

WebSphere Application Server Version 6.1

Sales and Technical Enablement Workshop

Lab 05 – Web services

Introduction

The *Application Server Toolkit (AST)* provides basic support for the creation of new applications targeting WebSphere Application Server V6.1. This includes wizards and tools for creating new Web applications, Web services, portlets, and EJBs, as well as annotation based programming support, new administration tools for the creation and maintenance of wsadmin Jython files, and tools to edit WebSphere-specific bindings and extensions.

Lab Requirements

This lab assumes that the following setup is complete prior to starting the lab:

- VMware Player 1.0.x or VMware Workstation v5.5.x installed on your machine. A free VMware player is available from <http://www.vmware.com/products/player/>
- A machine with 2 GB is RAM is preferred.

Overview:

This exercise will highlight how to perform development and assembly tasks for Web services. You will create and validate WSDL files using the graphical editor, use the Web service wizard to create Web services and clients from a variety of resources, and test Web services using the Web services Explorer and the Universal Test Client.

You will develop and test a Stock Quote Web service in the Application Server Toolkit. The Stock Quote Web service is a simple, sample application that is shipped with WebSphere Application Server Version v6.1. It takes as input a stock symbol and returns a random, current price quotation.

Part 1: Start the Application Server Toolkit

As an introduction to the Application Server Toolkit, start the Workbench and begin developing the StockQuote Web service application.

- ____ 1. From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the 'N' icon.



- ____ 2. Select **IBM WebSphere** → **Application Server Toolkit V6.1** → **Application Server Toolkit V6.1**



- ____ 3. Create a workspace of `/root/AST/Lab05-workspace` and click **OK**



- ____ 4. From the Application Server Toolkit Welcome page, click on **Workbench – Go to the workbench**

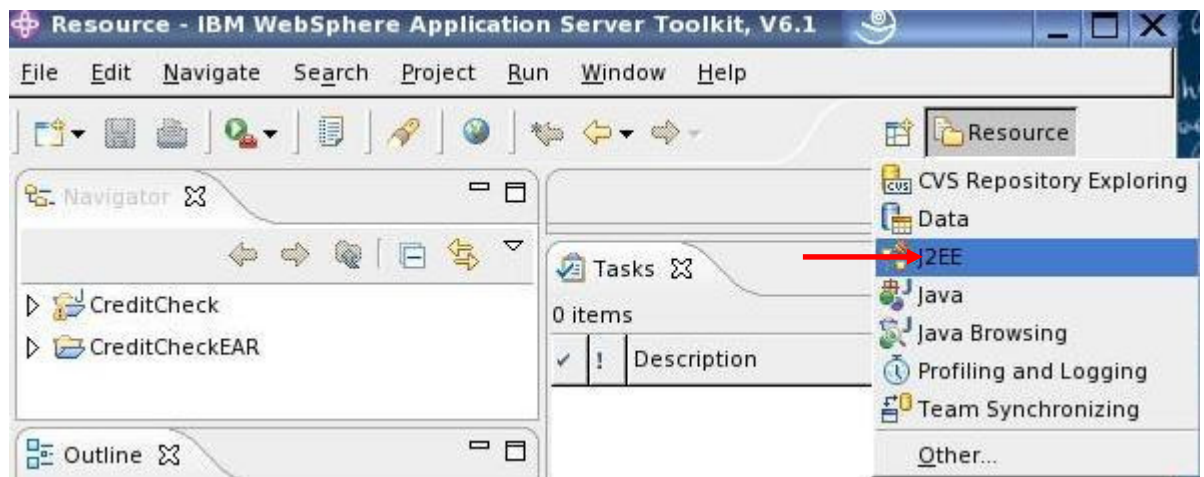


- ____ 5. Initially, in the first Workbench window that is opened, the Resource perspective is displayed. A shortcut bar appears in the top right corner of the window that allows the user to open new perspectives and switch between ones already open. Open a Data perspective.

- 1) From the toolbar, click on the **Open Perspective** button



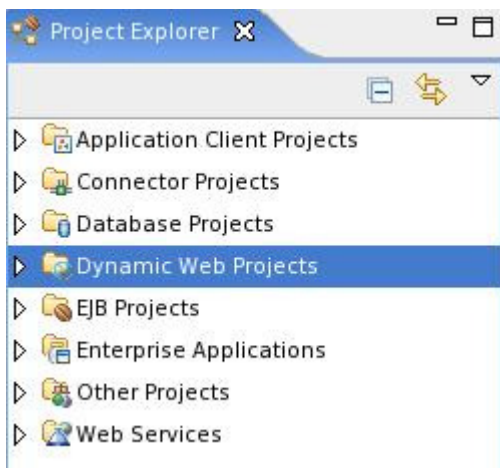
- 2) Select **J2EE**



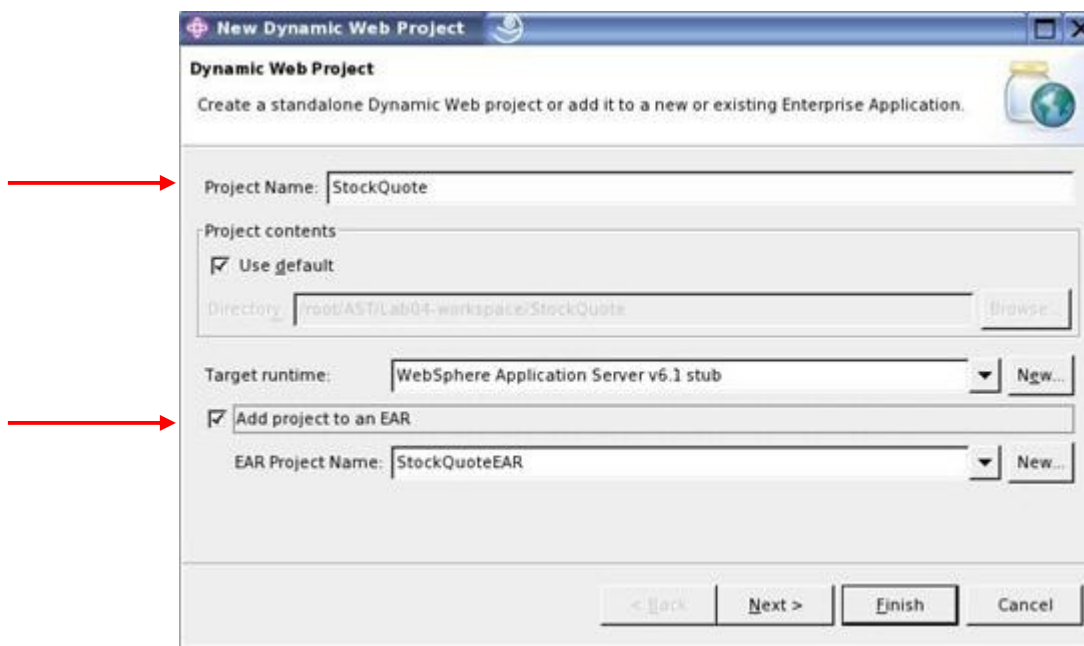
Part 2: Developing Web Service applications

The WebSphere® Application Server Toolkit (AST) provides tools for working with Extensible Markup Language (XML) files. You can import, validate, and edit DTD, XML, or XML schema files. The AST also allows you to creating and validate WSDL and use the WSDL editor to create or modify WSDL files graphically so that you can create top-down Web services.

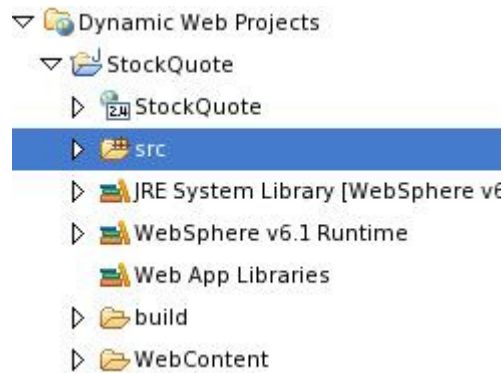
- ___ 1. Use the WSDL editor to create the **StockQuote.wsdl** file. You will use this WSDL file a later part of the lab to create a Web service.
 - ___ a. Locate the **Project Explorer** view and select **Dynamic Web Projects**
 - ___ b. Right-click and select **New → Dynamic Web Project**



- ___ c. Enter a Project Name of **StockQuote**. Select the **Add project to an EAR** checkbox. Click **Finish**

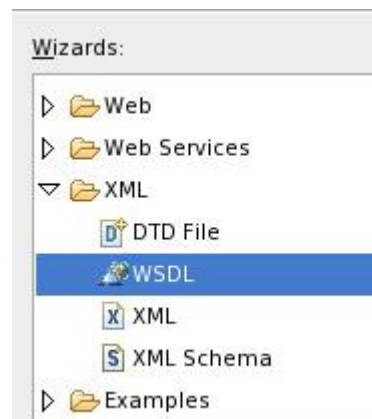


__ d. Expand the newly created **StockQuote** Web project and select the **src** folder

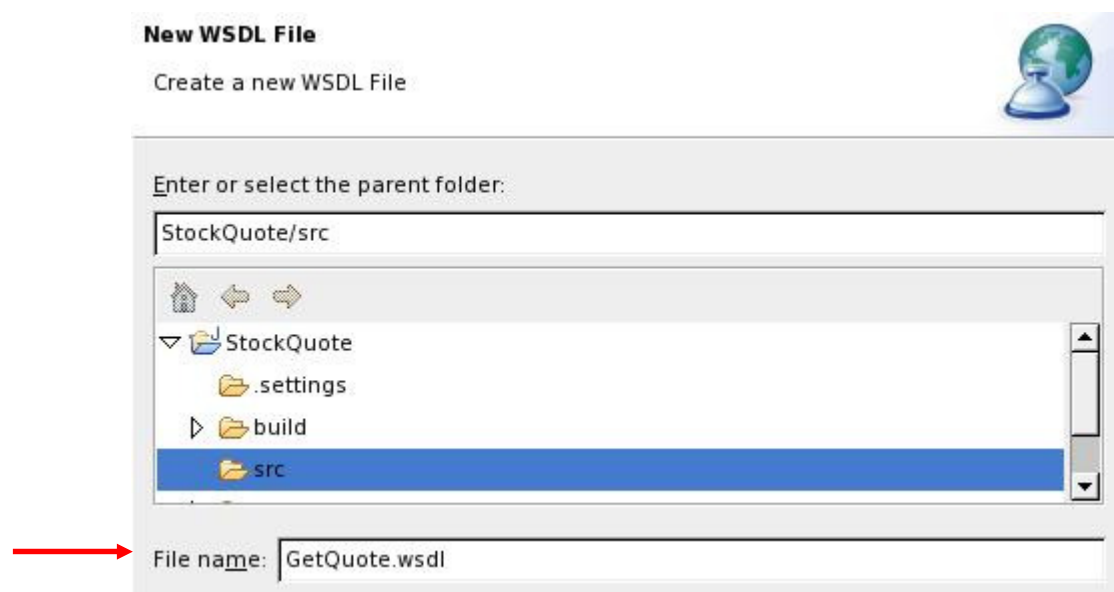


__ e. Right-click and select **New → Other**

__ f. In the **Wizards** panel, expand **XML** and select **WSDL**. Click **Next**.



__ g. Enter a File name of **GetQuote.wsdl**. Click **Next**.



- ___ h. Examine the choices for WSDL creation. Using the defaults, the wizard will create a WSDL skeleton and use Document Literal SOAP bindings for WS-I compliance. Click **Finish**.

New WSDL File

Options

Specify the attributes for the new WSDL file.

Target namespace:

Prefix:

☒ Create WSDL Skeleton

Protocol:

SOAP Binding Options

☒ document literal

☐ rpc literal

< Back Next > Finish Cancel

- ___ i. The skeleton **GetQuote.wsdl** file is opened in the graphical WSDL editor. Examine the layout of the WSDL file.

GetQuote.wsdl

Imports

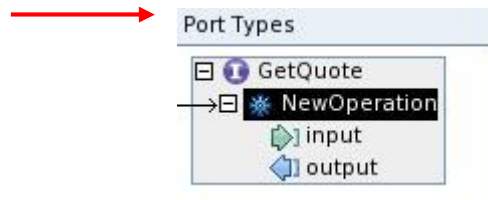
Types

Services

Bindings

Port Types

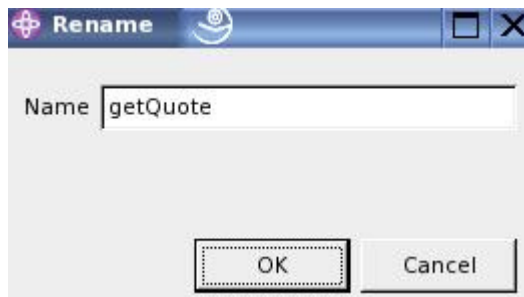
___ j. In the **WSDL** editor, locate the **Port Types** section. Expand **GetQuote**.



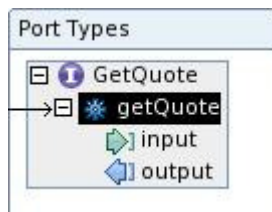
___ k. Select **NewOperation**

___ l. Right-click and select **Rename**

___ m. Enter a name of **getQuote**



___ n. Click **OK**



___ o. Use **CTRL-S** to save the file

___ p. You just performed a short example of developing a WSDL file using the graphical editor. You can develop a complete WSDL file using these techniques. However, to save time in this lab, you will now import **GetQuote.wsdl** from the filesystem.

1) From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the 'Personal Files' icon.

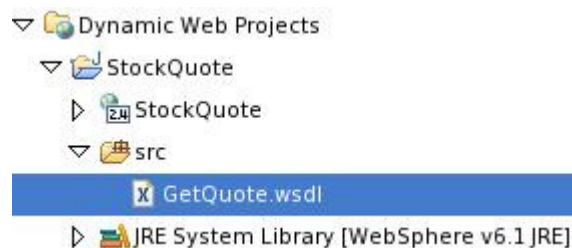


- 2) Navigate to:
`/opt/IBM/WAS61/AppServer/samples/src/WebServicesSamples/Clients/simpleClients/samples/stock`. You should see four files.



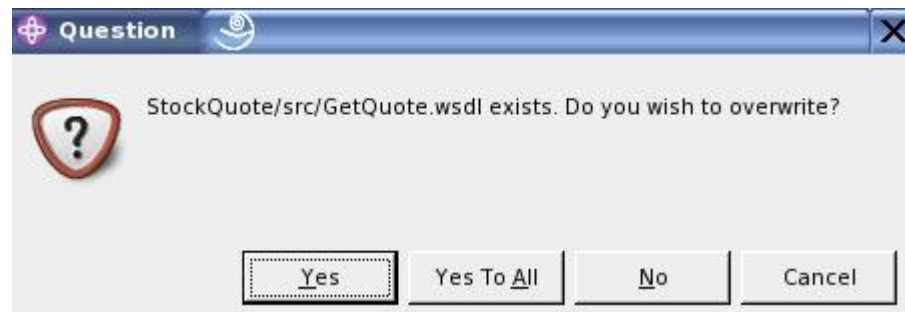
- 3) Right-click on **GetQuote.wsdl** and select **Copy**. (you want to copy the entire file, not the contents)

- 4) Return to the **AST**. Expand the **src** folder and select **GetQuote.wsdl**

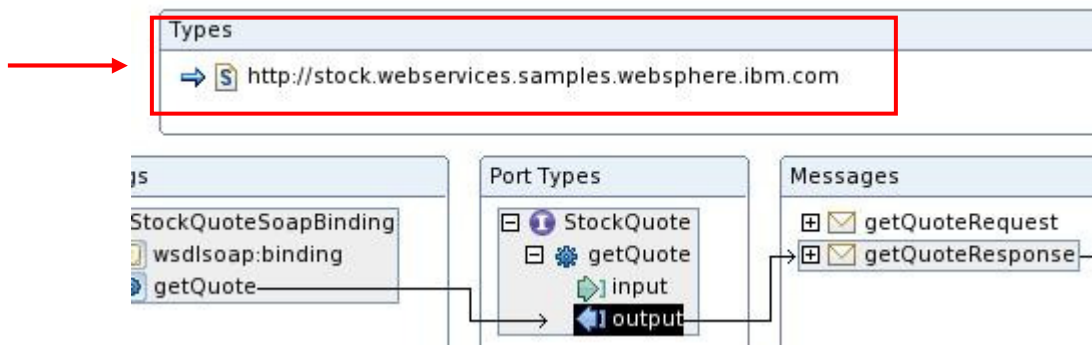


- 5) Use the **CTRL+V** key to paste

- 6) Answer **Yes** to overwrite the file.



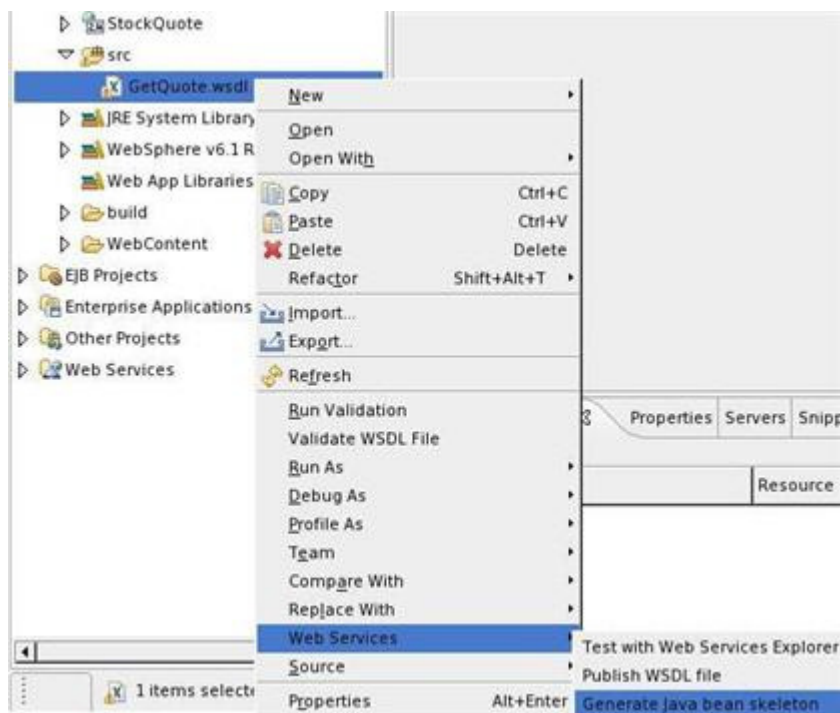
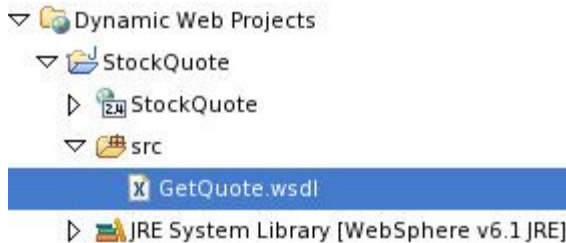
- ___ q. Examine the contents of **GetQuote.wsdl** in the WSDL editor. Locate the **Types** section. It should read <http://stock.webservices.samples.websphere.ibm.com>. **Close** the WSDL editor.



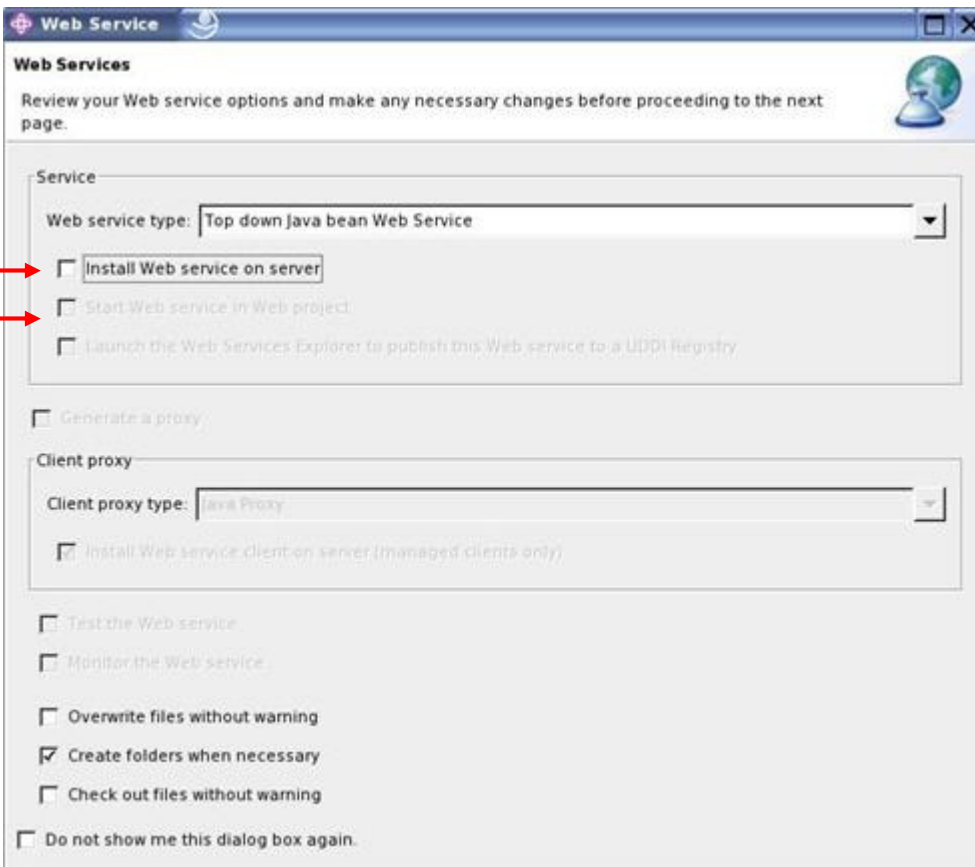
Part 3: Developing Web services applications

The WebSphere® Application Server Toolkit allows creation of top-down and bottom-up Web services. Wizards are provided that allow you to create Web services and clients from a variety of resources.

- ___ 1. Build a top-down Web service from the **GetQuote.wsdl** file
 - ___ a. From the **Project Explorer** view, open the **src** folder and highlight **GetQuote.wsdl**
- ___ b. Right-click on **GetQuote.wsdl** and select **Web Services → Generate Java Bean Skeleton**

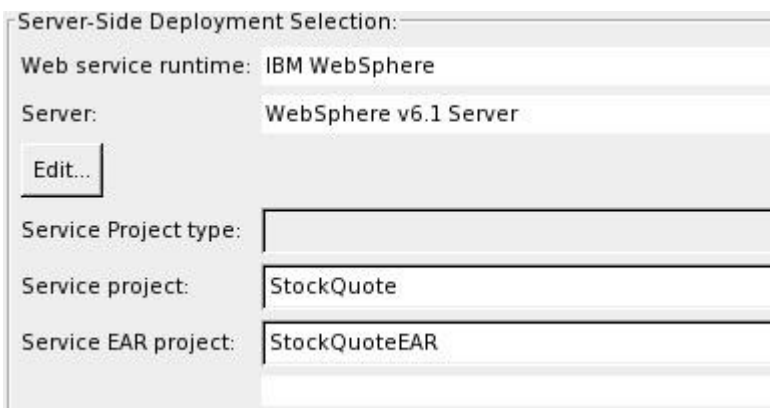


- ___ c. In the Web Services panel, unselect 'Install Web service on server' and 'Start Web service in Web Project'. Click **Next**



The image shows a 'Web Service' configuration dialog box. The title bar says 'Web Service'. Below the title bar, it says 'Web Services' and 'Review your Web service options and make any necessary changes before proceeding to the next page.' The main area is divided into sections. The 'Service' section has a dropdown for 'Web service type' set to 'Top down Java bean Web Service'. Below this are three checkboxes: 'Install Web service on server' (unchecked, with a red arrow pointing to it), 'Start Web service in Web project' (unchecked, with a red arrow pointing to it), and 'Launch the Web Services Explorer to publish this Web service to a UDDI Registry' (unchecked). The 'Generate a proxy' checkbox is also unchecked. The 'Client proxy' section has a dropdown for 'Client proxy type' set to 'Java Proxy' and an unchecked checkbox for 'Install Web service client on server (managed clients only)'. At the bottom, there are several checkboxes: 'Test the Web service' (unchecked), 'Monitor the Web service' (unchecked), 'Overwrite files without warning' (unchecked), 'Create folders when necessary' (checked), 'Check out files without warning' (unchecked), and 'Do not show me this dialog box again' (unchecked).

- ___ d. In the **Object Section Page**, click **Next**
- ___ e. In the **Service Deployment Configuration**, accept the defaults. Click **Next**



The image shows a 'Server-Side Deployment Selection' dialog box. It has a title bar and a main area with several fields. The 'Web service runtime' is set to 'IBM WebSphere'. The 'Server' is set to 'WebSphere v6.1 Server'. There is an 'Edit...' button next to the server field. The 'Service Project type' is empty. The 'Service project' is set to 'StockQuote'. The 'Service EAR project' is set to 'StockQuoteEAR'.

- ___ f. In the Web service Skeleton Java Bean Configuration, accept the defaults. Click **Next**
- ___ g. In the Web service Publication dialog, do not publish the Web service. Click **Finish**

- ___ h. A new **StockQuoteSOAPBindingImpl.java** file is created and opened in the Java editor. This file contains the application logic of the `getQuote` method, which is the server-side Web service.



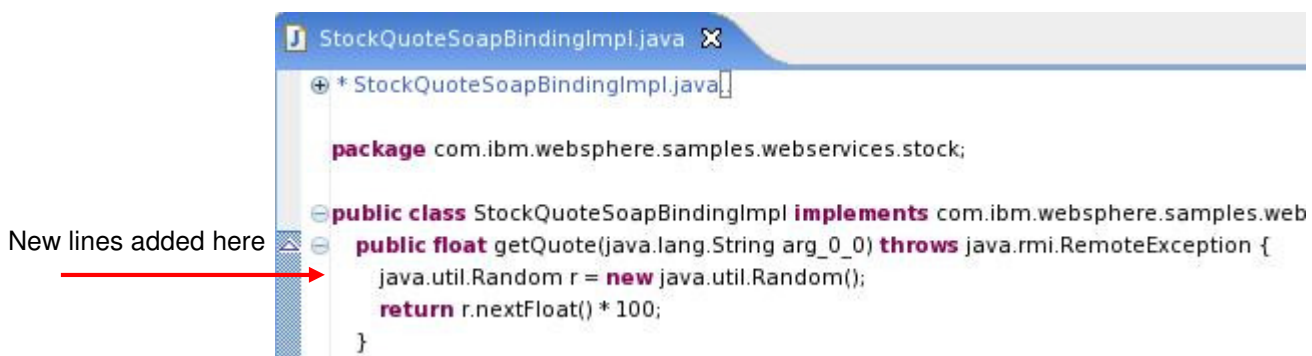
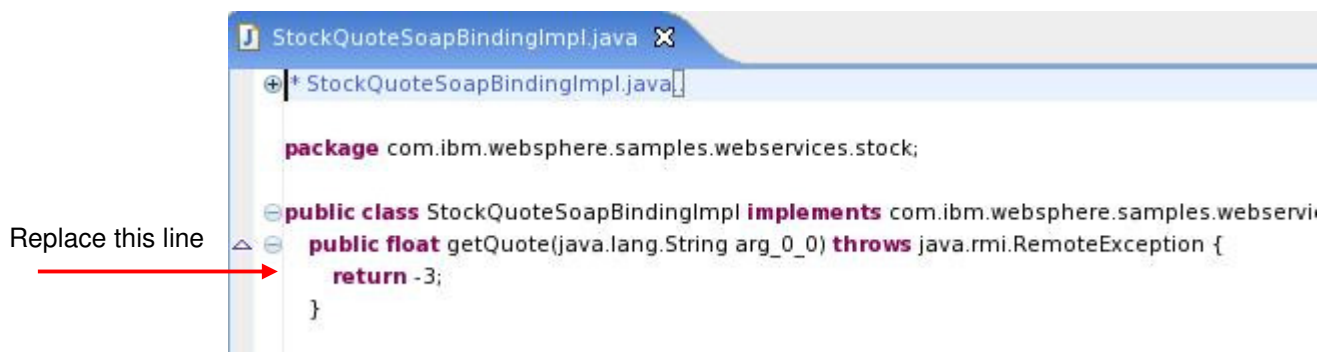
- ___ i. The application logic for the **getQuote** method needs to be updated. In this example, you are going to add Java code that will return a random number between 0 and 100. In the **StockQuoteSOAPBindingImpl.java** file, replace the existing line:

```
return -3;
```

with

```
java.util.Random r = new java.util.Random();
```

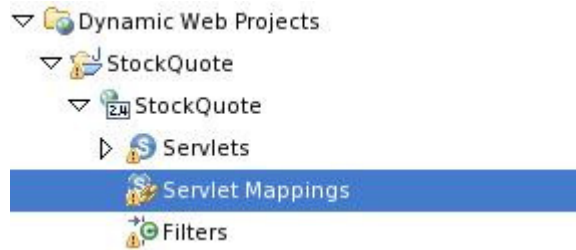
```
return r.nextFloat() * 100;
```



- ___ j. Save the **StockQuoteSOAPBindingImpl.java** file using the **CTRL+S** key.

___ k. Close the **StockQuoteSOAPBindingImpl.java** file

___ l. Return to the StockQuote Web project. Expand the StockQuote deployment descriptor. Highlight **Servlet Mappings**.



___ m. Right-click and select **New → Servlet Mapping...**



___ n. Enter a URL pattern of **services/StockQuote** This action maps the Web service servlet to a standard URL naming pattern.



___ o. Click **Finish**

Part 4: Configuring the WebSphere test environment

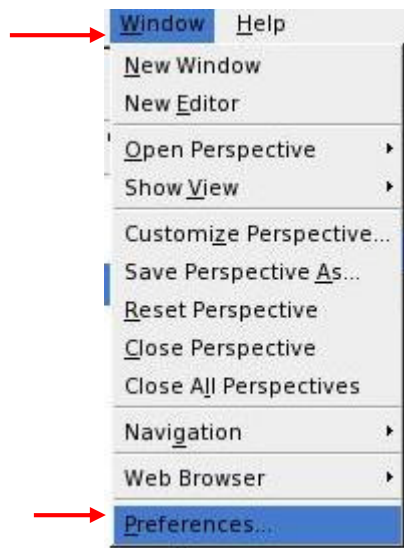
The *WebSphere® test environment* is a runtime environment that is integrated into the workbench for testing applications that are targeted for WebSphere Application Server.

The test environment for WebSphere Application Server v6.1 requires a **full** installation of the WebSphere Application Server and is enabled through a **Run server with resources within the workspace** publishing setting.

For this lab, WebSphere Application Server v6.1 is already installed on your machine and there is no need to install a test environment.

1. Configure and set the WebSphere test environment

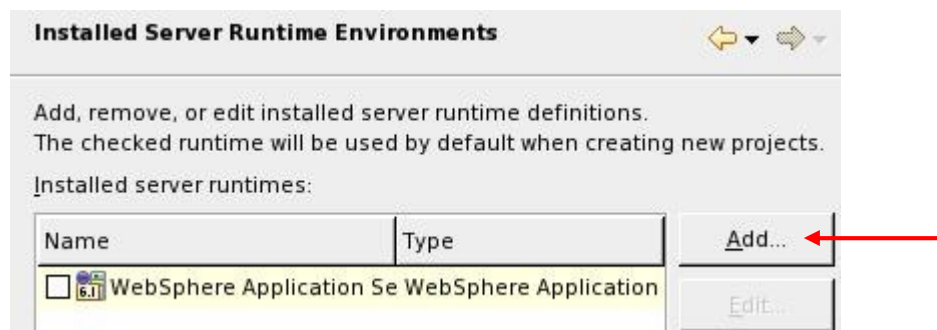
a. From the Application Server Toolkit menubar, select **Window → Preferences...**



b. On the left-hand menu, expand **Server**, select **Installed Runtimes**



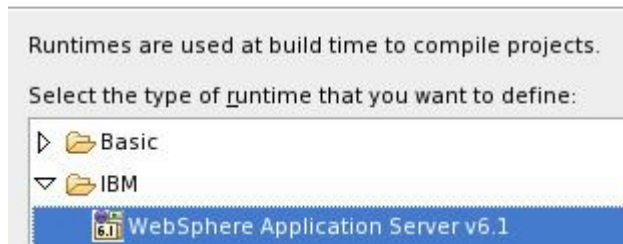
c. In the **Installed Server Runtime Environments** panel, click **Add...**



- ___ d. In the **New Server Runtime** dialog, accept the default server runtime of WebSphere Application Server v6.1 and click **Next**.

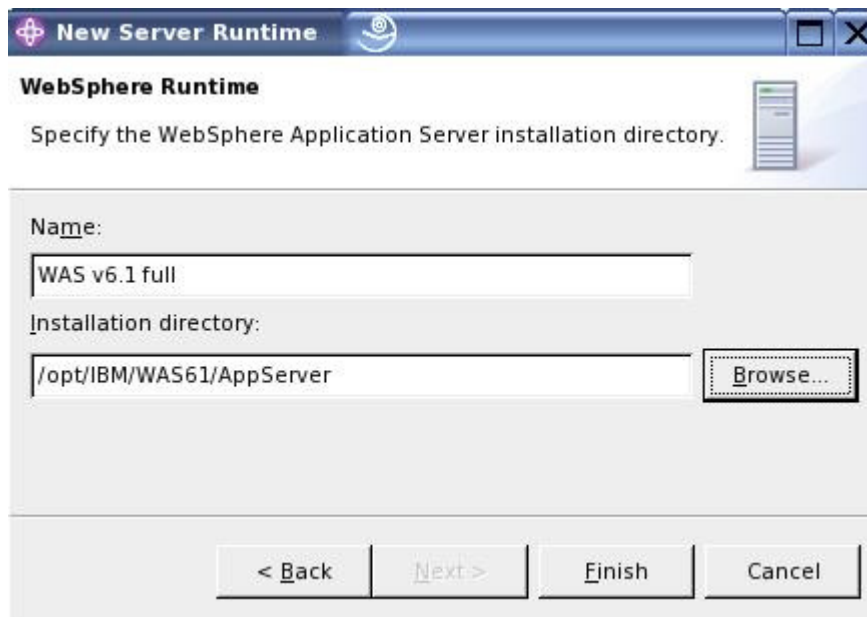
New Server Runtime

Define a new installed server runtime environment

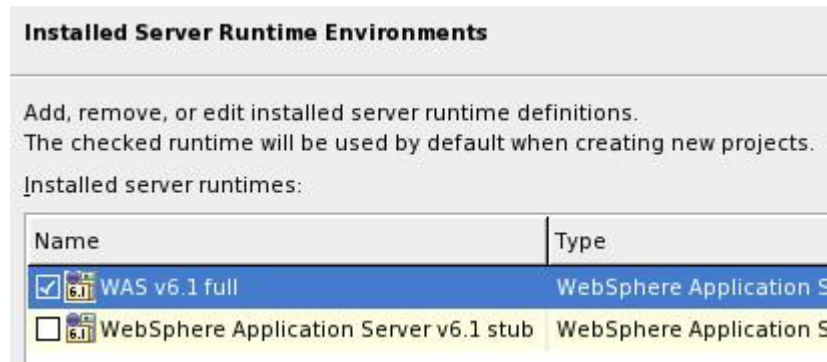


- ___ e. In the WebSphere Runtime panel -

- 1) Enter a **Name** of **WAS v6.1 full**
- 2) Set the installation directory to **/opt/IBM/WAS61/AppServer** – use the **Browse...** button
- 3) Click **Finish**



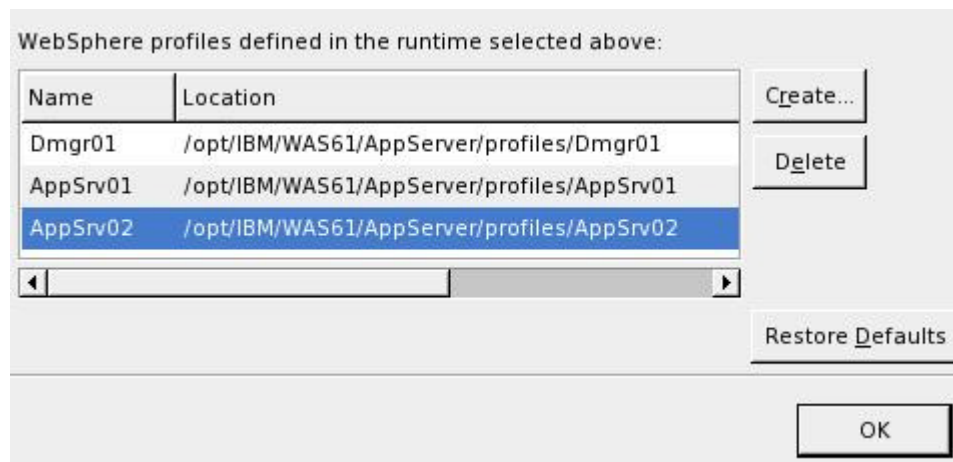
- ___ f. In the Installed Server Runtime Environments panel, select **WAS v6.1 full** as the default runtime



- ___ g. If you have already created WebSphere profiles in the previous labs, skip to the next page and begin with **2) Add a new WebSphere Application Server v6.1 test server**
- ___ h. If you do not have any existing WebSphere profiles, return to the left-hand side preferences and select **Server → WebSphere**

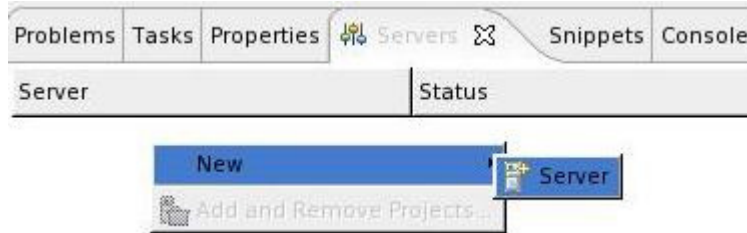


- ___ i. On the right-hand side of the view, the configured WebSphere runtimes and profiles are shown. Click on the **Create...** button. Create a new Application Server profile, with the Typical Profile options. When finished, click **OK** to return to the J2EE perspective.

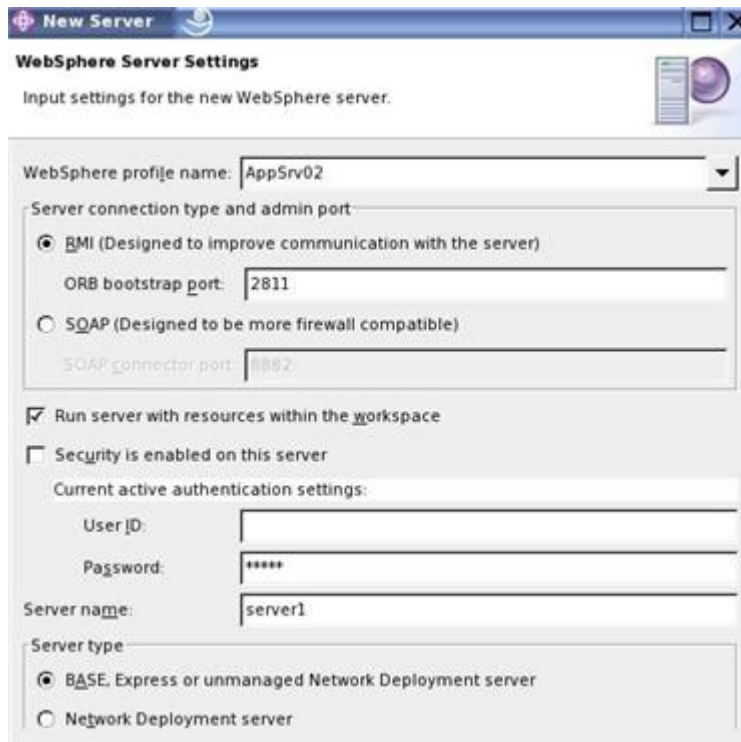


2. Add a new WebSphere Application Server v6.1 test server

- a. Locate and select the **Servers** view. The Servers view allows you to manage the servers. This view displays a list of all your servers and configurations that are associated with that server. You can use this view to start, start in debug mode, start in profile mode, restart, or stop the servers.
- b. Right-click in the Servers view and select **New → Server**



- c. In the **Define a New Server** panel, ensure the server type is **WebSphere v6.1 Server** and the Server Runtime is **WAS v6.1 full**. Click **Next**.
- d. In the WebSphere Server Settings:
 - 1) Select a WebSphere profile name of **AppSrv02** (or another profile you have created). Notice the tool automatically determines the correct WebSphere Administration port of 2811.
 - 2) **Uncheck** 'Security is enabled on this server'. Also notice that the tool is designed to work with Base and Express servers, as well as a Network Deployment topology.
 - 3) Click **Next**

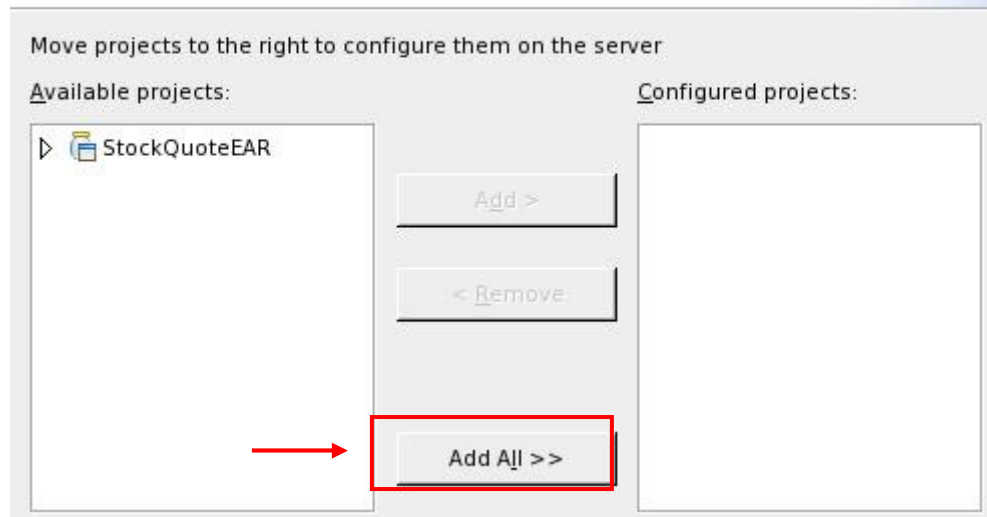


___ e. In the **Add and Remove Projects** panel

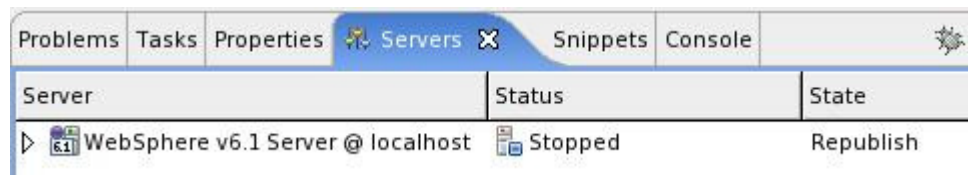
- 1) Click **Add All >>** to add the **StockQuoteEAR** project to the server.
- 2) Click **Finish** when you are done.

Add and Remove Projects

Modify the projects that are configured on the server



___ f. In the **Servers** view, you should now have a WebSphere v6.1 Server in the Stopped state.



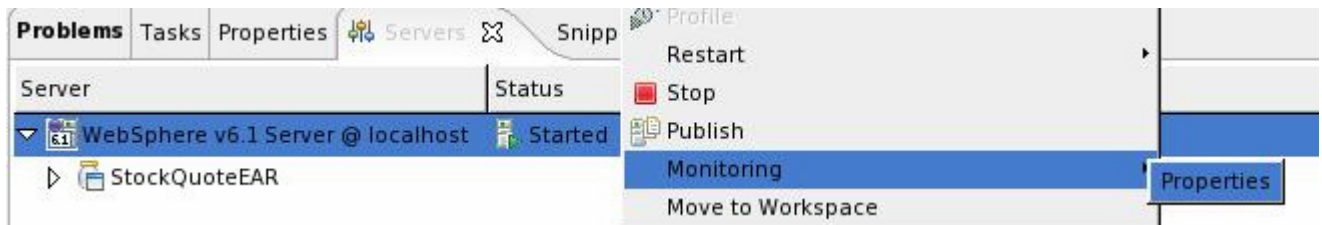
___ g. Highlight **WebSphere v6.1 Server @ localhost**. Click the **Start** icon.



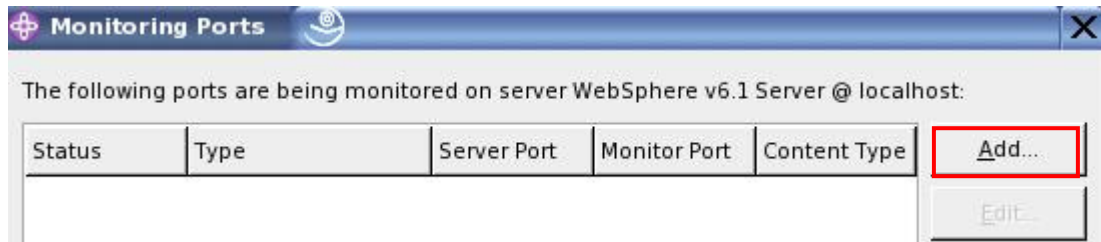
___ h. Examine the **Console** view while the server starts.



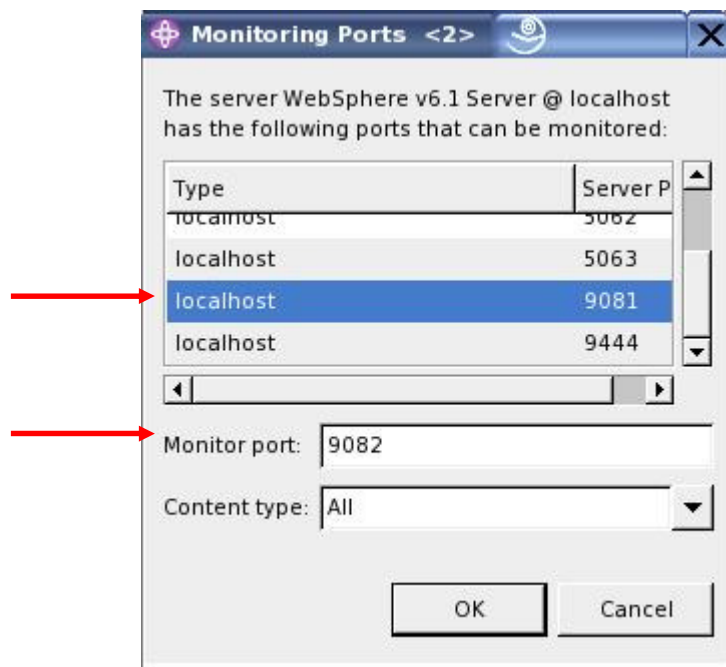
- ___ i. Once the server has started. Right-click and select **Monitoring → Properties**



- ___ j. In the Monitoring window, click on the **Add** button



- ___ k. A list of ports currently being used by the Application Server is shown. Scroll down and select port **9081** (this port may vary depending on the profile you are using). Take note of the **Monitor Port**. In this example, the Monitor Port is **9082**. You will need this information later in the lab. Click **OK**.



___ l. In the Monitoring Ports panel, **select** the newly added row. Click **Start**



___ m. The Status should change from Stopped to **Started**.

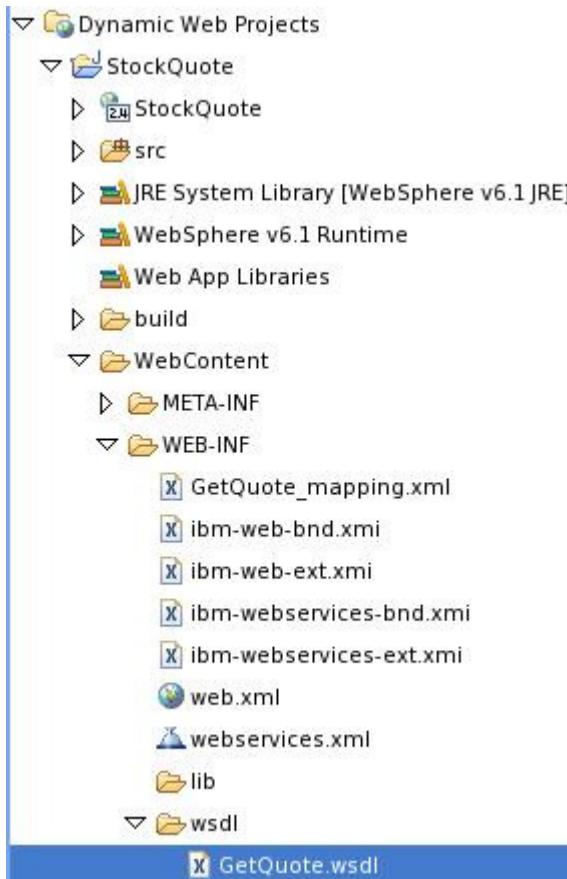


___ n. Click **OK** to finish and close the monitoring window.

Part 5: Testing the StockQuote application

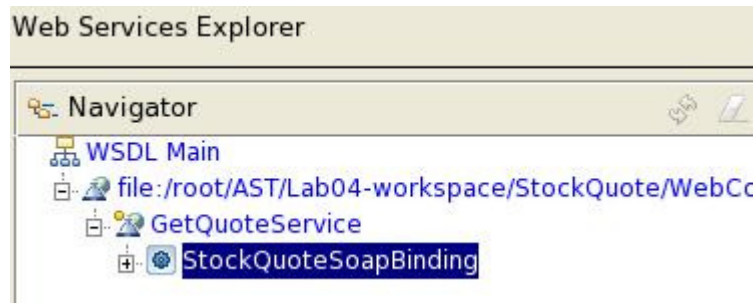
The AST allows for testing Web services using the Web services Explorer or the Universal Test Client. Additionally, the AST allows for validation of your Web service, WSDL files, or even the SOAP traffic passing through the service using a variety of validation tools.

- ___ 1. In this part of the lab, you will test the StockQuote Web service
 - ___ a. From the **Project Explorer** view, return to the StockQuote Web Project. Expand the **WebContent** → **WEB-INF** → **wsdl** folders. Select **GetQuote.wsdl**



- ___ b. Right-click and select **Web Services** → **Test with Web Services Explorer**

- ___ c. The **Web Services Explorer** runs in a browser. From the browser, locate the **Navigator** section which shows the Web Service components. You can expand and click on the entries to see additional information about the Web service.



- ___ d. With **StockQuoteSoapBinding** selected in the Navigator, locate the WSDL Binding Details section. To the right of the **Endpoints** label, click **Add**



- ___ e. An additional entry will be added to the Endpoints list. Change the port number from 9080 to **9082** (9082 is the **Monitor Port** you recorded earlier in the lab with the TCP/IP Monitor - enter your specific port number).



___ f. Select the **http://localhost:9082/StockQuote/services/StockQuote** checkbox. Then click **Go**

▼ **Endpoints** [Add](#) [Remove](#)

Endpoints	
<input type="checkbox"/>	http://localhost:9080/StockQuote/services/StockQuote
1) <input checked="" type="checkbox"/>	http://localhost:9082/StockQuote/services/StockQuote

2)

___ g. In the Operations section, click on **getQuote**

▼ **Operations**

Name	Documentation
getQuote	--

___ h. In the **Invoke a WSDL operation** section, enter a stock symbol (e.g. **IBM**) and click **Go**.

Invoke a WSDL Operation

Enter the parameters of this WSDL operation and click **Go** to invoke.

Endpoints

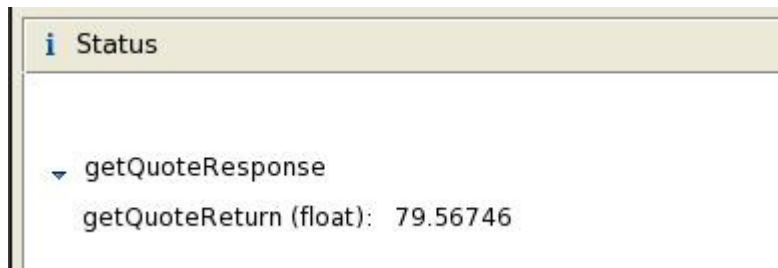
http://localhost:9082/StockQuote/services/StockQuote ▼

▼ [getQuote](#)

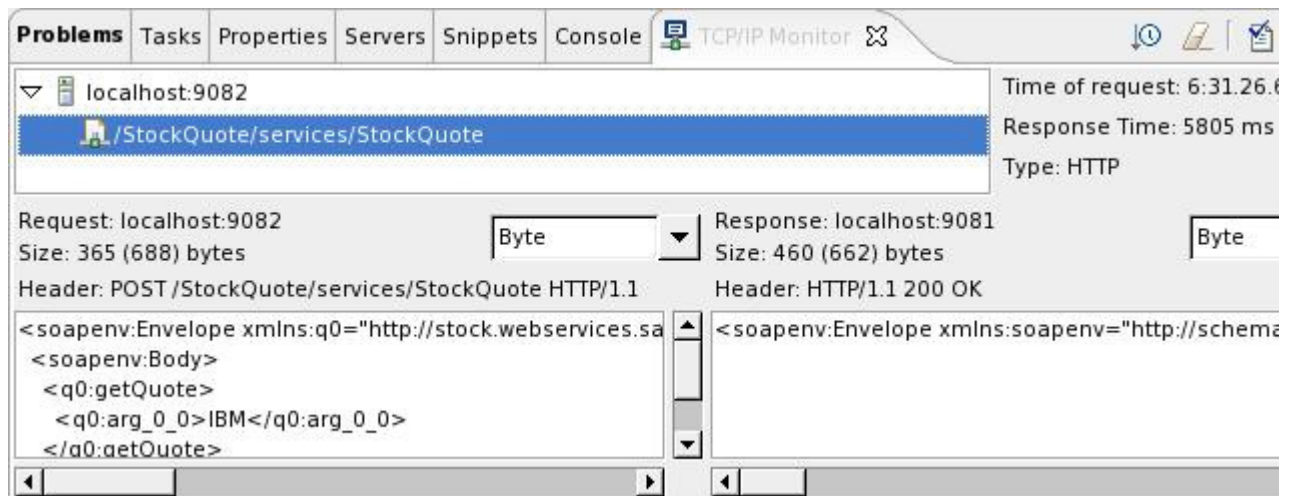
[arg_0_0](#) string

IBM

- ___ i. In the **Status** section, you will see the result of the **getQuote** Web service call. You can experiment with the **getQuote** operation using additional stock symbols if desired.



- ___ j. Returning to the **AST**, locate the **TCP/IP Monitor** view. This view is effective for seeing the complete details of the Web service invocation.

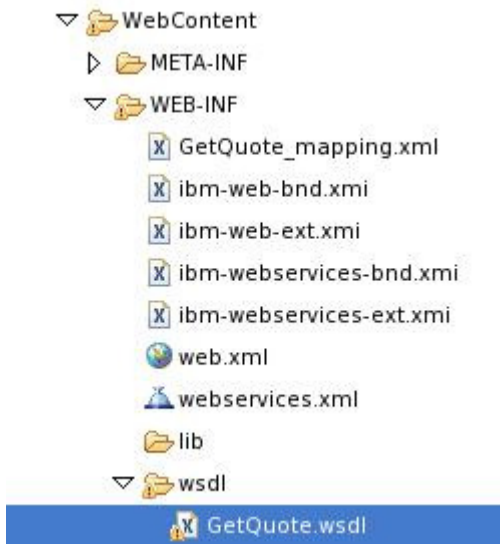


- ___ k. It may be helpful to change the Response output from **Byte** to **XML** for easier reading.

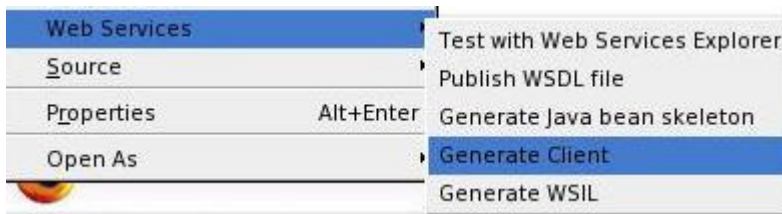


2. **Additional Web Service Client testing**

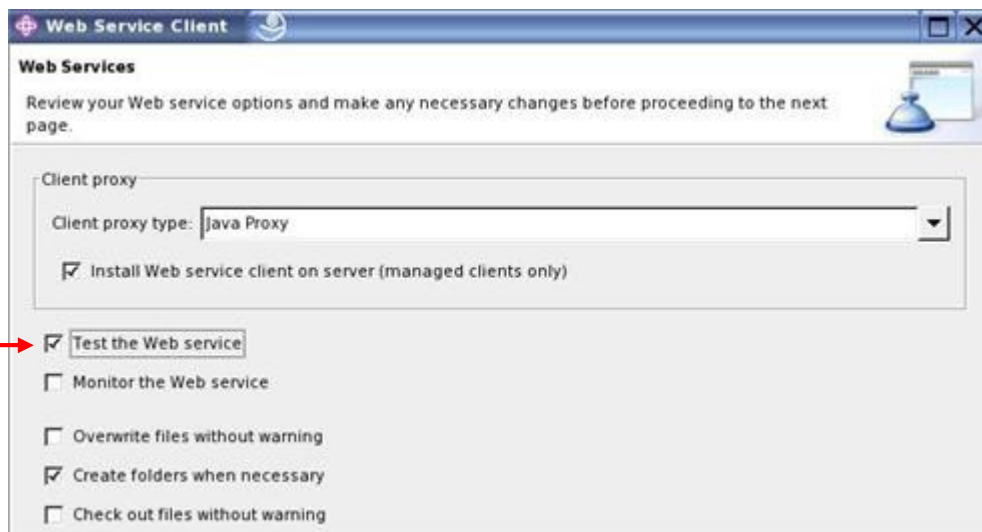
- ___ a. From the Servers view, highlight **WebSphere v6.1 Server @ localhost**.
- ___ b. Right-click and select **Restart → Start**. Wait for the server to restart. (Restarting the server here works around a potential issue with the Generate Client in the next steps)
- ___ c. From the Project Explorer view, ensure the **GetQuote.wsdl** file is still selected



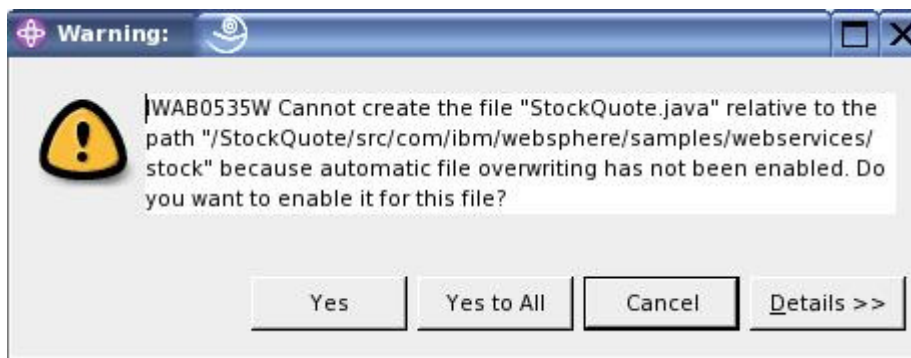
- ___ d. Right-click on **GetQuote.wsdl** and select **Web Services → Generate Client**



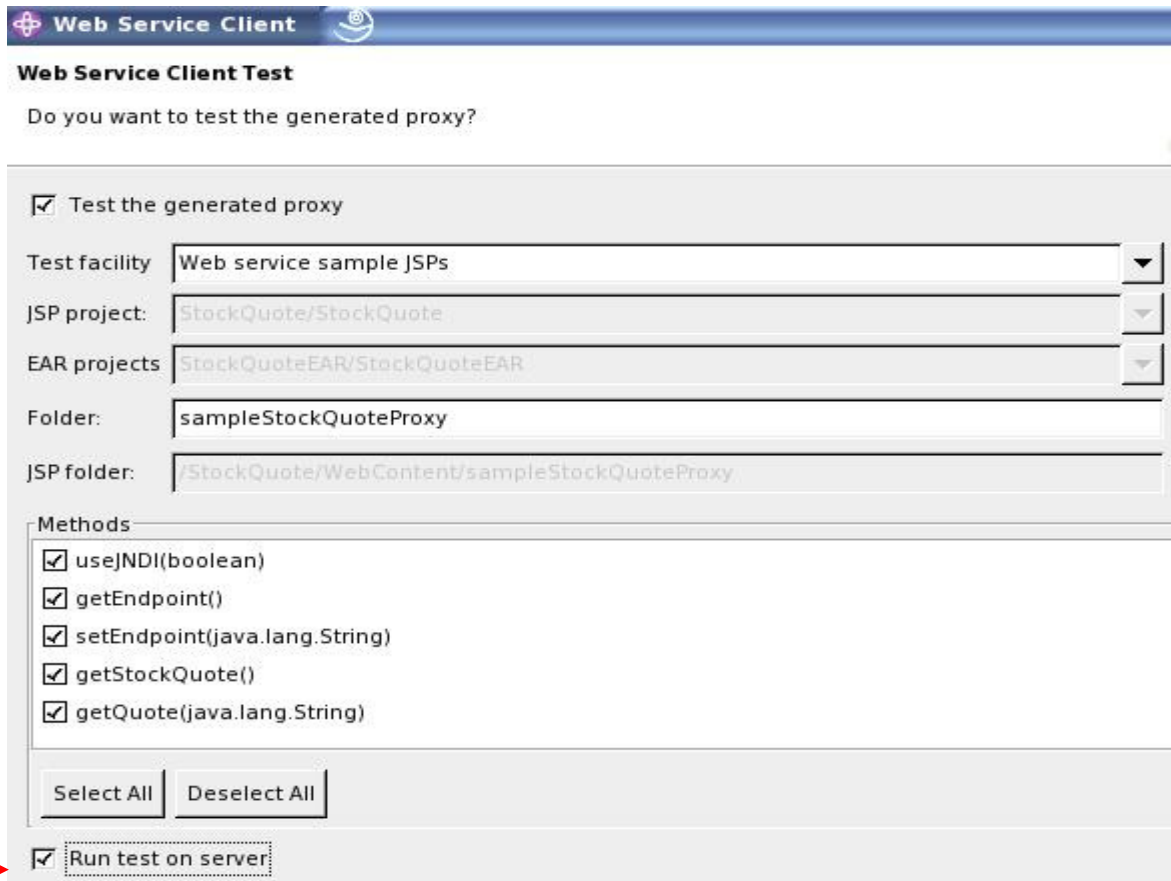
- ___ e. From the Web Service client wizard, make an additional section of '**Test the Web Service**'. Click **Next**.



- ___ f. On the Web Service Selection page, click **Next**
- ___ g. On the Client Environment Configuration, click **Next**
- ___ h. On the Web Service Proxy page, click **Next**.
- ___ i. You will get a **Warning** before you can continue. Answer "**Yes to All**" to enable automatic file overwriting when creating the files.



- ___ j. On the Web Service Client Test page, make an additional selection of **Run test on server**. Click **Finish**



Web Service Client

Web Service Client Test

Do you want to test the generated proxy?

☒ Test the generated proxy

Test facility: Web service sample JSPs

JSP project: StockQuote/StockQuote

EAR projects: StockQuoteEAR/StockQuoteEAR

Folder: sampleStockQuoteProxy

JSP folder: /StockQuote/WebContent/sampleStockQuoteProxy

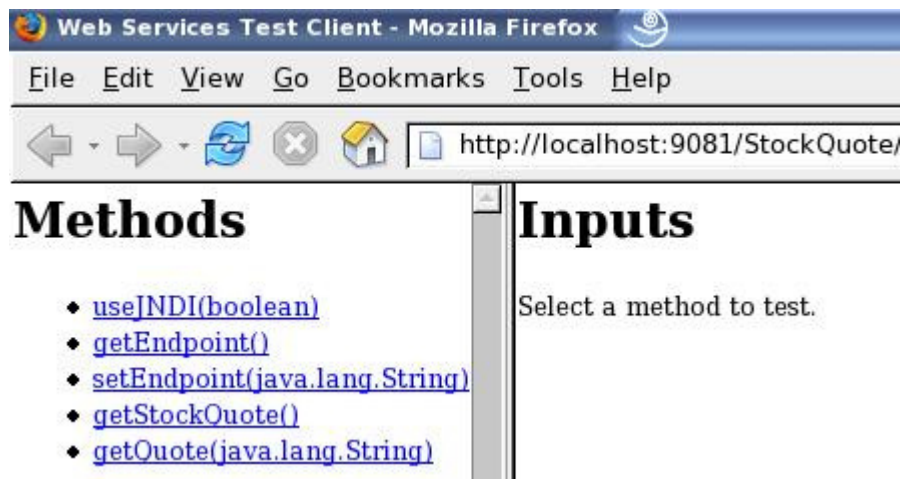
Methods

- ☒ useJNDI(boolean)
- ☒ getEndpoint()
- ☒ setEndpoint(java.lang.String)
- ☒ getStockQuote()
- ☒ getQuote(java.lang.String)

Select All Deselect All

☒ Run test on server

___ k. Once the proxy is generated, the **Web Services Test Client** will start in a browser.



___ l. In the **Methods** section, click on the **setEndpoint(java.lang.String)** method.

___ m. In the **Inputs** section, enter an endpoint of <http://localhost:9082/StockQuote/services/StockQuote>

Inputs

endpoint:

- ___ n. Click the **Invoke** button
- ___ o. Returning to the **Methods** section, click on the **getEndpoint()** method.
- ___ p. In the **Inputs** section, click **Invoke**. This should return <http://localhost:9082/StockQuote/services/StockQuote> in the **Result** section
- ___ q. Returning to the **Methods** section, click on the **getQuote (java.lang.String)** method
- ___ r. In the **Inputs** section, enter a stock symbol and click **Invoke**.

Inputs

arg_0_0:

- ___ s. In the **Result** section, examine the results

Result

26.605368

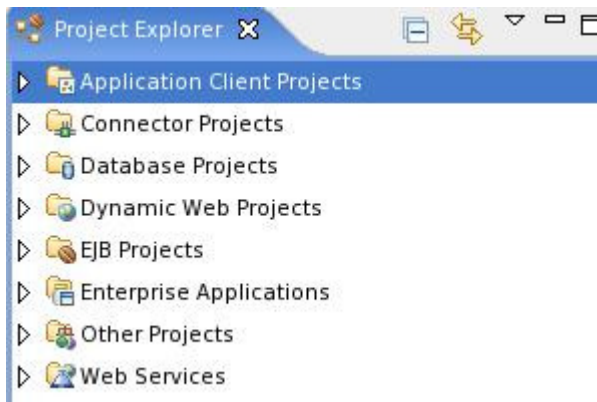
- ___ t. In the AST, the **TCP/IP Monitor** view is still active where you may also examine the results.
- ___ u. **Close** the browser when you are finished.

Part 6: Testing with a J2EE application client

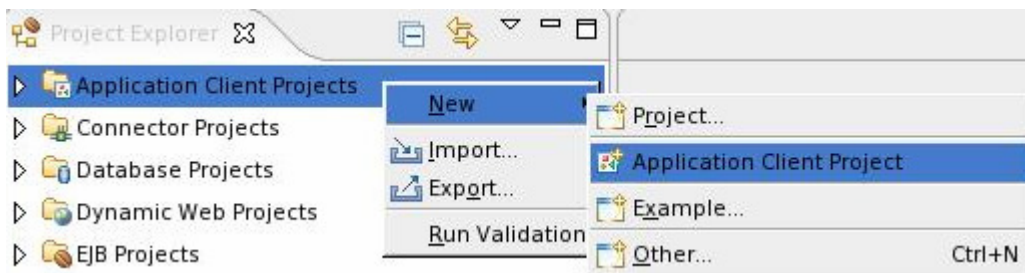
J2EE application clients are like regular Java™ applications. They contain a `main()` method that is executed, and they continue executing until the client virtual machine terminates. They can be run as typical "fat client" applications, to display a GUI that connects to a set of EJBs for persistence and business logic, or as server applications that provide services over the network. However, a J2EE application client has several advantages over regular Java applications, because it runs within a lightweight server container. This container can provide the application client with services that used to be available only to other J2EE components.

____ 1. Create a J2EE application client project using StockQuote

____ a. From the Project Explorer view, select **Application Client Projects**



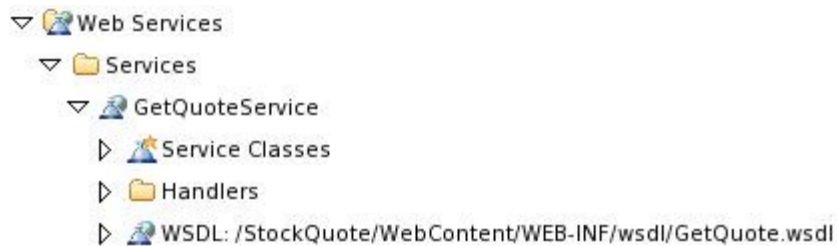
____ b. Right-click and select **New → Application Client Project**



____ c. Enter a Project Name of **StockQuoteClient**. Click **Finish**.



- ___ d. From the Project Explorer view, locate the **Web Services** section. Expand **Web Services** → **Services** → **GetQuoteService**. This is convenient section to quickly examine your Web services.

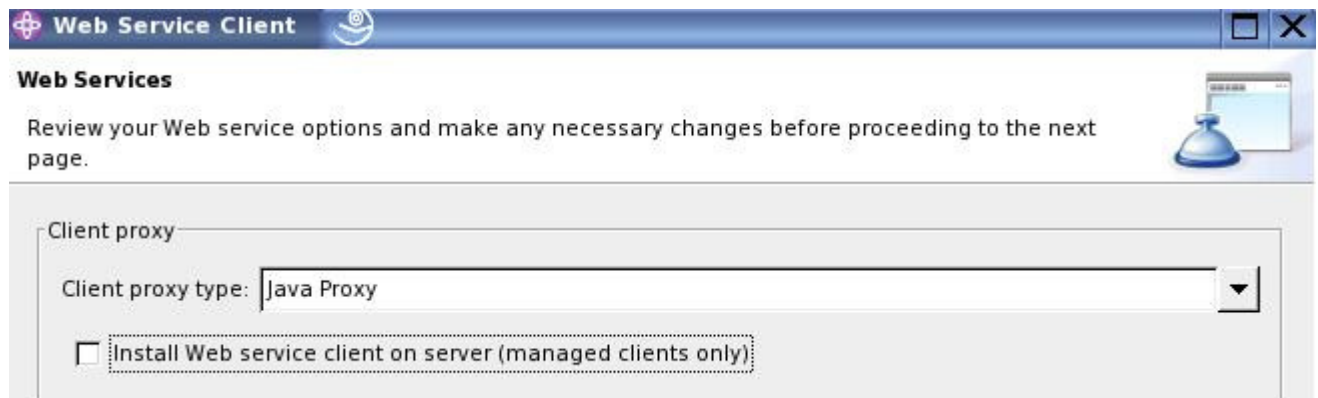


- ___ e. Select **WSDL: /StockQuote/WebContent/WEB-INF/wsdl/GetQuote.wsdl**



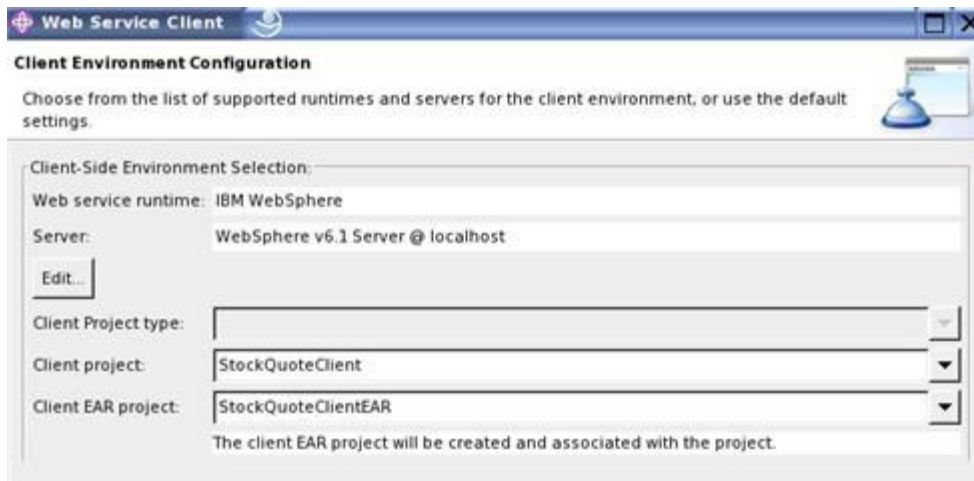
- ___ f. Right-click and select **Generate Client**

- ___ g. On the Web Service Client options, **uncheck** *Install Web service client on server*. Click **Next**



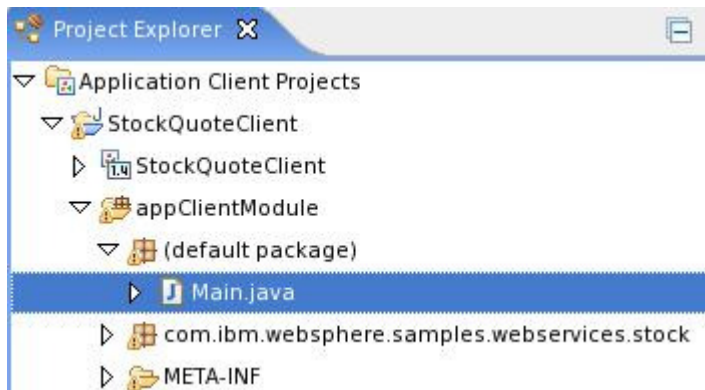
- ___ h. On the Web Service Selection Page, click **Next**

- ___ i. From the Client Environment Configuration panel, use the pulldown menu to select a Client Project of **StockQuoteClient**. The Client EAR project will change to **StockQuoteClientEAR**. Click **Finish**.



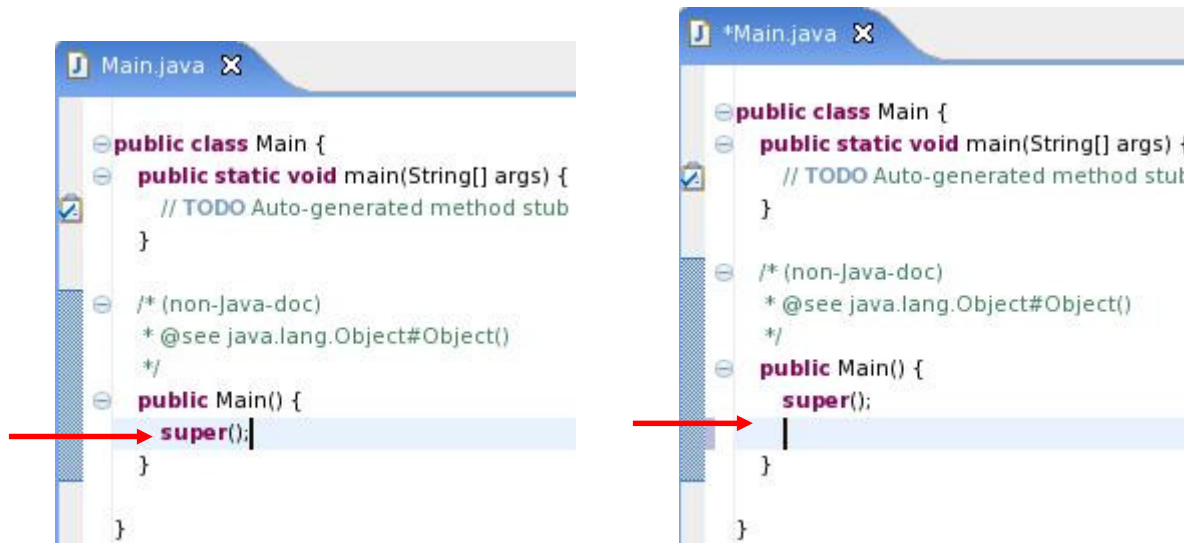
___ 2. Develop the StockQuote client using a Snippet

- ___ a. From the Project Explorer view, locate the Application Client Projects. Expand **StockQuoteClient** → **StockQuoteClient** → **appClientModule** → **(default package)** and select **Main.java**

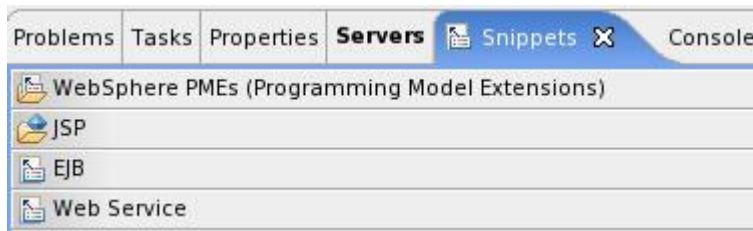


- ___ b. Double-click on **Main.java** to open it in the Java editor

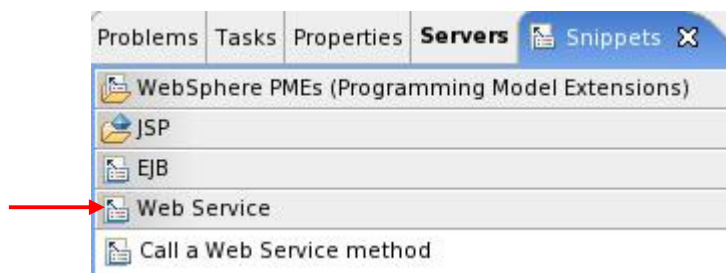
- ___ c. Locate the line **super ()** ; Position the cursor at the end of this line and press **Enter** to create a new line.



- ___ d. Locate the **Snippets** view at the bottom of the J2EE perspective

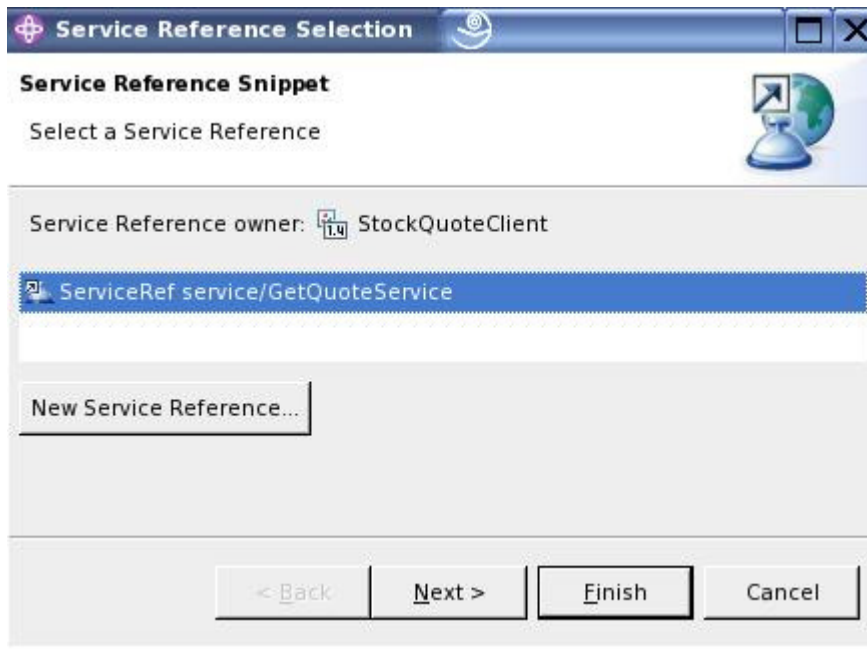


- ___ e. In the Snippets view, click on **Web Service**

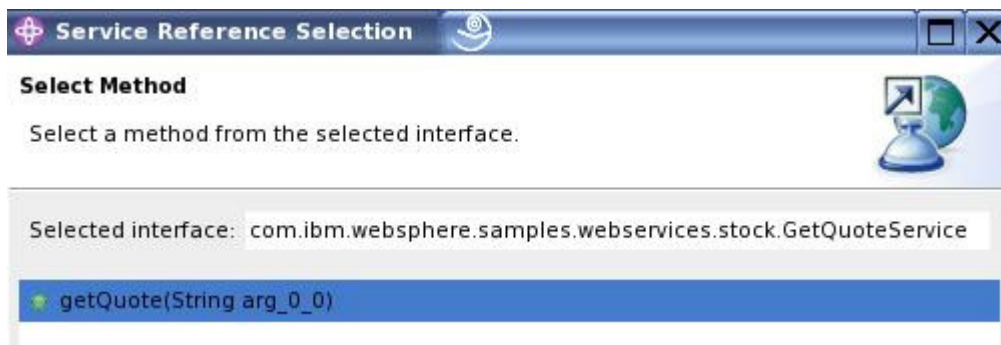


- ___ f. Double-click on **Call a Web Service method**

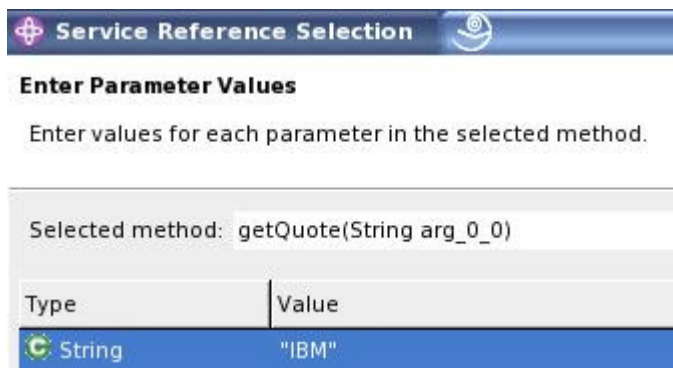
___ g. In the Select a Service Reference panel, ensure **service/GetQuoteService** is selected. Click **Next**



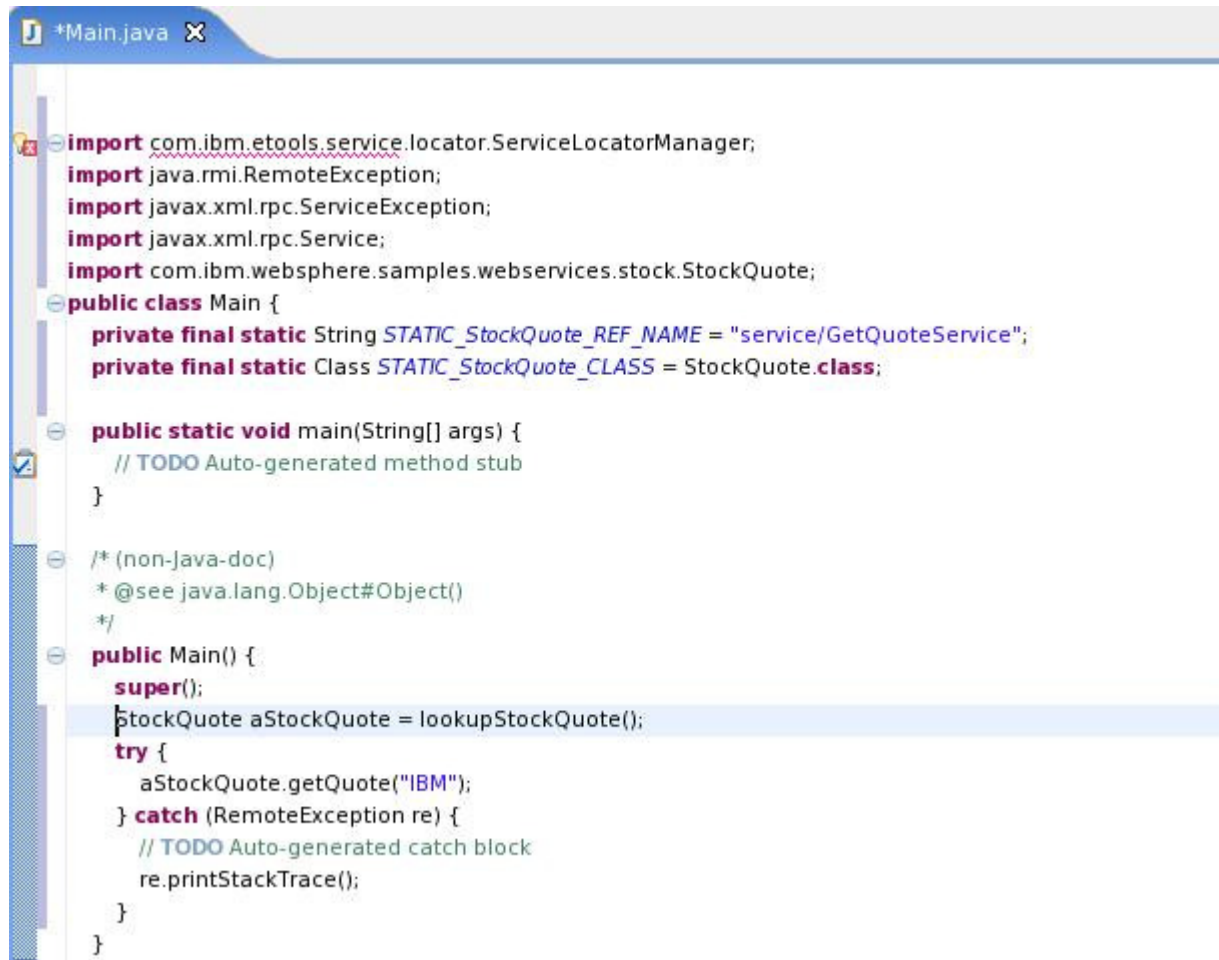
___ h. In the Select Method panel, click **Next**



___ i. In the Enter Parameter Values panel, enter a value of **"IBM"** (include the double-quotes) and click **Finish**



__j. The code required to access the StockQuote Web service has been automatically generated.



```

+Main.java X
import com.ibm.etools.service.locator.ServiceLocatorManager;
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import javax.xml.rpc.Service;
import com.ibm.websphere.samples.webservices.stock.StockQuote;

public class Main {
    private final static String STATIC_StockQuote_REF_NAME = "service/GetQuoteService";
    private final static Class STATIC_StockQuote_CLASS = StockQuote.class;

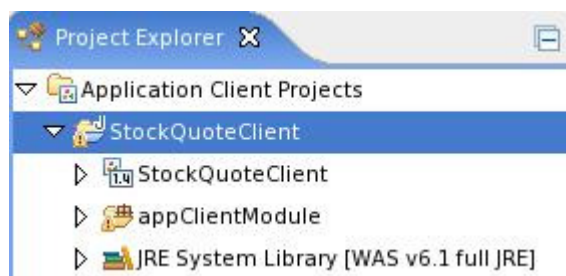
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }

    /* (non-java-doc)
     * @see java.lang.Object#Object()
     */
    public Main() {
        super();
        StockQuote aStockQuote = lookupStockQuote();
        try {
            aStockQuote.getQuote("IBM");
        } catch (RemoteException re) {
            // TODO Auto-generated catch block
            re.printStackTrace();
        }
    }
}

```

__k. **Note:** The code generated by the snippet requires a **serviceLocatorMgr.jar** file. This is a utility JAR file supplied by the AST. The **serviceLocatorMgr.jar** file includes a `ServiceLocatorManager` class that is used within the inserted snippets of Java code. This class optimizes the lookups of the home interfaces and InitialContexts, and ensures that they are only looked up once for the entire application. The **serviceLocatorMgr.jar** file should automatically be added to the application client project, but due to a defect in the AST, you will now manually add it to the project. This issue will be addressed in a future fix pack.

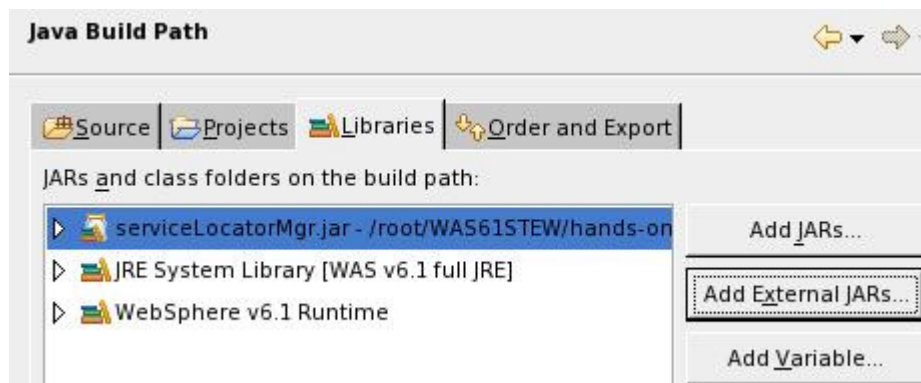
__l. Returning to the Project Explorer view, select the **StockQuoteClient** project.



- ___ m. Right-click and select **Properties**
- ___ n. From the Properties window, select **Java Build Path**
- ___ o. From the Java Build Path Workspace, click on the **Libraries** tab. Click on the **Add External JARs** button.

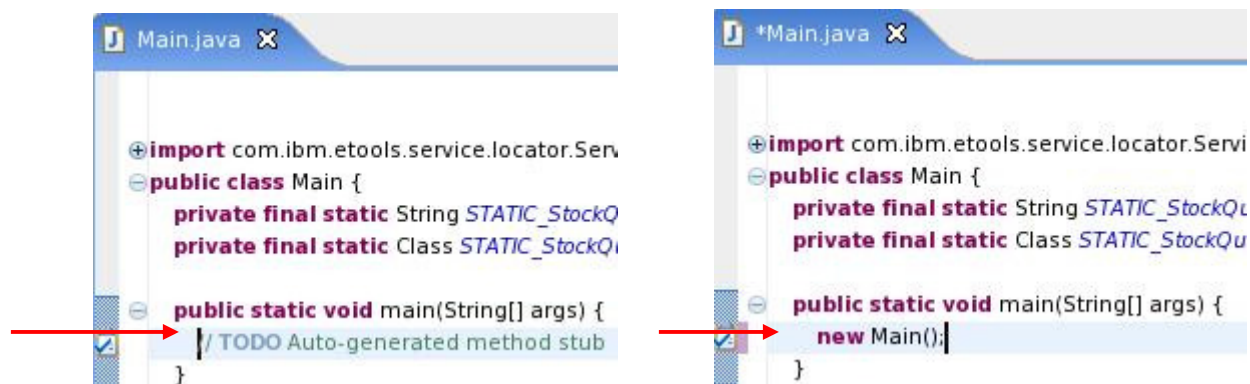


- ___ p. From the JAR selector dialog, choose `/root/WAS61STEW/hands-on/Lab05-WebServices/serviceLocatorMgr.jar` and click **OK**



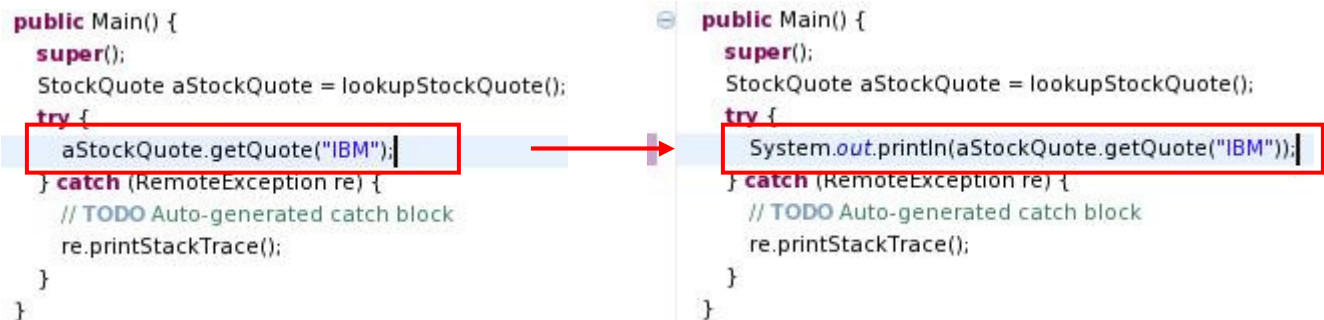
- ___ q. Click **OK** to close the Preferences window. This should fix all errors in **Main.java**.
- ___ r. In the **Main.java** file, locate the `//TODO Auto-generated method stub` line. This is the first TODO in the file. Refer to the picture below for reference. Replace this line with the following text:

```
new Main();
```



- ___ s. In the **Main.java** file, locate the `aStockQuote.getQuote("IBM");` line. Refer to the picture below for reference. Change this line to read:

```
System.out.println(aStockQuote.getQuote("IBM"));
```

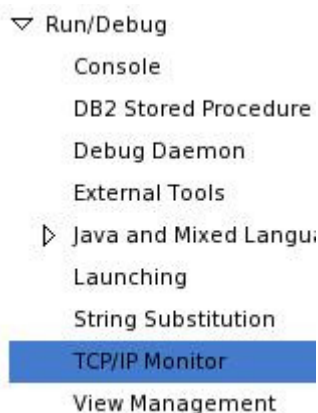


- ___ t. Use **CTRL+S** to save the file
- ___ u. Close the **Main.java** editor.

3. Test the J2EE StockQuote client

You will now test the J2EE application client. The default Web service endpoint is 9080. To avoid changing the Web service endpoint to 9081 programmatically, you will configure a second Monitoring port. If you are using a WebSphere profile with different port numbers, adjust accordingly.

- ___ a. From the AST main menubar, select **Window → Preferences**
- ___ b. From the Preferences window, expand **Run/Debug** and select **TCP/IP Monitor**.



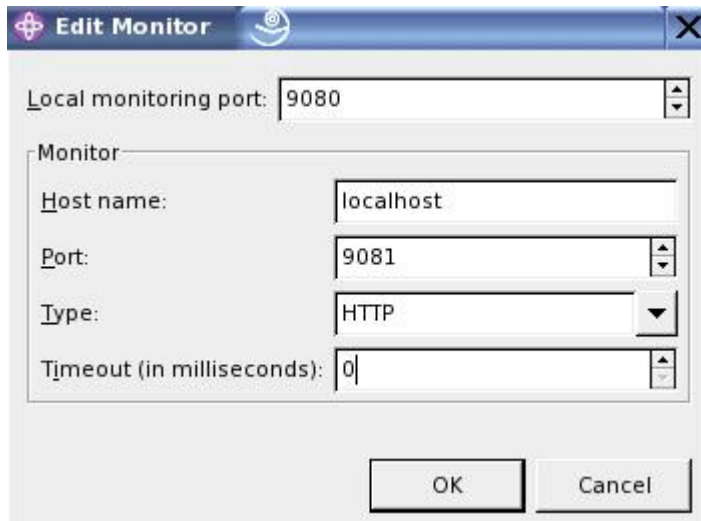
___ c. In the TCP/IP Monitor Workspace, click on the **Add...** button

___ d. Enter the following values:

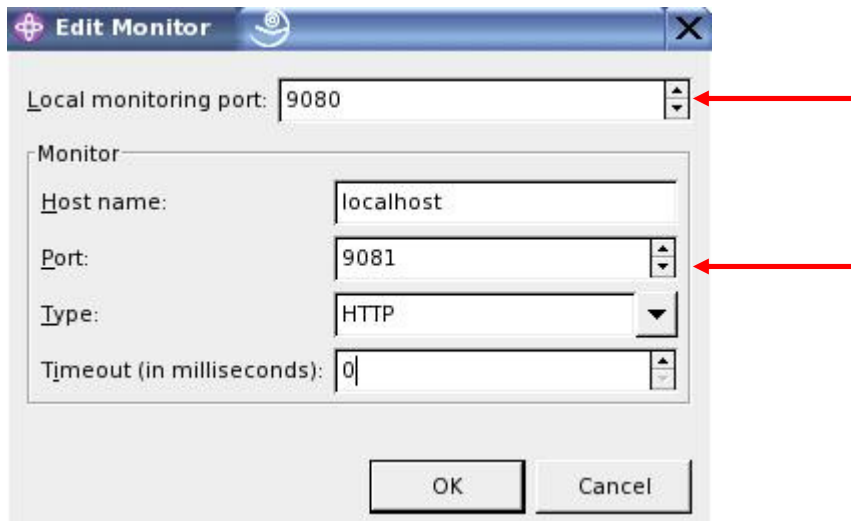
Local monitoring port: 9080

Host name: localhost

Port: 9081

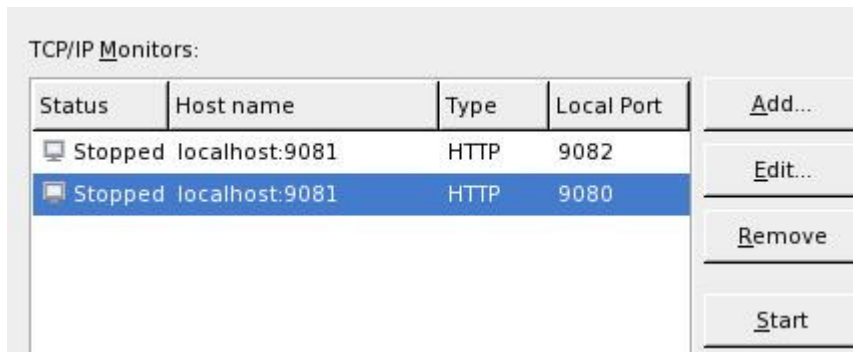


___ e. Before clicking on the **OK** button, **click once** on the up and down arrow buttons for the **Local Monitoring port** and **Port**. The ports will sometimes not get the correct values if you do not do this step.

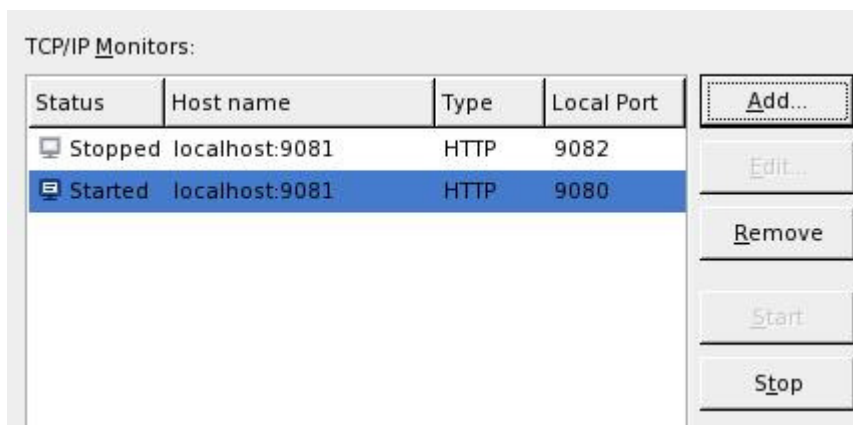


___ f. Click **OK**

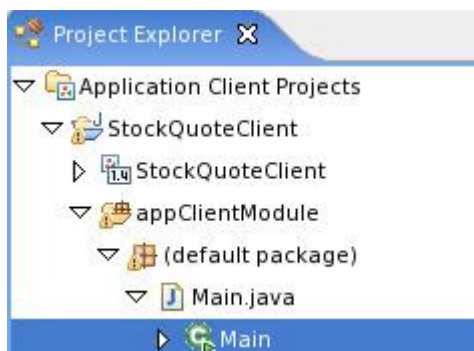
- ___ g. In the TCP/IP Monitors workspace, ensure new entry is correct. The Hostname is `localhost:9081` and Local Port is 9080. If the entry is incorrect, use the **Edit...** to correct the problem. Highlight the new entry.



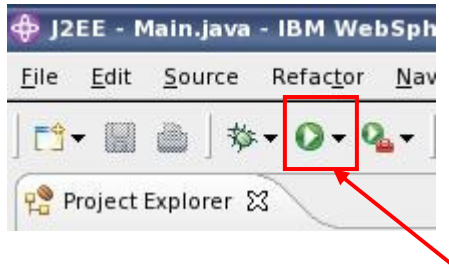
- ___ h. Click the **Start** button and the Status should change to started.



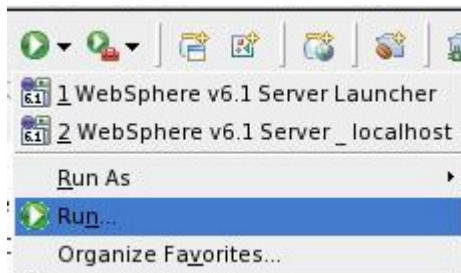
- ___ i. Click **OK** to close the Preferences window.
- ___ j. From the Project Explorer view, expand **Main.java** and select **Main**



___ k. From the main toolbar, click on the **Run icon** pulldown



___ l. Select **Run...**



___ m. From the Run configuration, select **WebSphere v6.1 Application Client**



___ n. Below **WebSphere v6.1 Application Client**, locate and click the **New** button.

- ___ o. Enter a name of **StockQuoteClient**. Select **Enable application client to connect to a server**. Select **Use provider URL** and enter **iiop://localhost:2811**

(The port number varies based on your profile. If you do not know your port number, go to the Servers view and double-click on your server. Use the ORB bootstrap port or SOAP connector port)

The screenshot shows the 'Application Client' configuration window in WebSphere. The 'Name' field is set to 'StockQuoteClient'. The 'WebSphere Runtime' is 'WAS v6.1 full'. The 'Enterprise application' is 'StockQuoteClientEAR'. The 'Application client module' is 'StockQuoteClient'. The checkbox 'Enable application client to connect to a server' is checked. Under this checkbox, the radio button 'Use provider URL' is selected, and the text field below it contains 'iiop://localhost:2811/'.

- ___ p. Click **Apply**
- ___ q. Click **Run**
- ___ r. From the **Console** view, the client will run and show the Stock Quote value:

The screenshot shows the 'Console' tab in the WebSphere IDE. The log output is as follows:
<terminated> StockQuoteClient [WebSphere v6.1 Application Client] /opt/IBM/WAS6
WSDL0901: Component initialization completed successfully.
WSDL0035: Initialization of the J2EE Application Client Environment has completed.
WSDL0014: Invoking the Application Client class Main
51.603554
A red arrow points to the final line of the log output, '51.603554'.

Congratulations, you have now finished Lab05.

In this exercise you used the *Application Server Toolkit (AST)* to perform development and assembly tasks for Web services. You created and validated WSDL files using the graphical editor, used the Web service wizard to create Web services and clients from a variety of resources, and tested Web services using the Web services Explorer and the Universal Test Client.