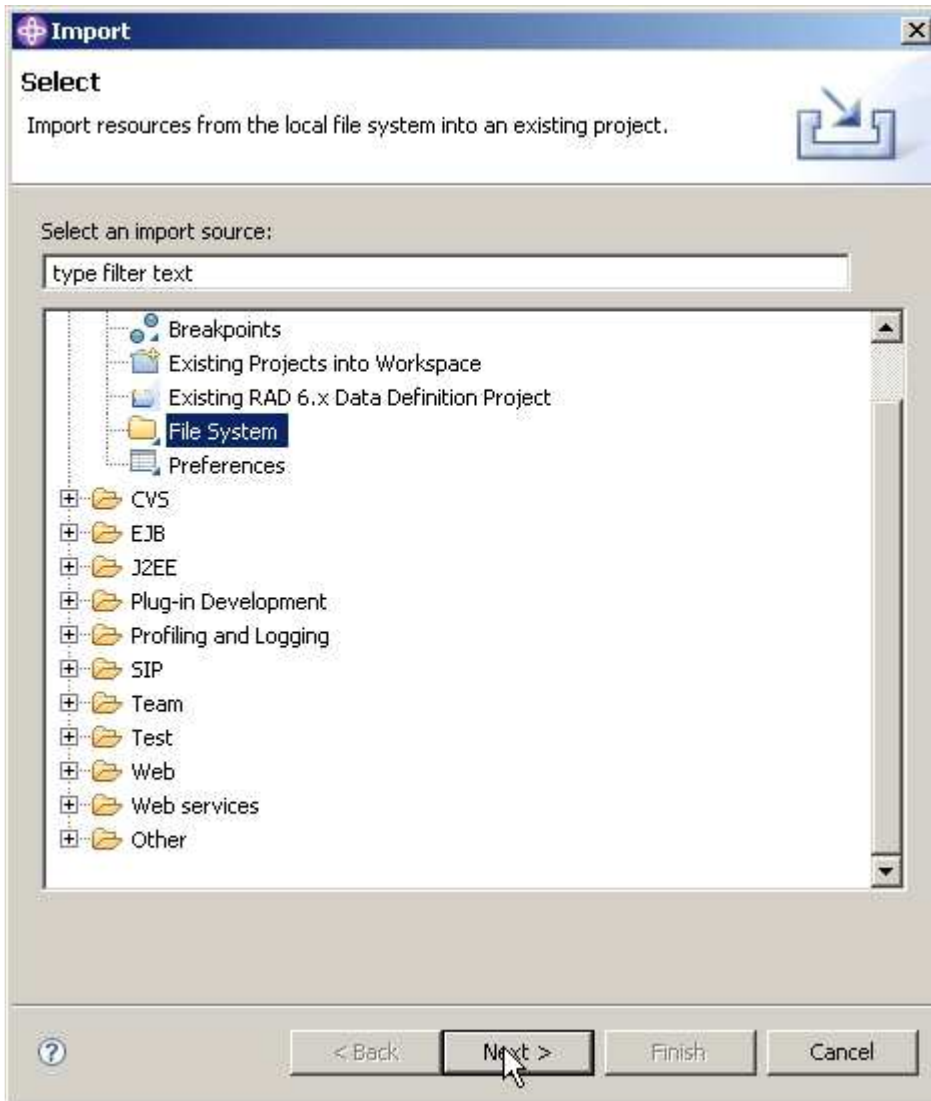
Part 5: Creating the client code

In this section of the lab, a Java Web services client will be created to interact with the Web Service that has been created, deployed, and started.

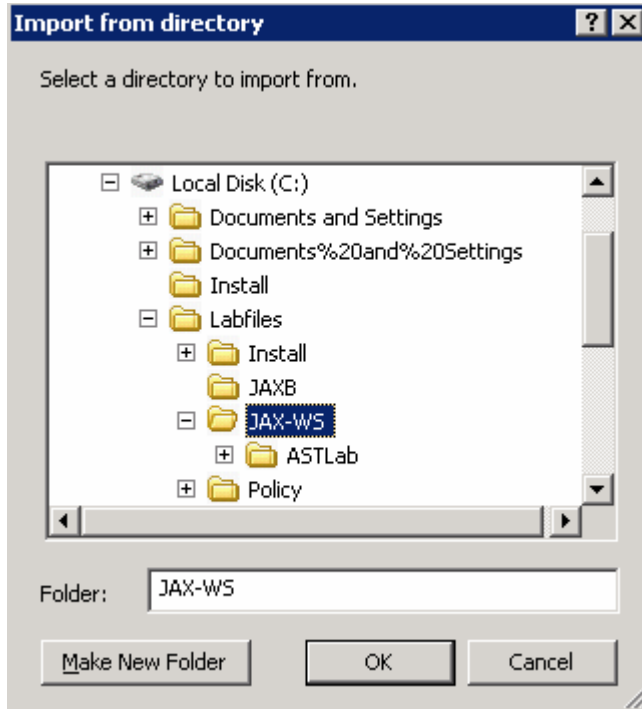
- ___ 7. Import the client implementation code.
 - ___ a. In the AST **Project Explorer** panel, expand **WSFPLab1Client**. This is the client project that was created by **AST** at the same time the Web Service was created. It contains the client side artifacts and files but does not implementation code. In the following steps you will import a template for the implementation code for a simple client and will make the changes needed for it to run against the Web service that you have installed.
 - ___ b. Expand **Java Resources** and then expand **src**. **Right click** on the package named **com.ibm.was.wssample.lab** and then click on **Import**.



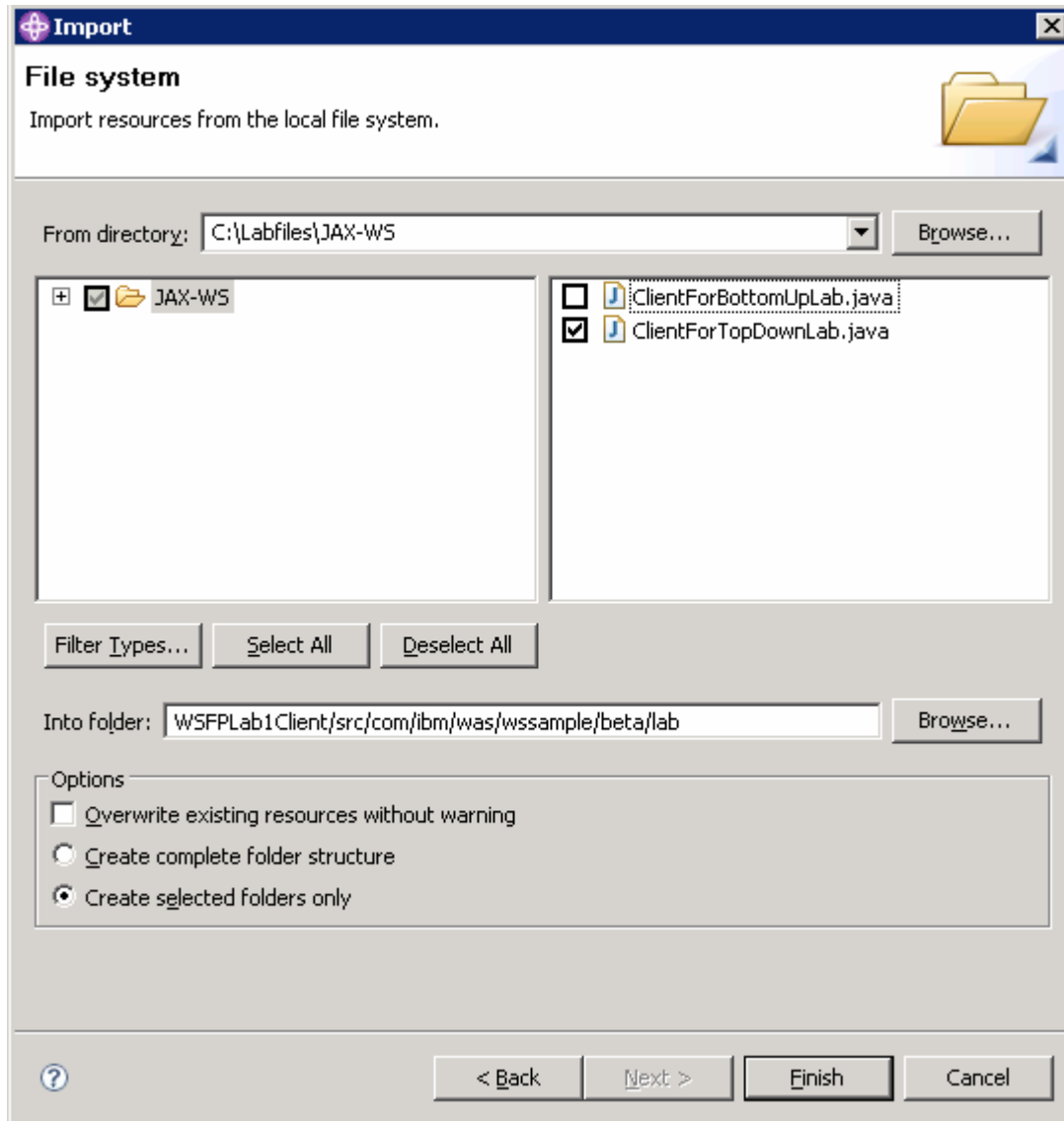
___ c. On the **Import** panel select File System and click **Next**.



__ d. Navigate to the **C:\Labfiles\JAX-WS** directory and click **OK**.



- ___ e. Select the checkbox on the left next to **simple**. Then on the right make sure to select only the checkbox for the **ClientForTopDownLab.java**. Click **Finish**.



- ___ 8. Modify the client code - **ClientForTopDownLab.java** is built from a template for a JAX-WS stand-alone client. It must be modified so that it will work for this lab.

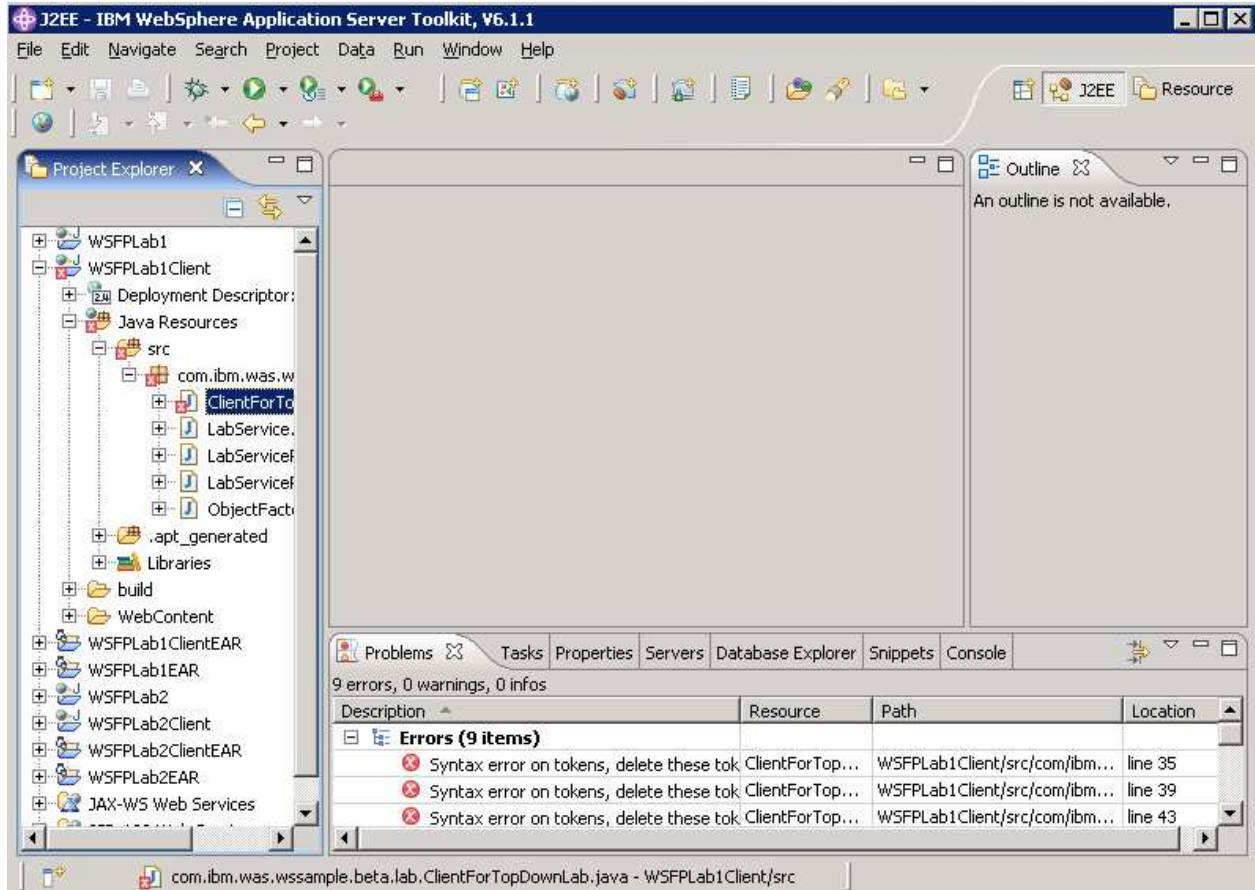
The lines that will require changes are preceded by one or more lines of the form:

//! Change - <instructions>

You will need to read the instructions and make the appropriate modifications.

Until you make the changes needed, the file will fail compilation – this is deliberate. Reference the wsdl that is in your browser window from the earlier step to identify the values to use for some of the changes you will need to make.

- ___ a. Some of the statements in ClientForTopDownLab.java must be changed based on your environment. In the AST, expand the package **com.ibm.was.wssample.beta.lab**. Double click on the **ClientForTopDownLab.java** file contained within, to edit it.



___ 9. Change #1

Change from:

```

!!! Change - The package name must agree with the package name where this
!!! Change - file is placed in AST
package ???;

```

To:

```

!!! Change - The package name must agree with the package name where this
!!! Change - file is placed in AST
package com.ibm.was.wssample.beta.lab;

```

___ 10. Change #2

Change from:

```

!!! Change - The package name must agree with the package name where this
!!! Change - file is placed in AST but use "/" as delimiter instead of "."
    private static final String PACKAGE_NAME = ???;

!!! Change - The value of the name attribute from
!!! Change - the <wsdl:service ... > element
    private static final String SERVICE_NAME = ???;

!!! Change - The value of the "name" attribute from
!!! Change - the <wsdl:port ... > element
    private static final String PORT_NAME = ???;

```

To:

```

!!! Change - The package name must agree with the package name where this
!!! Change - file is placed in AST but use "/" as delimiter instead of "."
    private static final String PACKAGE_NAME = "com.ibm.was.wssample.beta.lab";

!!! Change - The value of the name attribute from
!!! Change - the <wsdl:service ... > element
    private static final String SERVICE_NAME = "LabService";

!!! Change - The value of the "name" attribute from
!!! Change - the <wsdl:port ... > element
    private static final String PORT_NAME = "LabServicePort";

```

11. Change #3

Change from:

```

//! Change - the URL to use to contact the Web Service
    private static final String
        URL_ENDPOINT = ???;

```

To (Use the endpoint appropriate for your system):

```

//! Change - the URL to use to contact the Web Service
    private static final String
        URL_ENDPOINT = "http://jim667:9080/WSFPLab1/LabService";

```

12. Change #4

Change from:

```

//! Change - Next real line is really like:
//! Change - <wsdl:portType.name> portTypeName =
//! Change -         service.getPort(QNAME_PORT, <wsdl:portType.name>.class);
    ??? portTypeName = service.getPort(QNAME_PORT, ??? .CLASS);

```

To:

```

//! Change - Next real line is really like:
//! Change - <wsdl:portType.name> portTypeName =
//! Change -         service.getPort(QNAME_PORT, <wsdl:portType.name>.class);
    LabServicePortType portTypeName = service.getPort(QNAME_PORT,
LabServicePortType.class);

```

13. Change #5

Change from:

```

//! Change - this line so that it is:
//! Change - String replyString = portTypeName.<operationName>(args);
//! Change - where you need to change <operationName> to the operation to invoke
//! Change - and "args" will be arguments to that operation
    String replyString = portTypeName.???;

```

14. To::

```

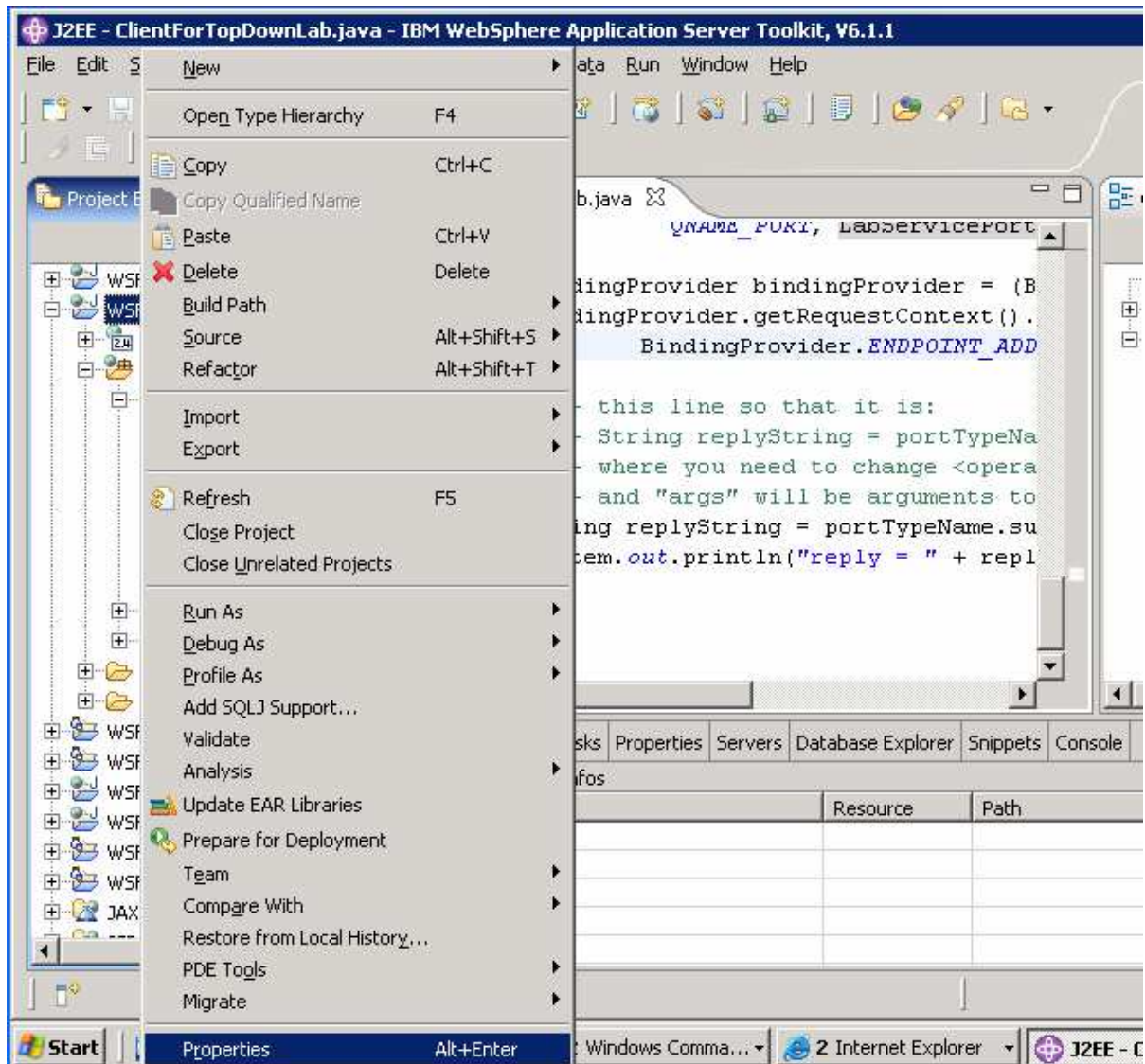
//! Change - this line so that it is:
//! Change - String replyString = portTypeName.<operationName>(args);
//! Change - where you need to change <operationName> to the operation to invoke
//! Change - and "args" will be arguments to that operation
    String replyString = portTypeName.submit("Welcome to Web Services");

```

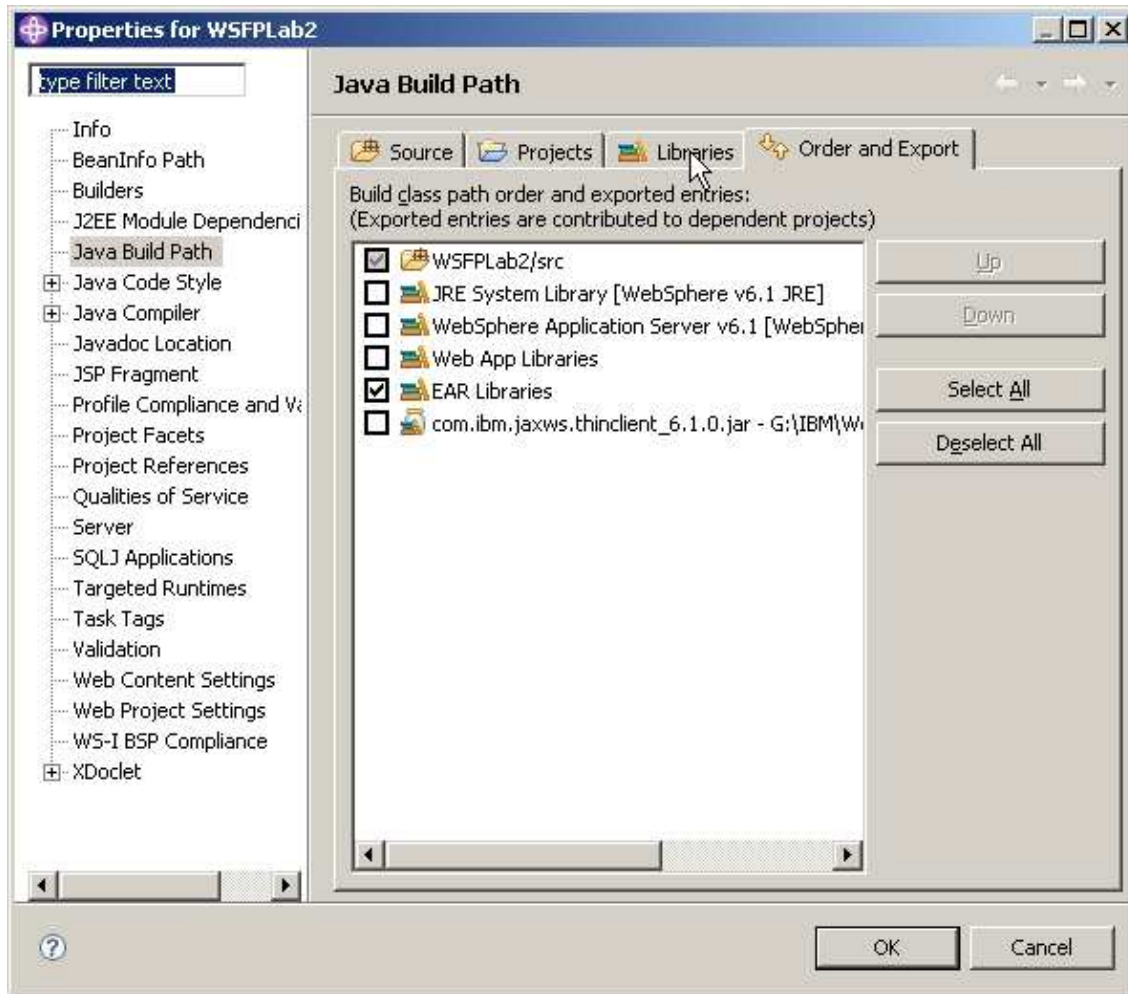
___ 15. Save your changes with **Ctrl + S**.

___ 16. Changing the path.

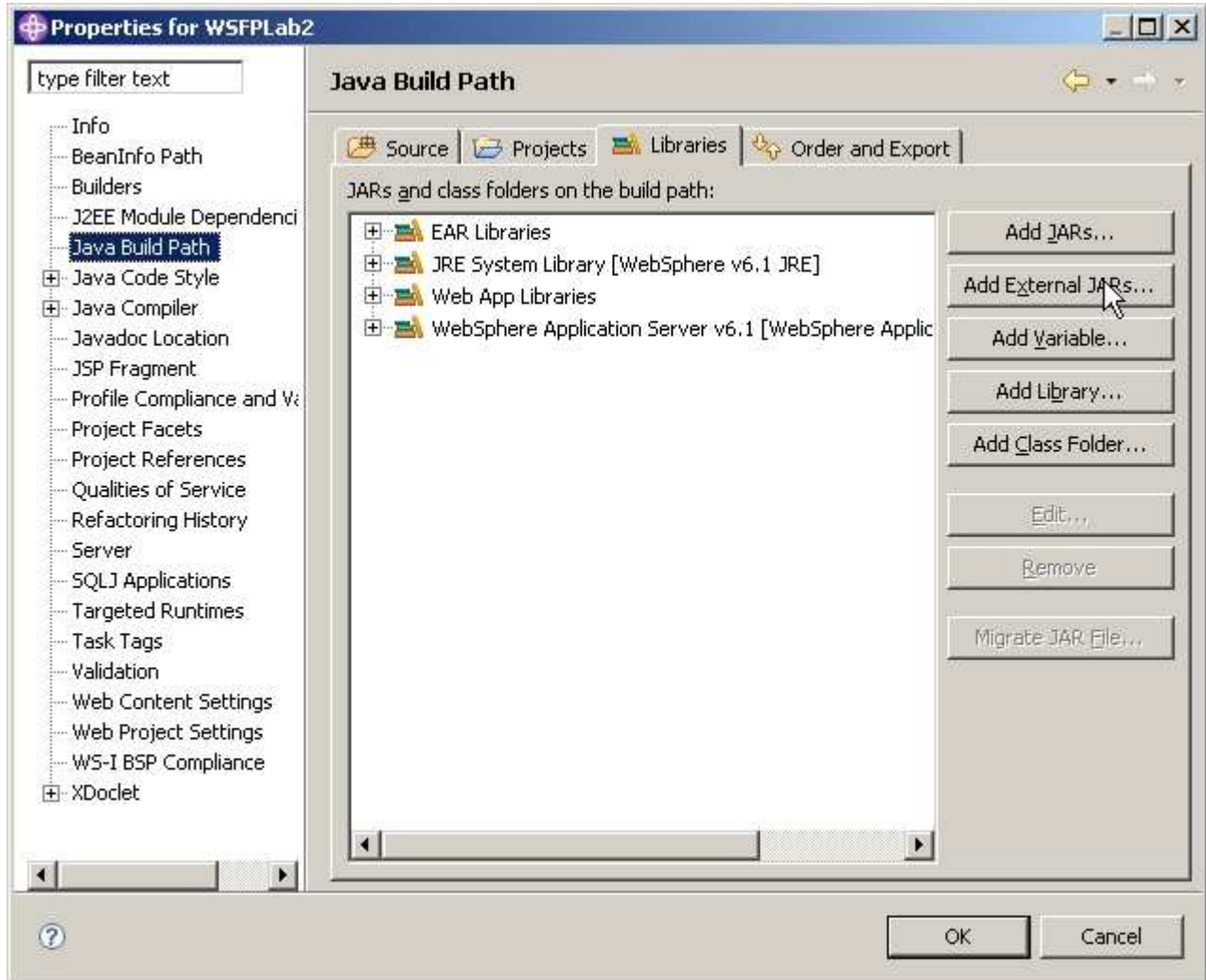
___ a. Before you can run the client code, you must change the build path being used. In the Project Explorer right-click on **WSFPLab1Client** and select **Properties**.



___ b. Select **Java Build Path**. Then select the **Libraries** tab.



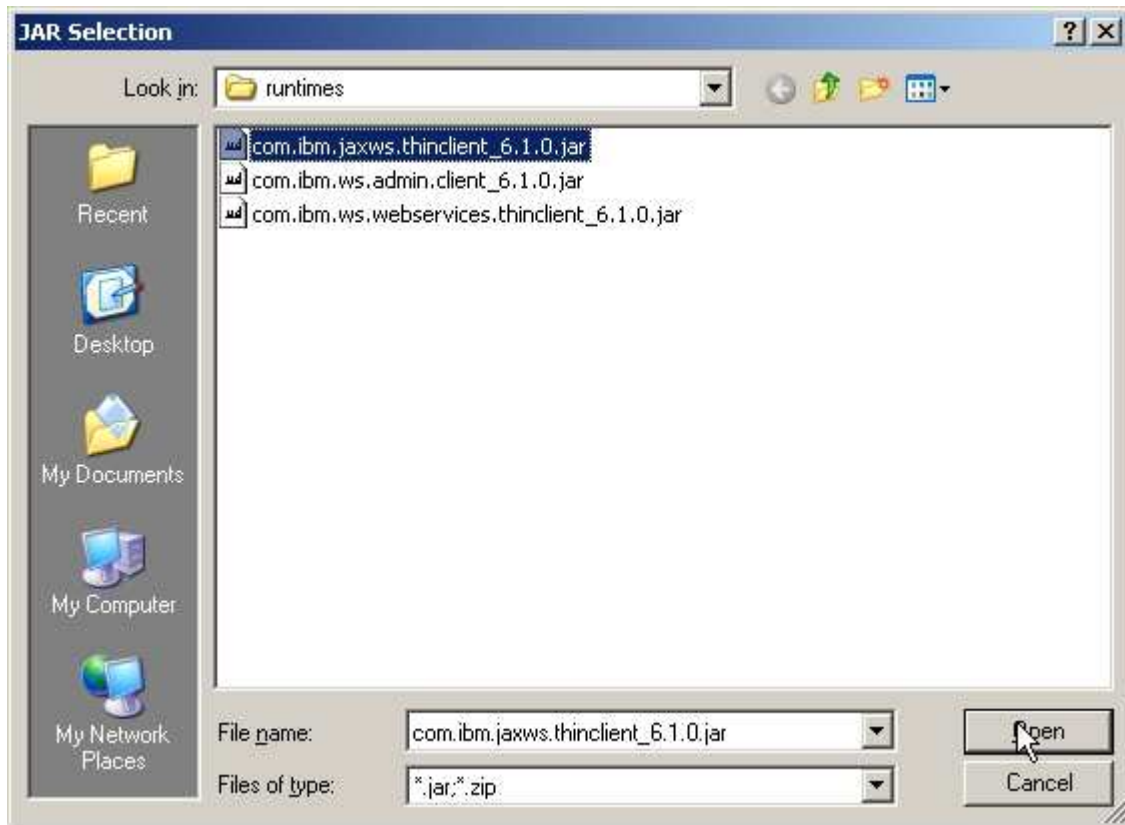
__ c. Select the button for **Add External JARs**.



__ d. Select the IBM WebSphere Thin Client for Web Services jar file by navigating to:

C:\WebSphere\Appserver\Runtimes

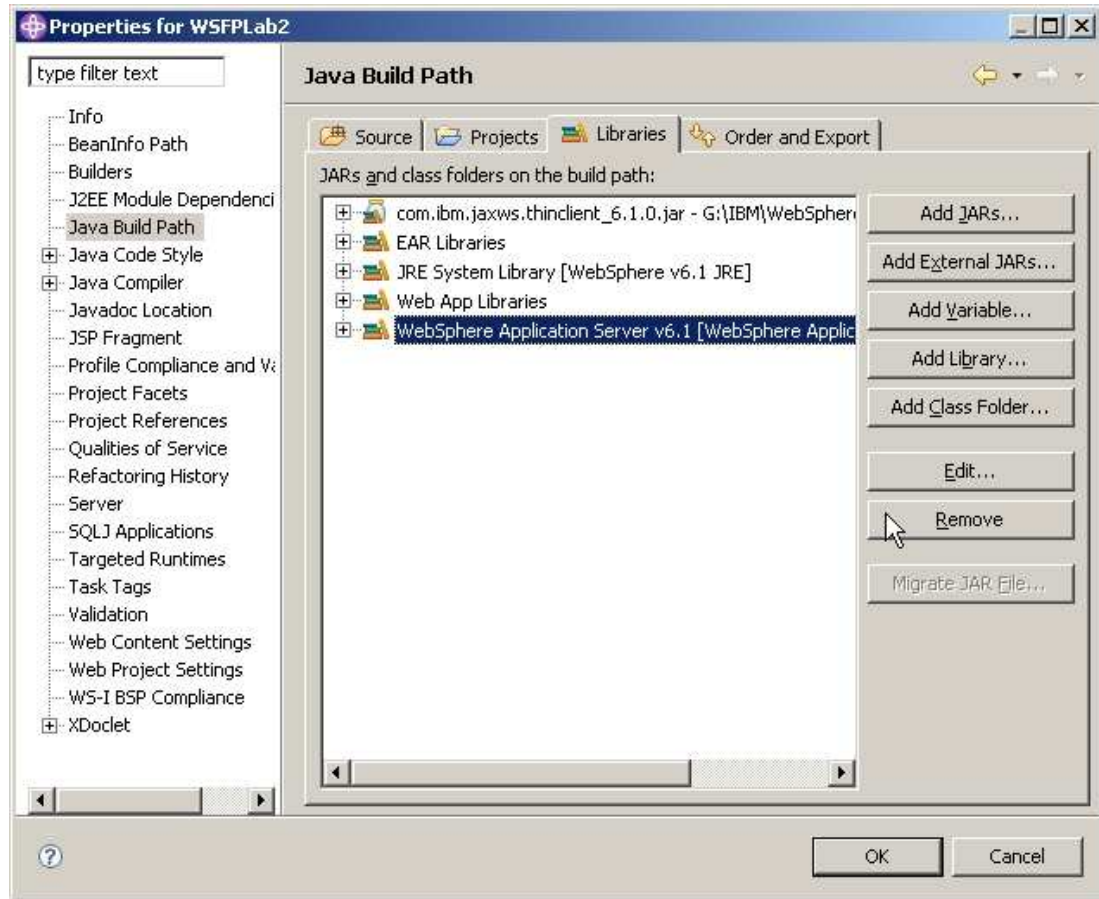
and selecting the jar named **com.ibm.jaxws.thinclient_6.1.0.jar** and clicking **Open**.



___ e. In the Libraries list, select the entry named:

WebSphere Application Server v6.1[WebSphere Application Server v6.1stub]

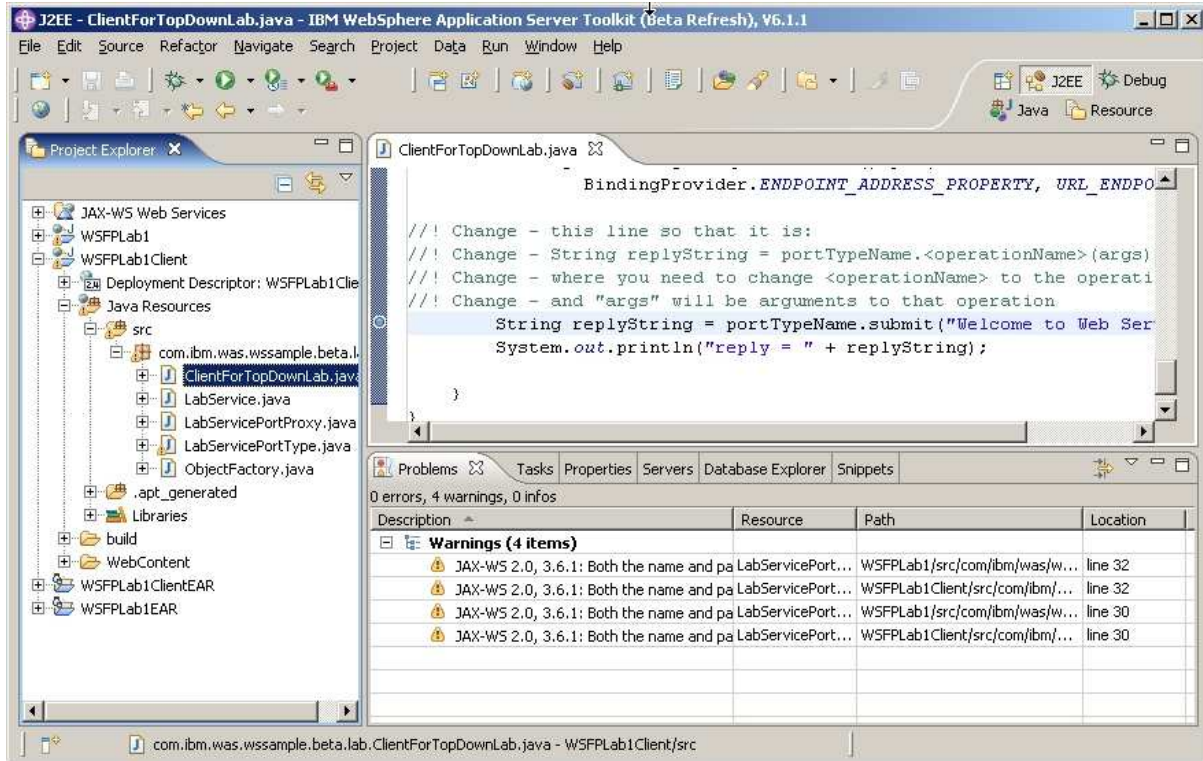
and click **Remove**.



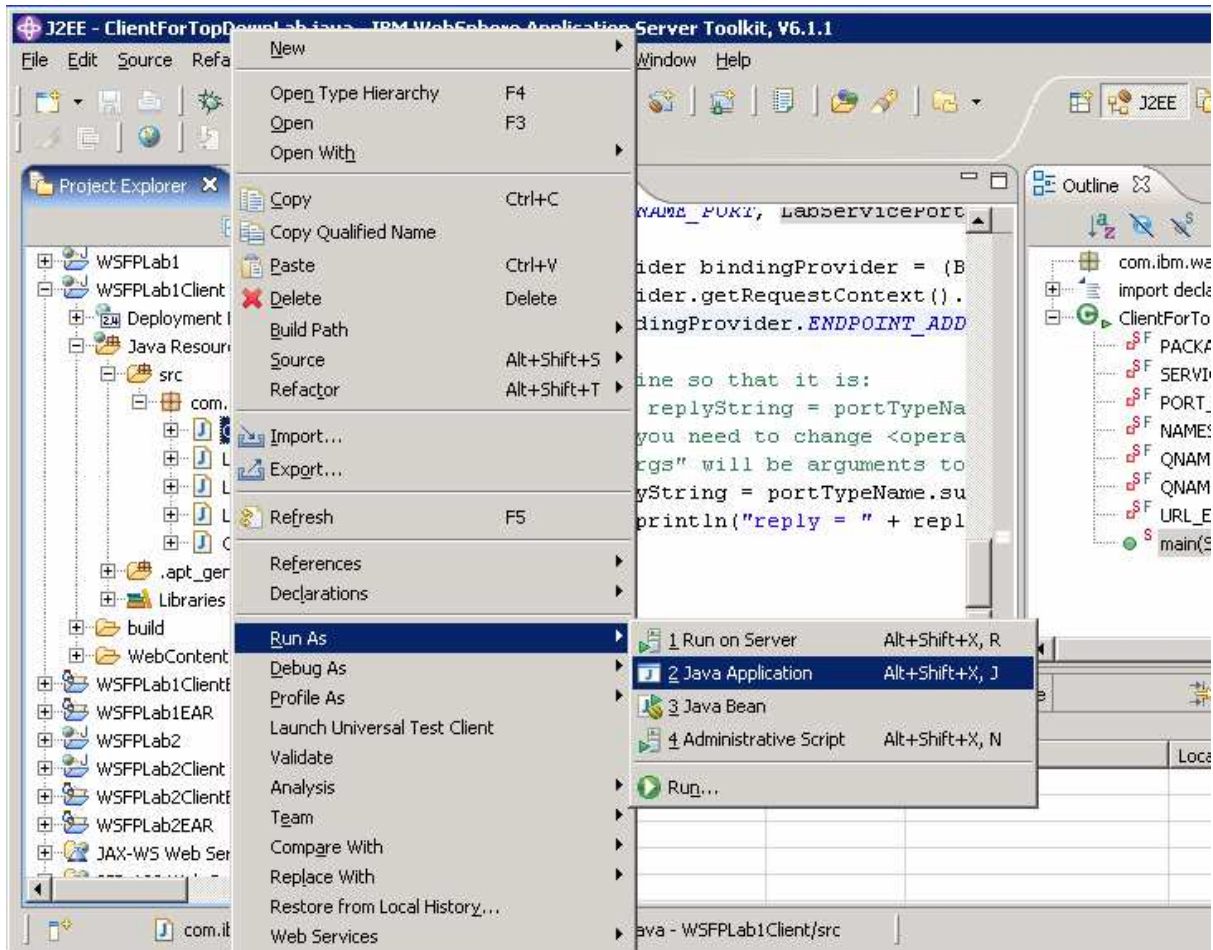
___ f. Click **OK**.

17. Running the client code.

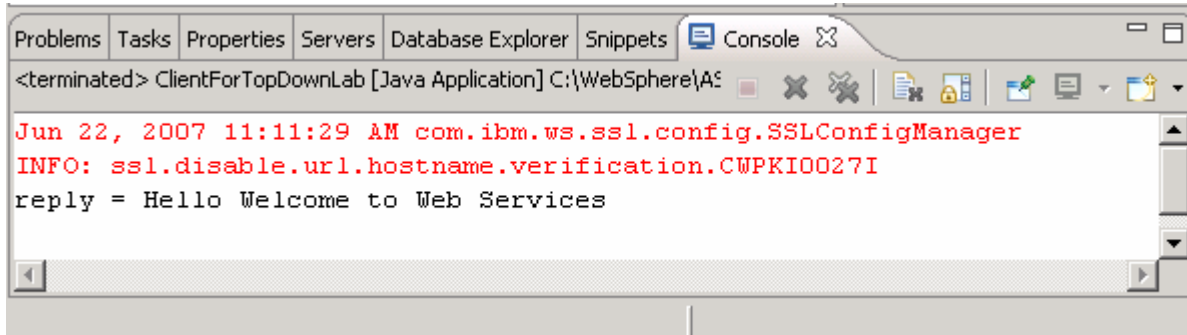
- a. Expand **WSFPLab1Client** from the **Project Explorer** in the **AST**, expand **Java Resources**, expand **src**, expand **simple** and select **ClientForTopDown.java**



__ b. Right-click **ClientForBottomUp.java** and select **Run As** and click **Java Application**.



__ c. The following output should appear in the console of the AST.



The screenshot shows a console window from an IDE. The title bar reads '<terminated> ClientForTopDownLab [Java Application] C:\WebSphere\AS'. The console output is as follows:

```
Jun 22, 2007 11:11:29 AM com.ibm.ws.ssl.config.SSLConfigManager  
INFO: ssl.disable.url.hostname.verification.CWPKI0027I  
reply = Hello Welcome to Web Services
```

Summary

In this lab, you took a top-down development approach to build a Web Service starting with a WSDL file containing a description of the Web Service. You performed the following steps to do this:

1. Created a Dynamic Web Project in the AST
2. Imported the WSDL File into the Dynamic Web Project
3. Created a Web Service From the WSDL File and exported the EAR file containing that Web Service
4. Installed the EAR file, started the Web Service application, and performed simple verification
5. Developed a Web Services client for the Web Service, ran the client code and interacted with the Web Service you created.