Introduction

Time required

To complete this tutorial, you will need approximately **1 hour and 30 minutes**. If you decide to explore other facets of Web services or EJBs while working on the tutorial, it could take longer to finish.

Prerequisites

In order to complete this tutorial end to end, you should be familiar with:

- Basic Web services concepts, such as SOAP and WSDL
- Basic XML
- EJB programming

Learning objectives

You will learn how to create a WS-I compliant Web service from a WSDL file. The wizard generates an EJB skeleton that contains a set of methods corresponding to the operations described in the WSDL document. When the EJB is created, each method has a trivial implementation that can be easily replaced by editing the EJB.

When you are ready, begin Exercise 1.1: Set up the workspace and create the required projects

Exercise 1.1: Set up the workspace and create the required projects

Enabling Web service capabilities

To enable the capabilities required for Web services development:

1. On the Welcome page, check to see if Web services are enabled by looking for the Web services icon in the lower right-hand corner:  . If the Welcome page has been closed, you can reopen it from the Help menu.
2. If Web services are not enabled, select the icon in the lower right corner that looks like a person. This will generate a list of capabilities that you can select from.
3. Select the Web services icon in the top left corner:

You have now enabled the tools used in Web services development.

Setting the WS-I compliance level

WS-I refers to Web service interoperability; this includes interoperability across platforms, operating

systems, and programming languages. The WS-I organization sets out standards collected in documents called Profiles that define the requirements needed to make a Web service interoperable. The Rational Developer products validate Web services against the WS-I Simple SOAP Binding Profile 1.0 (WS-I SSBP) and the WS-I Attachments Profile 1.0 (WS-I AP). For more information on WS-I, refer to their Web site: http://www.ws-i.org/

By default, the WS-I SSBP compliance level is set to **Suggest**. With this setting, a warning dialog box will appear if any non-compliant choices are made, but you will still be able to continue. This compliance level is used by the Web service wizards and the WSDL validation tool. This sample generates a WS-I compliant Web service, therefore you should set the WS-I compliance level to **Require**.

To change the WS-I compliance level:

1. On the main menu bar, click **Window** > **Preferences**. The Preferences dialog box opens.
2. Expand the **Web Services** branch and select **WS-I Compliance**.
3. Select the **Require** option from the drop-down list beside WS-I SSBP.
4. Click **OK**.

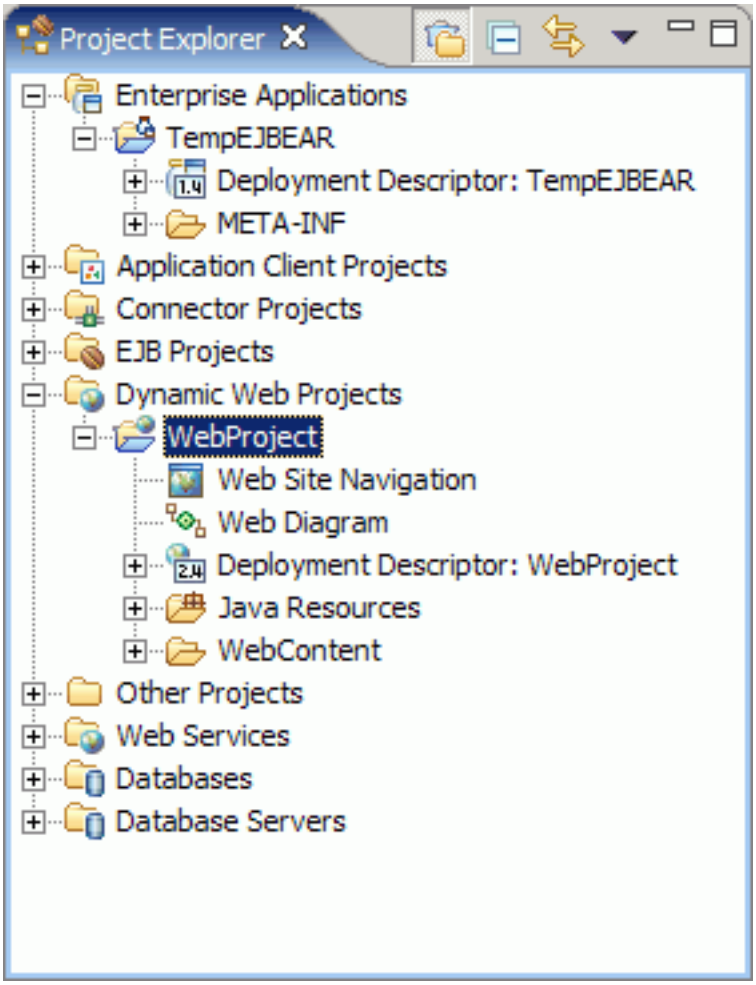
Creating the Web project

The remaining steps in this tutorial will be done in the J2EE perspective. If you are asked if you want to change to another perspective after performing a task, select **No**.

In WebSphere Studio, Web services must be contained in a Web project or an EJB project. For this particular tutorial, a Web project is used since the EJB skeleton deploys a Web service in the Web container. All resources required by the Web service, including your WSDL document, EJB and Web settings, are kept in this project.

To create a Web project:

1. On the main menu bar, click **File** > **New** > **Project...** > **Web** > **Dynamic Web Project**. Click **Next**.
2. Type `WebProject` in the **Name** text field. Click **Advanced**, and note that the target server is WebSphere Application Server v6. In the EAR project field, enter TempEJBEAR as the EAR name. This will ensure that the EJB that you will crate later and your router project are both referenced in the same EAR. Click **Finish**.
3. You have now created your Web project.



Adding the project to the server

You need to associate the project with the server that your Web service will run on. To do this:

1. Right-click the server in the Servers view and select **Add and Remove projects**. If the Servers view is not open in your workspace, open it from the **Window** menu by selecting **Show View** > **Servers**.
2. In the window that opens, select your dynamic Web project and its associated EAR file, and click **Add**.
3. Click **Finish**.

Now you are ready to begin Exercise 1.2: Import and validate the WSDL file.

Exercise 1.2: Import and validate the WSDL file

Before you begin, you must complete Exercise 1.1: Set up the workspace and create the required projects.

Importing the Temperature Conversion WSDL document

The Temperature Conversion WSDL document has been provided for you. The WSDL file that you will use in this tutorial converts temperature from Fahrenheit to Celsius and Celsius to Fahrenheit.

You can create a new WSDL document in WebSphere Studio or import an existing one. The Temperature Conversion WSDL document used in this tutorial has been provided for you. Complete the following steps to import the Temperature Conversion WSDL document into the workbench:

1. In the Project Explorer view, select **Dynamic Web Projects** > **WebProject** > **WebContent**.
2. Right-click the **WebContent** folder and click **New** > **Folder** to create a folder called `wsdl`. Click **Finish**.
3. Click **File** > **Import** to open the Import wizard.
4. Click **File system** to import the resources from the local file system. Click **Next**.
5. Click **Browse** to locate and select the following folder which is located in the Rational Developer product's install directory: **com.ibm.etools.webservice.tdejb.tutorial.doc\resources**.
6. Click **OK**.
7. Select the **ConvertTemperature.wsdl** check box.
8. Click **Finish** to import the WSDL document and close the wizard.

Validating the WSDL document

WebSphere Studio provides a tool called the WSDL Validator for validating WSDL semantics and WS-I compliancy.

To validate the Temperature Converter WSDL document:

1. Select the ConvertTemperature.wsdl document in the Project Navigator.
2. Right-click, and click **Validate WSDL file**. A window is displayed stating that the WSDL document is valid. Any errors will be displayed in the Tasks view.

If no errors occur during validation, you can proceed to create the Web service.

Now you are ready to begin Exercise 1.3: Create the Web service.

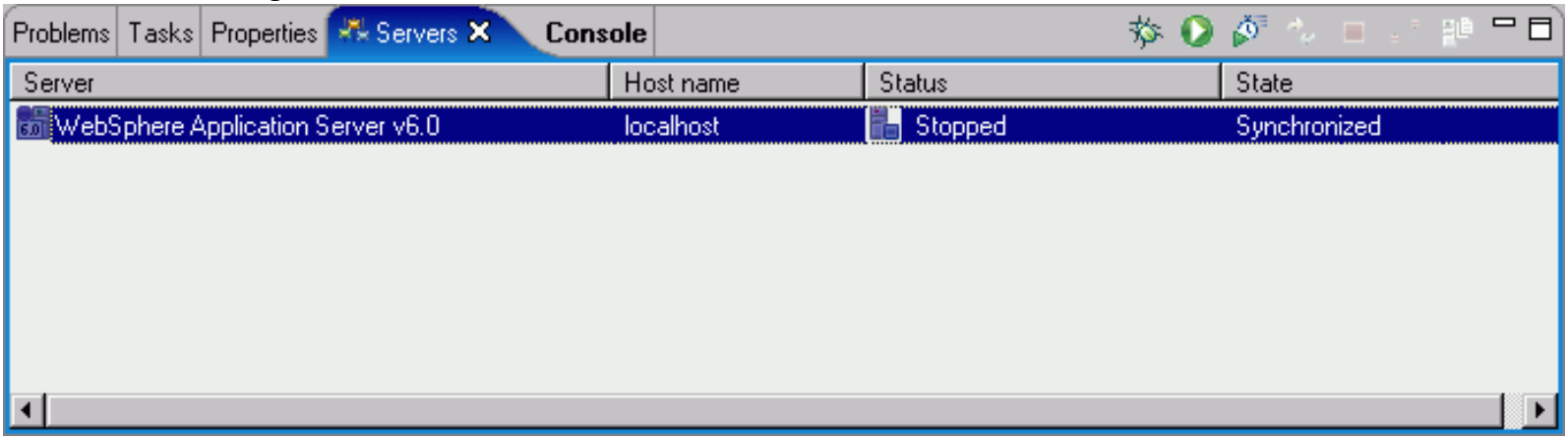
Exercise 1.3: Create the Web service

Before you begin, you must complete Exercise 1.2: Import and validate the WSDL file.

Starting the server

Before you attempt to create a Web service it is strongly suggested that you start the WebSphere Application Server on which the Web service will run. Although you can start the server in the Web service wizards, since it may take several minutes to start depending on the speed of your machine, starting the server before you begin will both increase the speed with which you complete the wizard and reduce the chance that the wizard will generate an error because the server is taking too long to start.

To start the server, right-click the server in the Servers view and select **Start**:



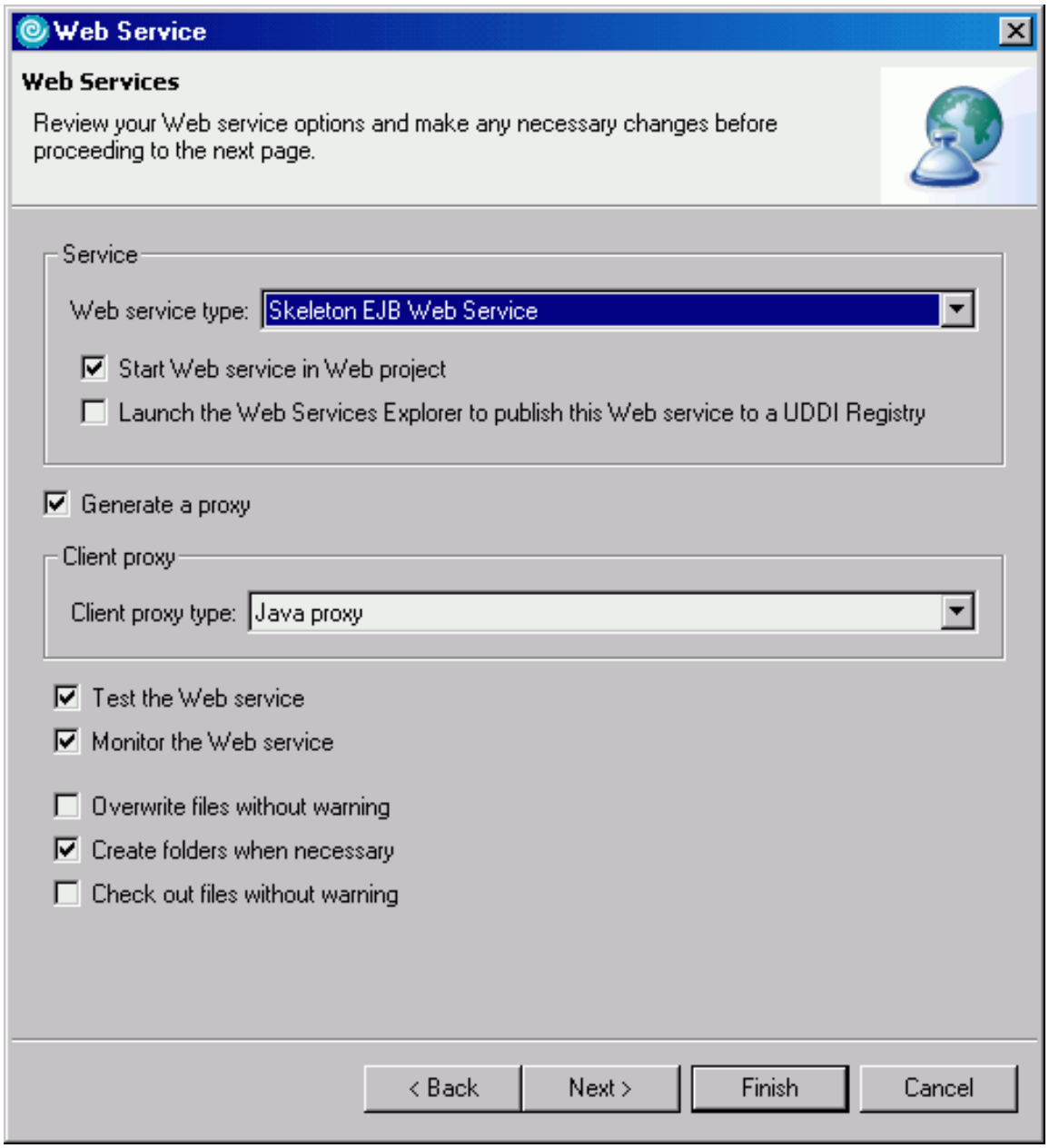
If the Servers view is not open in your workspace, open it from the **Window** menu by selecting **Show View** > **Servers**.

Create a Web service from a WSDL file

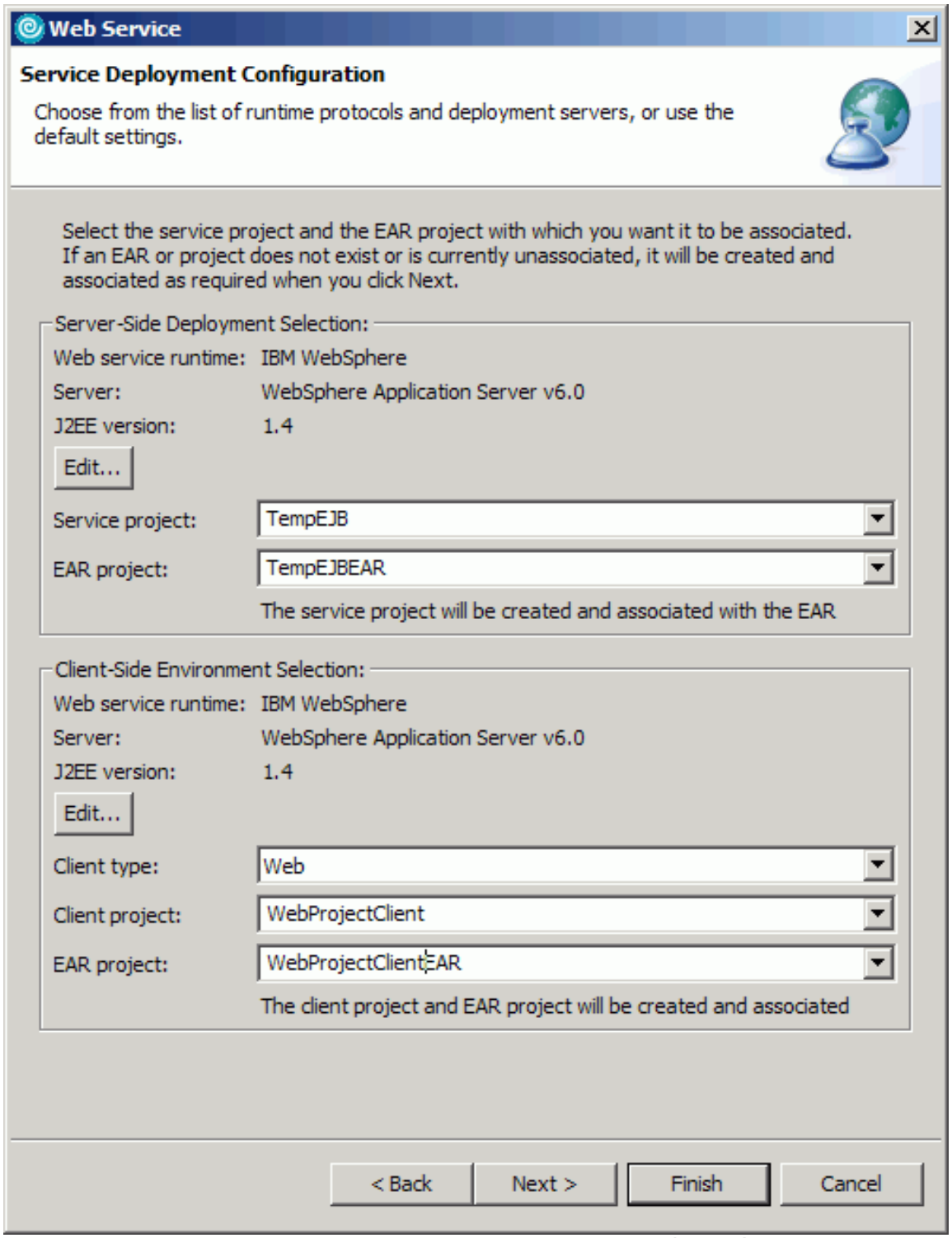
The Web Service wizard assists you in creating a new Web service, configuring it for deployment, and deploying the Web service to a server. Once your Web service is deployed, the wizard assists you in generating the client proxy and sample application to test the Web service. When you have completed testing, you can publish your Web service to a UDDI Business Registry using the Export wizard.

1. In the Project Explorer view, select the **ConvertTemperature.wsdl** document in your Web project.
2. Click **File** > **New** > **Other**. Select **Web Services** in order to display the various Web service wizards. Select the **Web Service** wizard. Click **Next.**
3. In the Web Services panel, select **Skeleton EJB Web Service** as your Web service type. Also select the following:
   - Generate a client proxy to the Web service. The EJB client proxy that is generated provides a remote procedure call interface to the Web service.
   - Test the Web service.
   - Monitor the Web service.

   The following selections should be selected for this example:



4. On the Object Selection page, the WSDL file should be pre-filled. If not, browse to the WSDL file that you imported.
5. In the Service Deployment Configuration page, you can specify the deployment settings. The IBM WebSphere run-time environment is set as the default run-time environment in both server-side and client-side configurations.
    a. In the Server-Side Deployment Selection section, type `TempEJB` in the **Server Project** field and `TempEJBEAR` in the server EAR field.
    b. In the Client-Side Environment Selection section, leave the client type as Web project, enter `WebProjectClient` for the **Client Project** field, and enter `WebProjectClientEAR` as the Client project EAR. Generating the service and client projects into different EARs will reduce the likelihood of encountering errors at run time.
    c. Click **Next.**



6. On the Web services skeleton EJB configuration page, type `WebProject` as your router project.
7. In the Web Service Test page, you can select a test facility to test your Web service before a client or proxy is developed. Select Web Services Explorer as the test facility for your Web service and click **Launch**. This step may take several seconds for the WebSphere Application server to start.
8. The Web Services Explorer is displayed in a Web browser. Select **fahrenheitToCelsius** or **celsiusToFahrenheit** from the operations list. Enter a number in the value field and click **Go**. A trivial implementation of each of these operations is provided, and a default value of `-3` is returned. If both operations complete successfully, close the browser window and click **Next** in the Web services wizard.
9. In the Web Service Proxy page, ensure **Generate proxy** is checked. Keep the Security Configuration selection at `No Security` to remain WS-I compliant. Click **Next.**
10. In the Web Service Client Test page, ensure **Test the generated proxy** and **Run test on server** are both selected. In the Methods section. ensure that all methods are selected, or click **Select All** to select all methods. If you want to

publish your Web service to a UDDI Registry, click **Next** to configure the Web Service Publication options. However this step will not be covered in this tutorial. Otherwise, click **Finish.**

11. The sample application is launched in a Web browser. You can use this application to test the Web service by selecting a method in the Methods frame, entering an input value in the Inputs frame, and clicking **Invoke** to view the result in the Result frame. Do not close the TestClient.jsp browser window yet - it will be used to test the Web service traffic for WS-I compliance later in this tutorial.

Now you are ready to begin Exercise 1.4: Implement the temperature conversion methods.

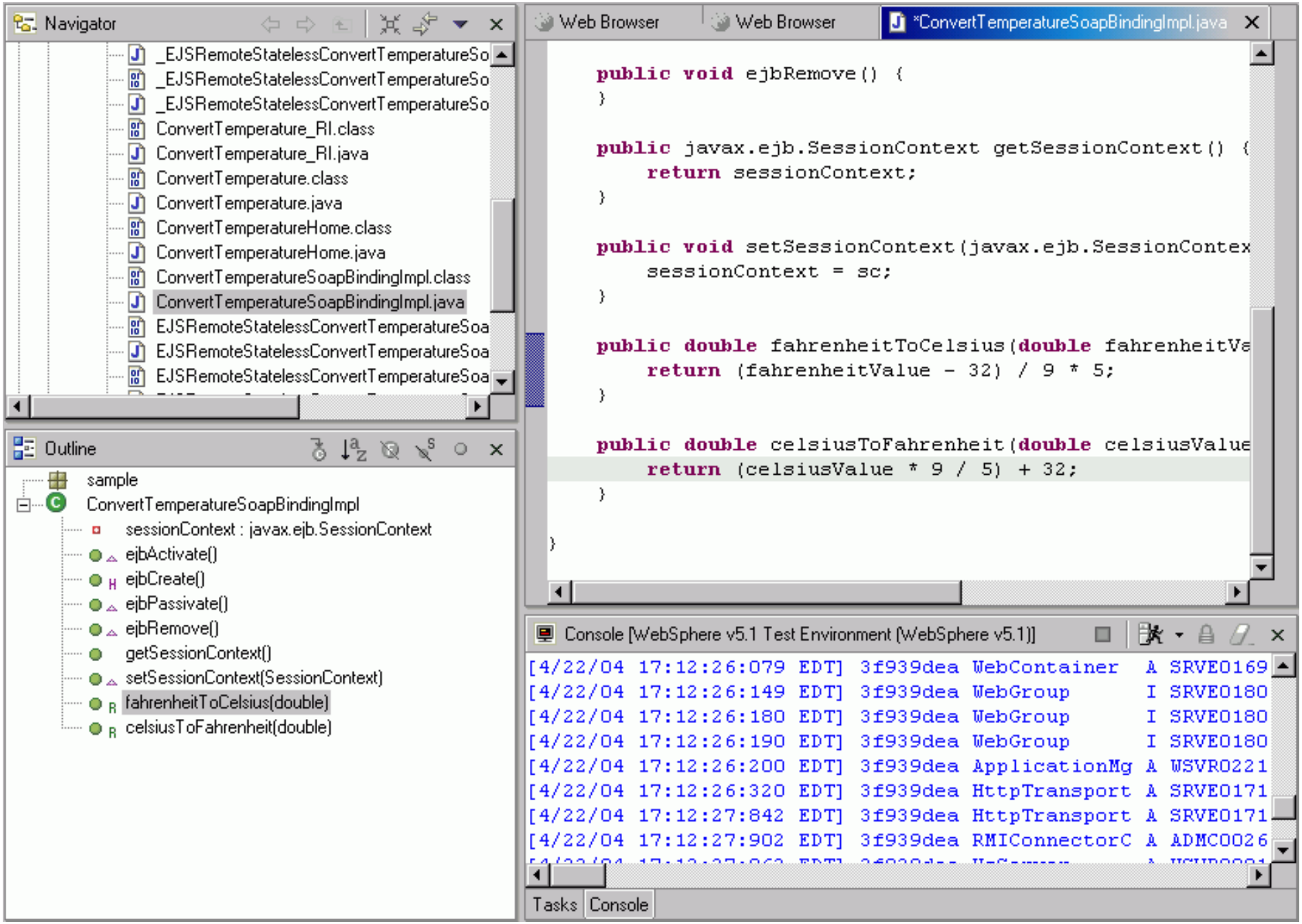
Exercise 1.4: Implement the temperature conversion methods

Before you begin, you must complete Exercise 1.3: Create the Web service.

Implementing the temperature conversion methods (optional)

Trivial implementations of the **fahrenheitToCelsius** and **celsiusToFahrenheit** methods were automatically generated when you created a Web service from your WSDL document. In this section you will replace these trivial implementations with more meaningful code, and perform the necessary steps to test your new methods.

1. In the Project Explorer view, select **ConvertTemperatureSoapBindingImpl.java** under **TempEJB** > **ejbModule** > **samples**.
2. Locate the `fahrenheitToCelsius` method and replace the current implementation with the following: `return (fahrenheitValue - 32) / 9 * 5;`
3. Locate the `celsiusToFahrenheit` method and replace the current implementation with the following: `return (celsiusValue * 9 / 5) + 32;`



4. Save your updates by clicking **File > Save**.
5. Restart the EAR by right-clicking **WebSphere Application Server v6.0** in the Servers view and clicking **Restart Project > TempEJBEAR**.
6. Click **Run** > **Launch the Web Services Explorer** from the main menu bar and repeat the instructions from the previous section to test your **fahrenheitToCelsius** and **celsiusToFahrenheit** methods.

Now you are ready to begin Exercise 1.5: Validate the Web service traffic WS-I compliance.

Exercise 1.5: Validate the Web service traffic WS-I compliance

Before you begin, you must complete Exercise 1.4: Implement the temperature conversion methods.

Validating the Web service traffic WS-I compliance

To ensure that the SOAP envelope request and response pairs are WS-I compliant, you need to direct your Web service traffic through the TCP/IP Monitor:

When creating a Web service using the Web service or Web service client wizards, you can select to set up and run the TCP/IP Monitor automatically. Since you chose this option when creating the Web service, the TCP/IP monitor view should be in your workspace. If it is not, you can open this view by selecting **Window** > **Show View** > **Other** > **Debug** > **TCP/IP Monitor**.

Alternately, you can set up the TCP/IP Monitor manually by completing the following steps:

1. In the sample application, invoke the getEndPoint method. Record this endpoint. The default endpoint for a Web service is:
   - WebSphere or Apache Axis run-time environment: http://localhost:<port>/<web module context root>/services/<port>
   - IBM SOAP run-time environment: http://localhost:<port>/<web module context root>/servlet/rpcrouter.
2. Create a server to act as the TCP/IP Monitor:
   a. From the **Window** menu, select **Preferences**.
   b. In the Preferences window, expand **Internet** and then select **TCP/IP Monitor**.
   c. Select the **Show TCP/IP Monitor View when there is activity** check box.
   d. Under the TCP/IP Monitors lists, click **Add**. A New Monitor dialog box opens.
   e. Specify the following settings:

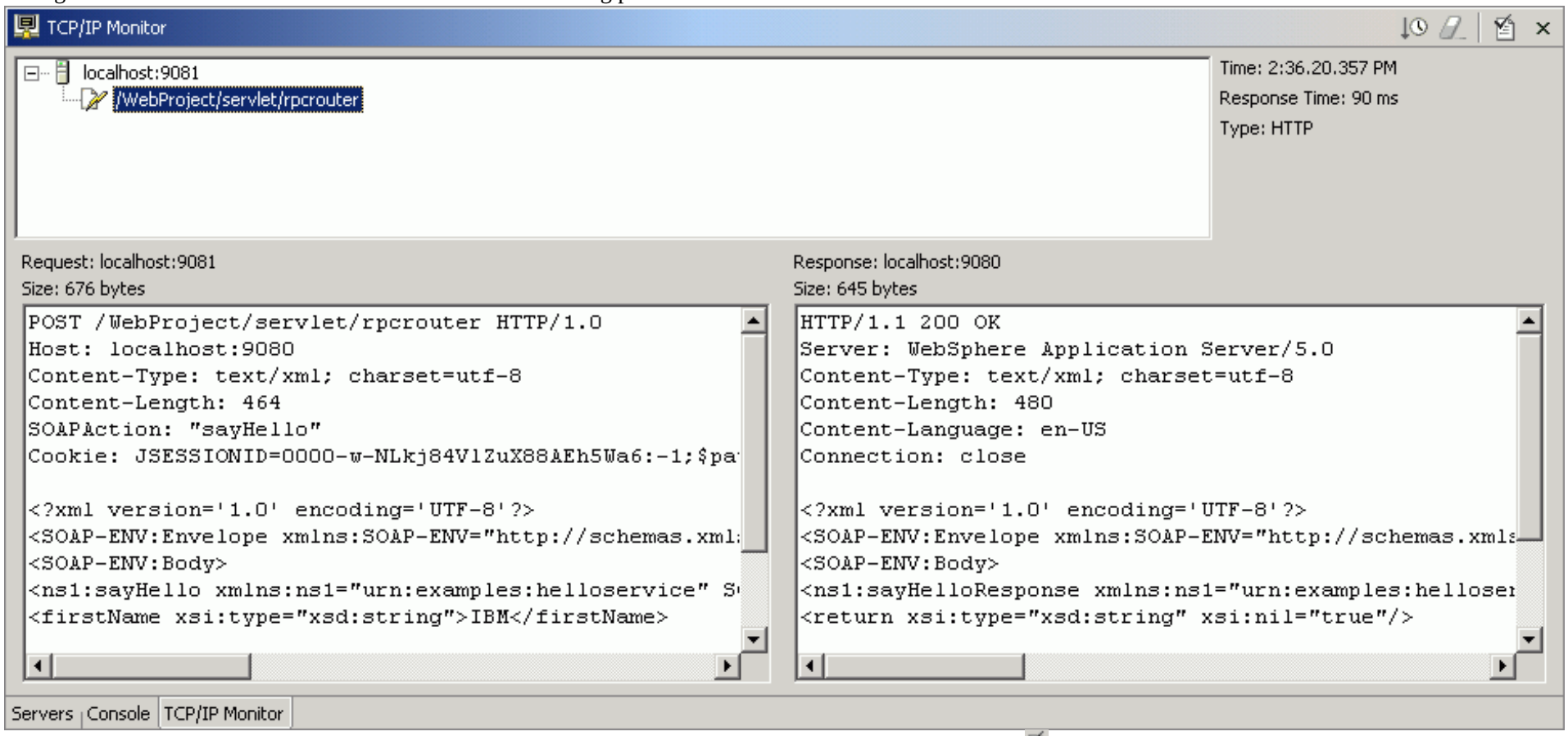
Option	Description
Local monitoring port	Specify a unique port number on your local machine.
Host name	Specify the host name or IP address of the machine where the server is running.
Port	Specify the port number of the remote server.
Type	Specify whether the request type from the Web browser are sent by HTTP or TCP/IP. If the HTTP option is selected the requests from the Web browser are modified so that the HTTP header points to the remote machine and separated if multiple HTTP requests are received in the same connection. If the TCP/IP option is selected, all the requests are sent byte for byte.

3. In order to route the Web service through the monitor, the endpoint of the Web service client needs to be changed. The TCP/IP Monitor listens on port 9081. In the Web browser window used in step 1, invoke the setEndPoint method, and change the endpoint so that it directs to port 9081. For example, the default would be: `http://localhost:9081/web_module_context_root/servlet/rpcrouter` Invoke the getEndPoint method again to ensure that your change has been implemented.

To route traffic through the TCP/IP monitor and test the traffic for WS-I compliance:

1. Select a Web service method in the Methods pane. Invoke this method.

2. Change to the TCP/IP Monitor view by clicking the TCP/IP Monitor tab in the Servers view. This will display request and response pairs that are being routed through the TCP/IP Monitor. It will look similar to the following picture:



3. To ensure that your Web service SOAP traffic is WS-I compliant, you can generate a log file by clicking the icon. In the dialog box that opens, select a name for the log file and specify where you want it to be stored. This log file will be validated for WS-I compliance. You can open the log file in an XML editor to examine its contents.

Finish your tutorial by reviewing the materials in the Summary.

Summary

You have created a WS-I compliant Temperature Conversion Web service and an EJB skeleton from a WSDL file, and learned to implement some basic EJB methods.

Completed learning objectives

If you have completed all of the exercises, you should now be able to:

- Set the WS-I compliance level.
- Create a Web project called WebProject.
- Import the Temperature Conversion WSDL document.
- Validate the WSDL, and validate the WSDL file's WS-I compliance.
- Create the Web service and skeleton EJB, and test the methods included in the Web service using the Web Services Explorer.
- Implement the fahrenheitToCelsius and celsiusToFahrenheit methods (optional)
- Deploy the Web service to the WebSphere Application Server.
- Test the Web service traffic for WS-I compliance.

More information

For more information about Web services, WSDL, SOAP, and the WebSphere v6 run-time environment, consult the online help for WebSphere Studio by clicking **Help** > **Help Contents**. For more in-depth technical articles on Web services, consult www.ibm.com/developerworks/webservices