# Development of Web service using

# RATIONAL DEVELOPER FOR POWER SYSTEM – RDP

## Objectives

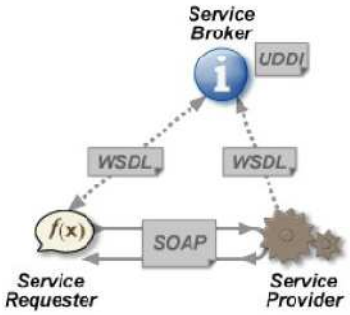
The hotspot in the current IT world is optimized Web Services. Now the enterprises have started their vision towards next generation Web Services. This white paper gives Information for the business managers and engineers in developing and deploying Web Services using Rational Developer for Power System (RDP) and Web Sphere Application Server (WAS), respectively. This paper explains the steps to create Web Service using RDP.

## What are Web Services?

Web Services are relatively a new technology that implements service-oriented architecture. The development of this technology, involves a major focus on making functional building blocks accessible over standard Internet protocols that are independent from platforms and programming languages. Web Services are self-contained, modular applications that can be described, published, located and invoked over networks. Web Services encapsulate business functions, ranging from a simple request-reply to full business process interactions and can be new or wrapped around existing applications.

## What are the Key Benefits of Web Services?

- Reusable Application Components
- Connecting Different Applications/Software.
- Web Services are self-contained
- Web Services are self-describing
- Web Services can be published, located and invoked across the Web.
- Web Services are modular
- Web Services are language independent and interoperable
- Web Services are inherently open standard
- Web Services are loosely coupled
- Web Services are dynamic
- Web Services provide programmatic access
- Web Services enables to integrate already existing and new applications



Web Service Architecture

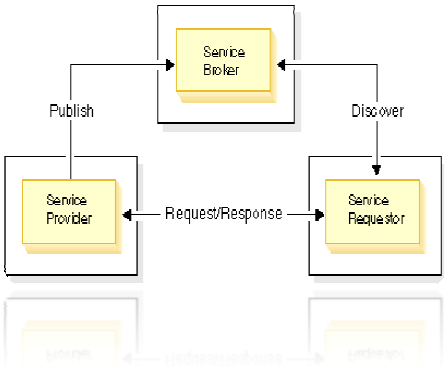
## Overview of RDP and IBM i Web Service and Java Tool

IBM I Web development tools give you the ability to create new e-business applications that use a Web-based front end to communicate with the business logic in an ILE or non-ILE language program residing on an IBM i server.

IBM I Web services are self contained, modular application that can be described, published, located, and invoked over the Internet, employing 'just-in-time' integration of services.

The Web Service wizard creates a Bottom up IBM i Program Web Service, with Web service operations that call one or more server programs or service program procedures on IBM i.

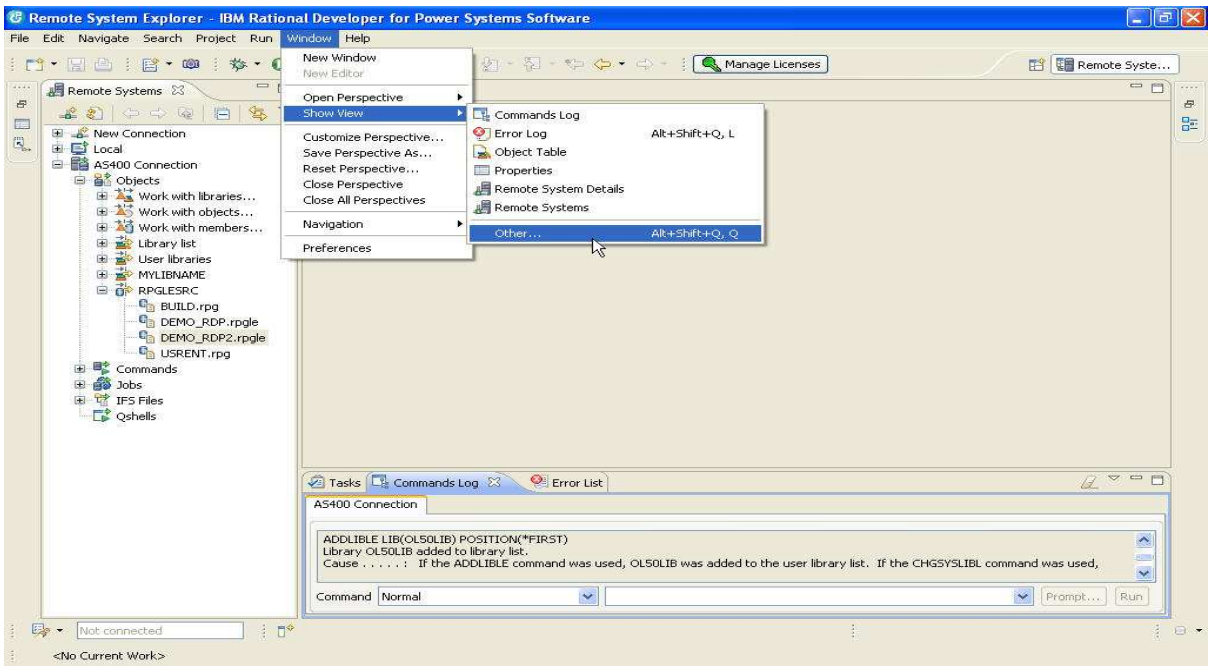
The Web Service wizard works in the context of a Web project and allows for creation, deployment, testing, generation of a proxy, and publication to a Universal Description, Discovery, and Integration (UDDI) registry of Web service.

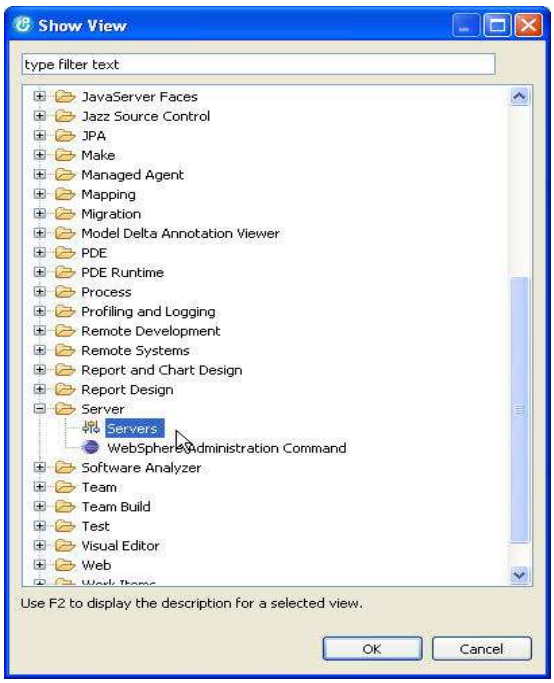


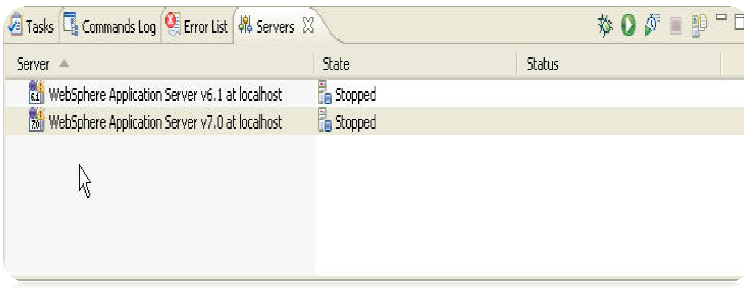
## Steps of Creating Web Service using RDP

Follow the Steps as shown in the screen shot and read the description for better understanding.

## Open Server Window

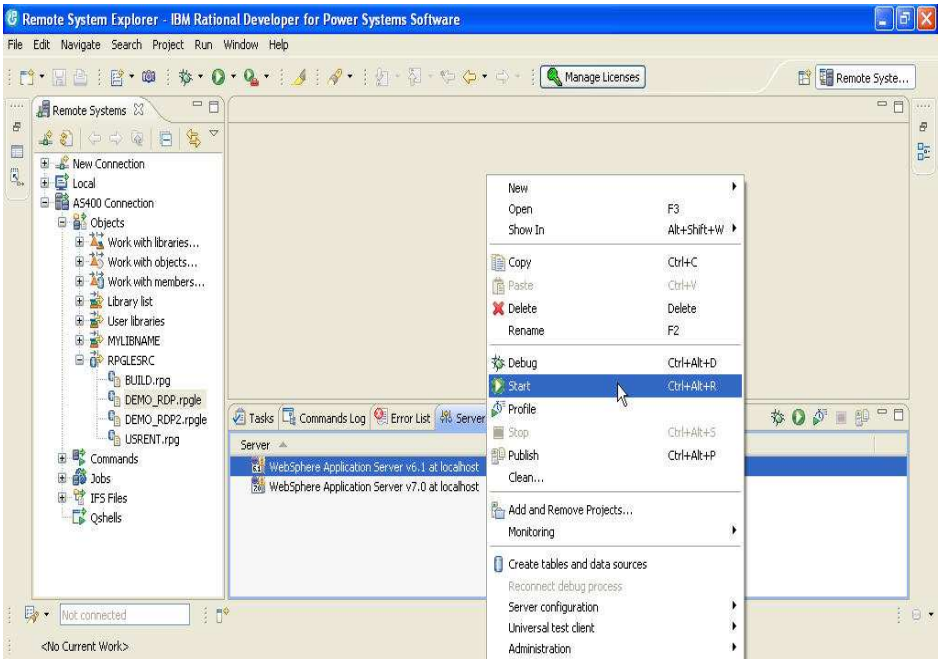



Click on the **Server**. It will open the WAS Server which are install on it. As shown in the Figure above

## Start Server

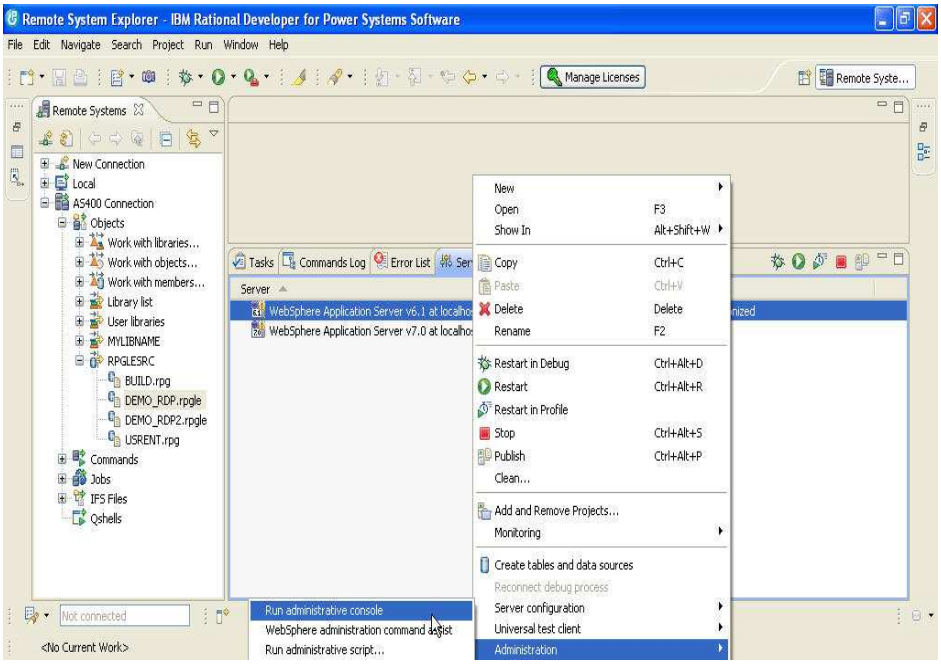



**Right Click** on the selected server and Click on **Start** as shown in figure.

*Note: When you start the server it might take several minutes to be start*

Start its status would be **Started**

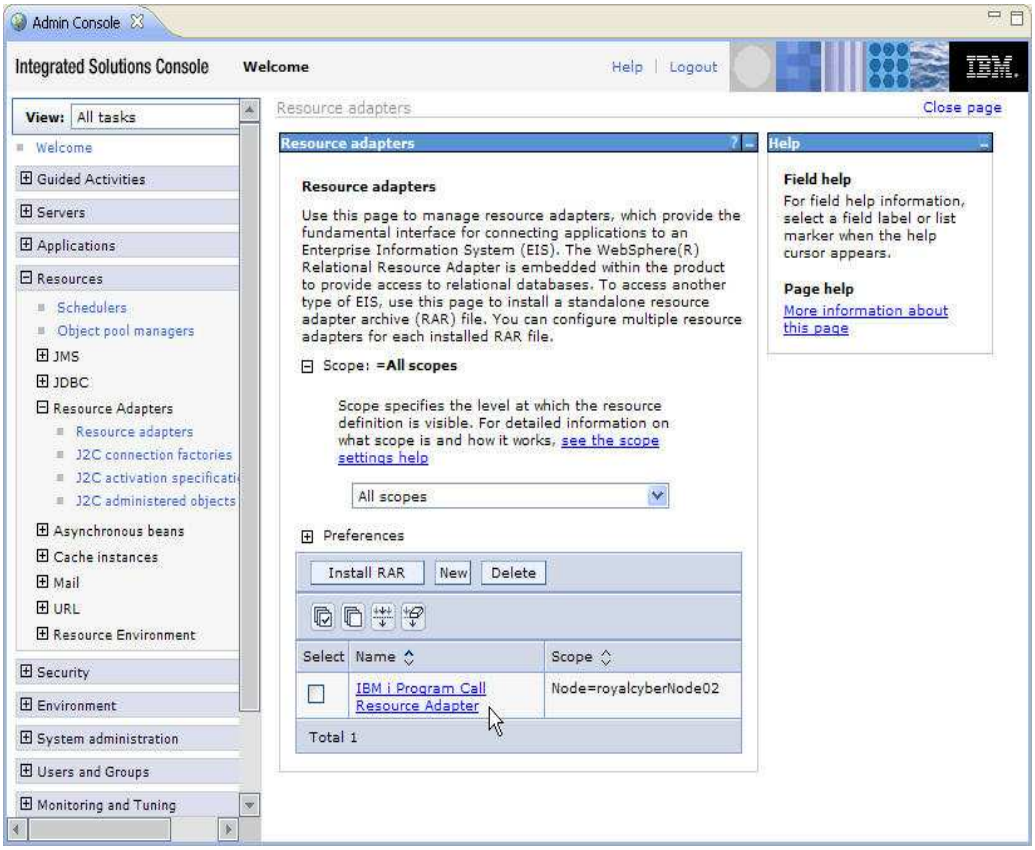
## Setting Up Server – Step 1




**Install a RAR file**

In the Servers view, **Right-click** the started server, and then select **Administration** > **Run administrative console**.

Enter **User ID** and **Password** if necessary.




**Install a RAR file**

In the left panel, expand **Resources** > **Resource Adapters** and click **Resource adapters**.

Select **Node=node** name in the Scope list.

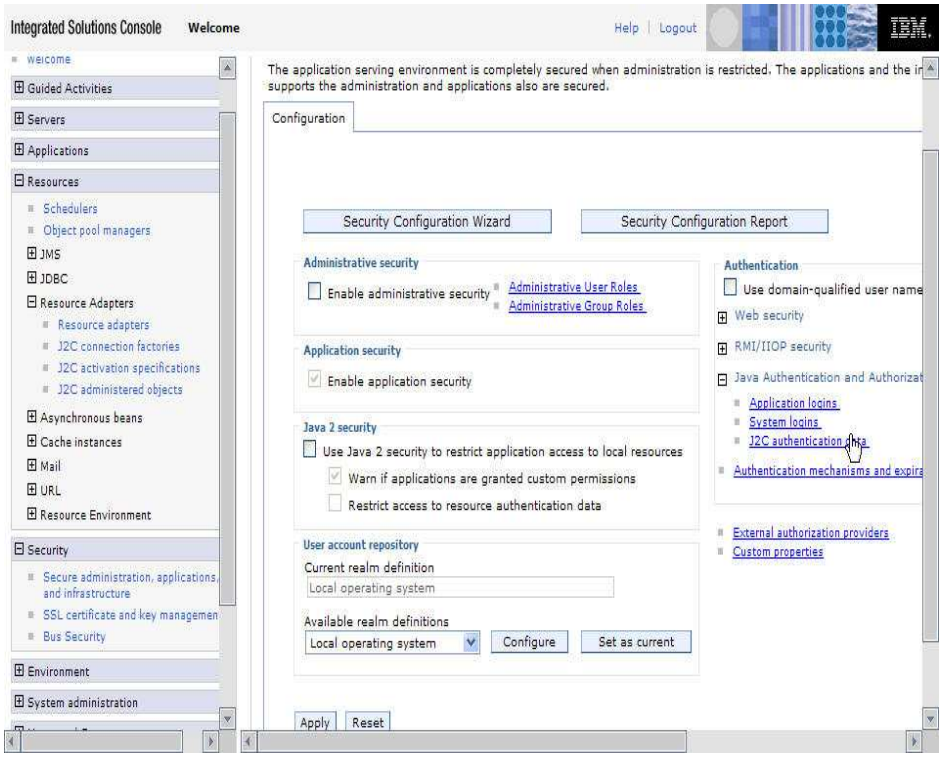
Click **Install RAR** to install a new RAR file. Click Browse in Local path to locate and select

*Program Files\IBM\SDPShared\plugins\com.ibm.etools.iseries.webtools.ae_version\lib\iseriespgmcall.rar*

Change the name to **IBM i Program Call Resource Adapter** in the Name field, and then click OK.

Click **OK** and then **Save** for directly save into master configuration.

## Setting Up Server – Step 2

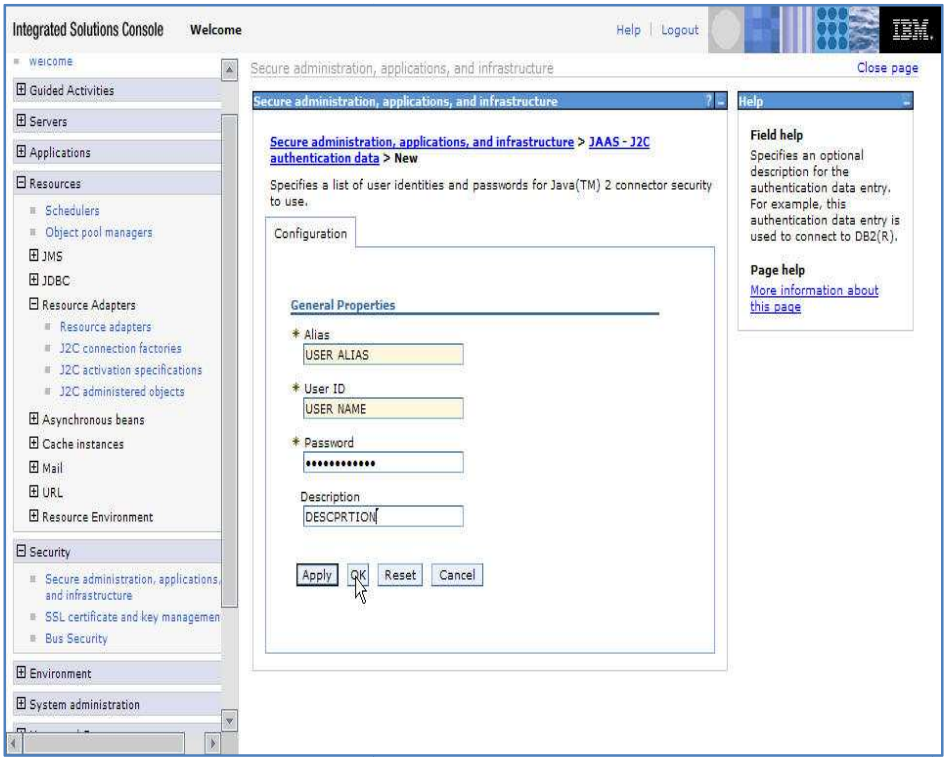



**Defining the connection factory**

**Create JAAS authentication entry**

In the left panel, expand **Security** and click **secure administration, applications, and infrastructure**.

In the **Authentication section**, expand **Java Authentication and Authorization Service** and click **J2C authentication data**.




**Create JAAS authentication entry**

Click **New** to create a new JAAS authentication entry.

Enter a name in the **Alias** field that represents the alias name of the JAAS login configuration for the role-based authorization for J2EE resources.

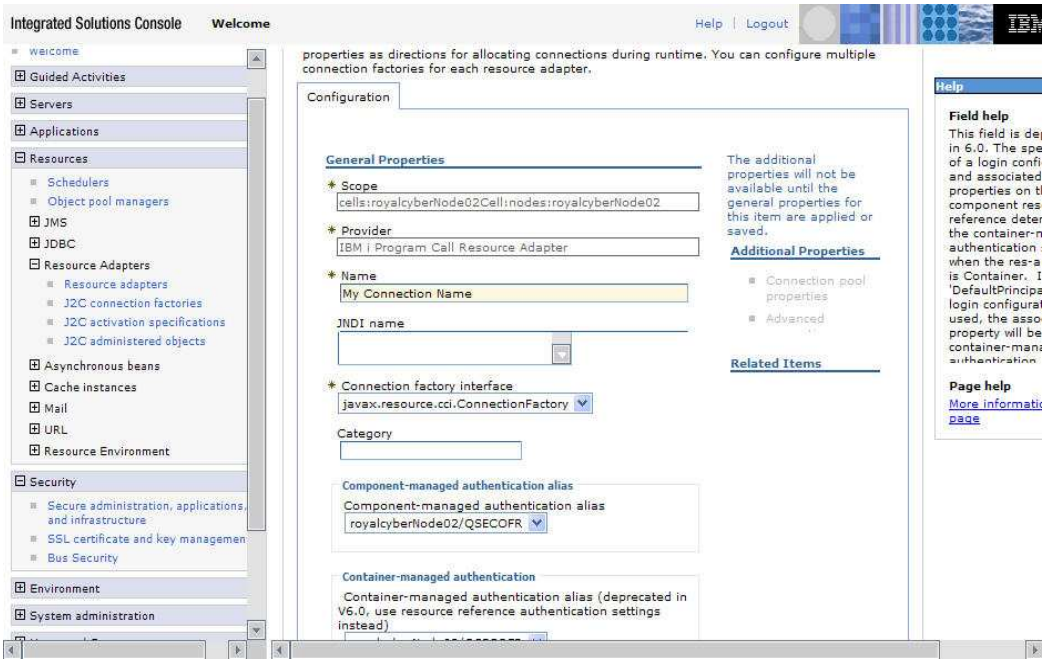
Enter an ID for the JAAS login configuration in the **User ID** field.

Enter a password for the JAAS login configuration in the **Password** field.

Enter a description for the JAAS login configuration in the **Description** field.

Click **OK**. Then Click **Save** to save directly to the master configuration
**Note:** The **USER ID** and **PASSWORD** that you specify in the JAAS authentication entry are used for signing on to the **IBM i system** for the program call.

## Setting Up Server – Step 3



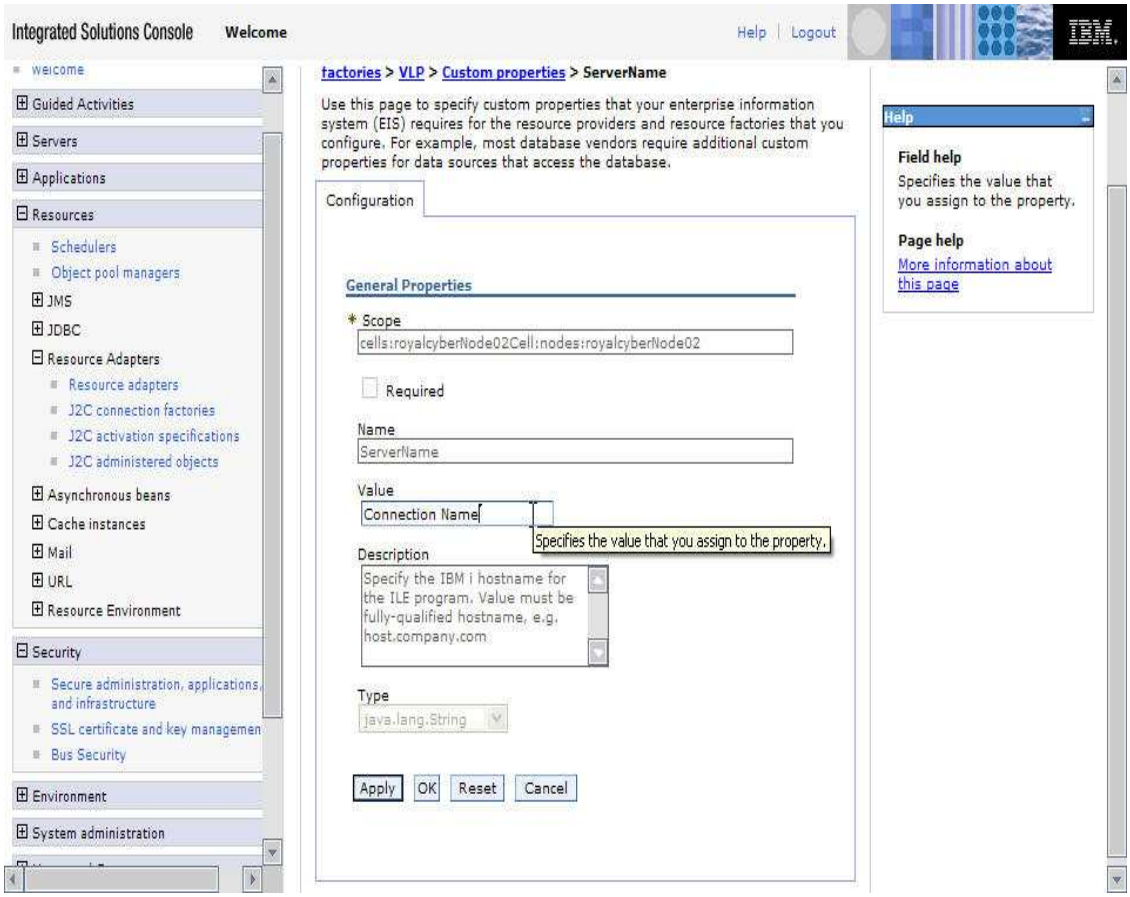

**Create a J2C connection factory**

- In the left panel, expand **Resources** > **Resource Adapters** and click **Resource adapters**.
- Click the resource adapter name: **IBM i Program Call Resource** Adapter you created.
- In **Additional Properties** section, click **J2C connection factories**.
- Click **New** to create a new **J2C connection factory**.
- Enter a name for the connection factory in the **Name** field.
- Define the JNDI name by entering the name in the **JNDI name** field. For example, enter a name in the form eis/name.
- Select the alias name, which you defined on the Security page in the JAAS Authentication Entries area, from the list for **Container-managed authentication** alias.

Select the same alias name as in the previous step from the list for **Component-managed authentication** alias.

Click **OK** to create the connection factory. Then Click **Save** to save directly to the master configuration.

## Setting Up Server – Step 4



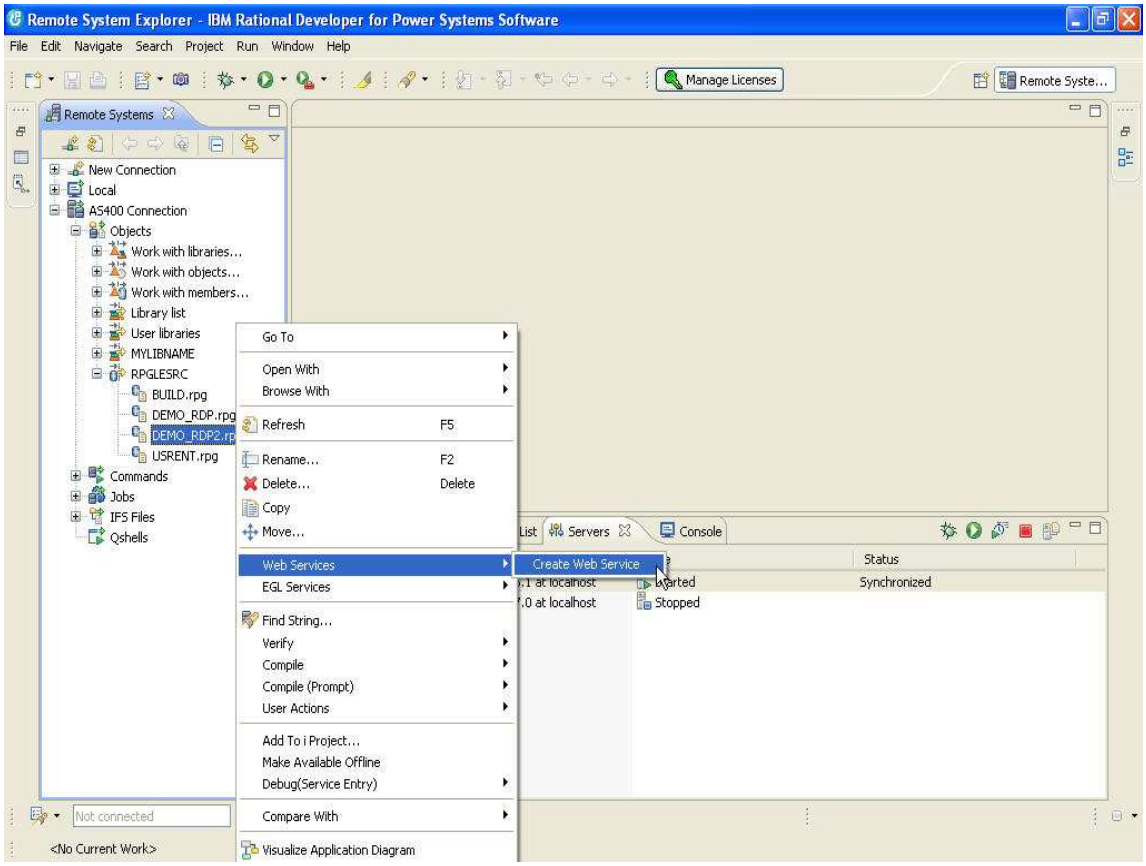

**Define resource properties associated with the connection factory**

- Click the connection factory name that you created.
- In **Additional Properties** section, click **Custom properties**. You should see **ServerName**, **UserName**, and **Password** in the **Name** column of the table. **ServerName** is a mandatory property. Click the **ServerName** in the **Name** column and enter its value in the **Value** field. **UserName** and **Password** are optional properties whose value is provided by the JAAS alias.

Click **OK**. Then Click **Save** to save directly to the master configuration.

***Note: after this close Admin panel and back REMOTE SYETEM EXPLORER***

## Creating IBM I Web Service – Step 1




You can create a Web service from ILE RPG or COBOL source or from PCML files using the Web Service wizard.

Start the application server before running the Web Service wizard.

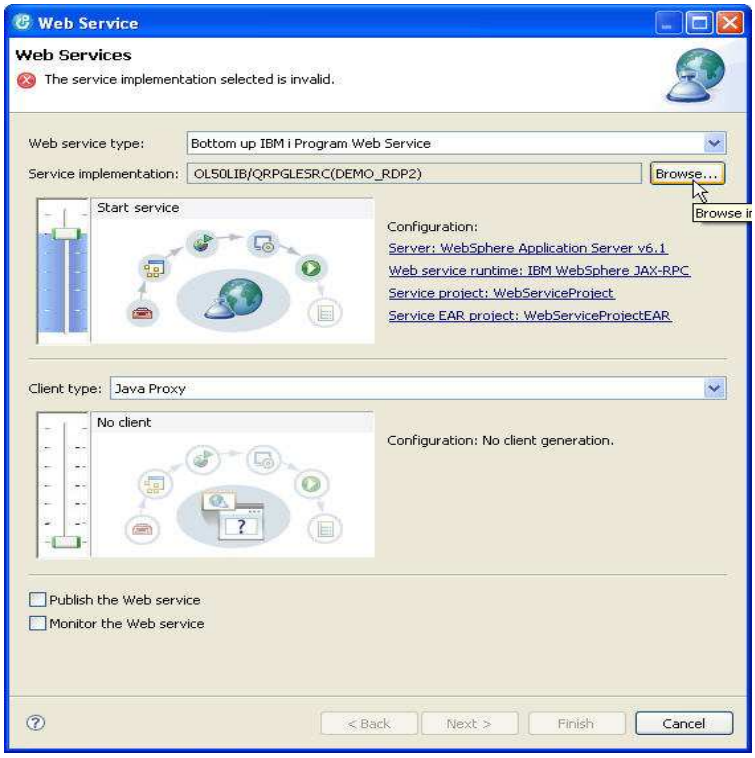
From the Servers view **(Window > Show View > Other > Server > Servers),** right-click on a server in the list and select Start.

**Right-click** the ILE RPG, COBOL, or PCML source object and select **Web Services** > **Create Web Service**.

*If the library containing your source object is not in the library list, expand Objects, right-click Library list, select Add Library List Entry, and enter the name of your library in the Additional library field.*

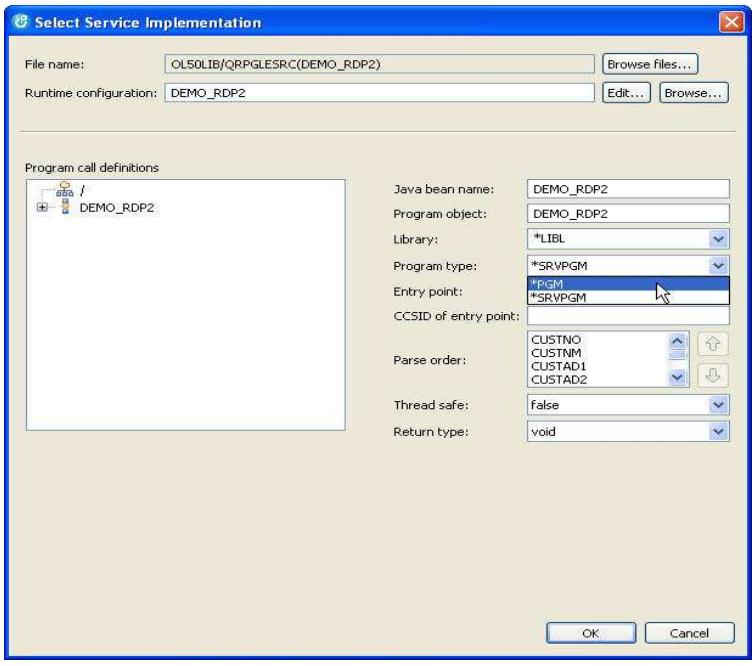
The Web Service wizard opens and is populated with data from the source object that you selected.

## Creating IBM I Web Service – Step 2



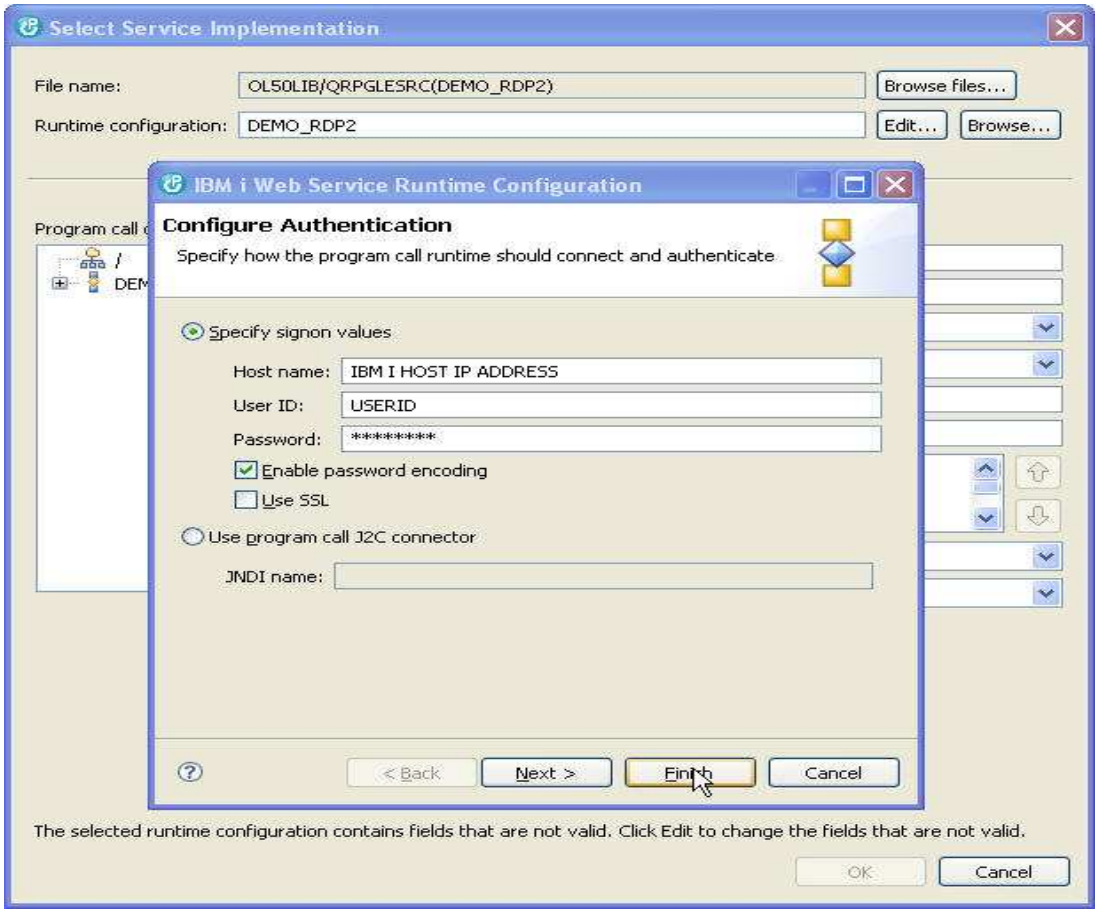

- In the **Web service** type field, **Bottom up IBM i Program Web Service** should be highlighted. If not, select it from the **drop-down list**.
- Click **Browse** to open the **Select Service Implementation** window.
- Ensure that the Program type is correct, especially if you are creating a **Web Service** from an ILE RPG or COBOL source file. Program type defaults to ***SRVPGM**, which may not be correct for your source file.

## Creating IBM I Web Service – Step 3




- Click **Browse** next to the **Runtime configuration** field to find authentication and runtime configuration values, or click **Edit** to set them. See Configuring your IBM i Web Service runtime environment.
- Ensure that the Program type is correct, especially if you are creating a **Web Service** from an ILE RPG or COBOL source file. Program type defaults to ***SRVPGM**, which may not be correct for your source file.

## Creating IBM I Web Service – Step 4




- Enter the **Host IP Address** or **Host Name**
- Enter **User id** of IBM I Server to Login
- Enter **Password**.

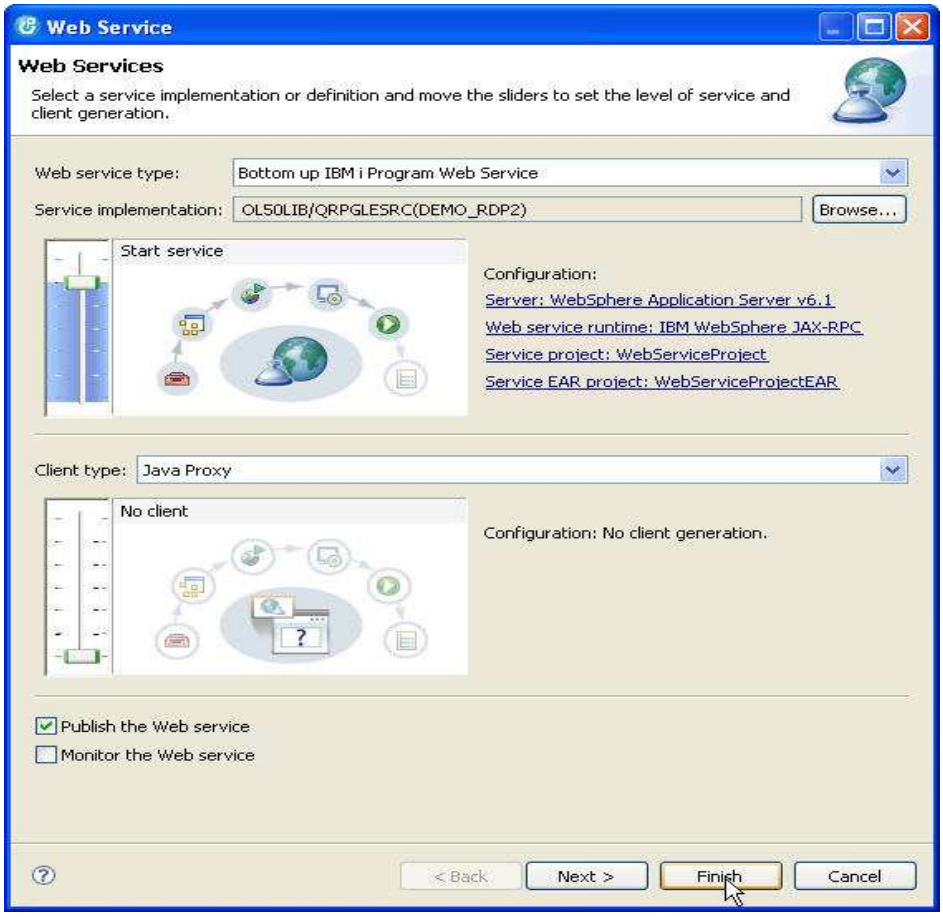
Click on **Finish**

Note: When you return to the Select Service Implementation window, click OK to set the values and to return to the first page of the Web Service wizard.

## Creating IBM I Web Service – Step 5

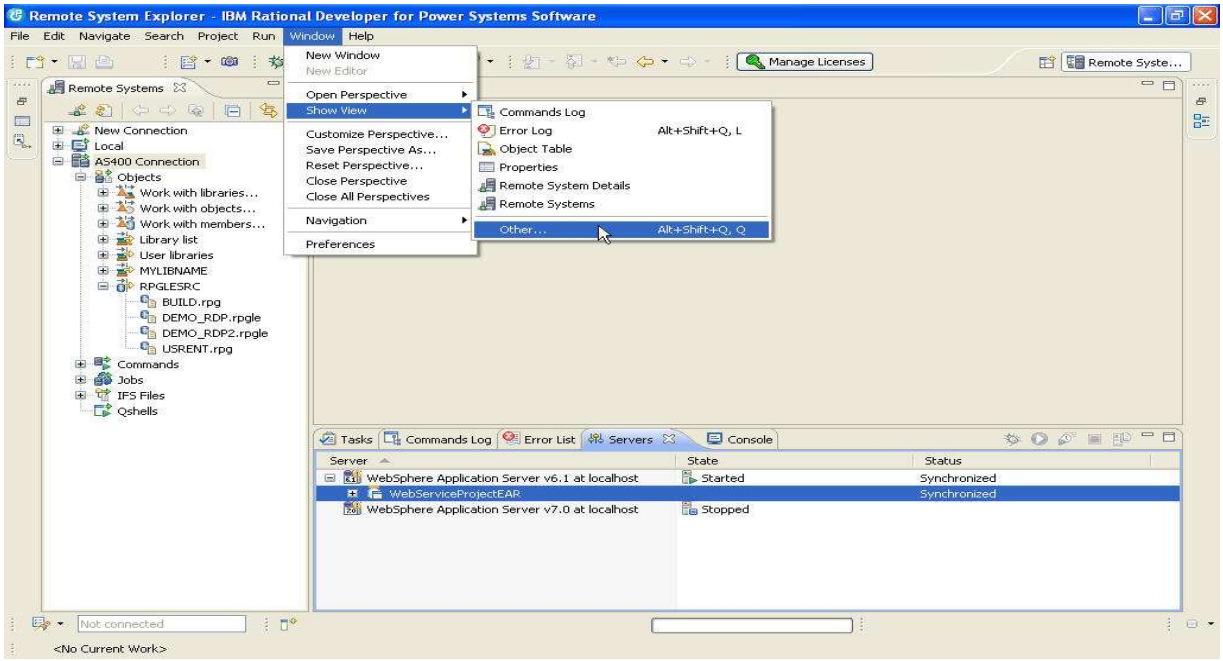
**Select the stages of the Web services development that you want to complete using the slider:**

- **Develop:** this develops the WSDL definition and implementation of the Web Service. It includes such tasks as creating modules, which will contain the generated code, WSDL files, deployment descriptors, and Java files when appropriate.
- **Assemble:** this ensures that the project that hosts the Web Service or client is associated to an EAR when required by the target application server.
- **Deploy:** this creates the deployment code for the service.
- **Install:** this installs and configures the Web module and EARs on the target server.
- **Start:** this starts the server once the service has been installed on it. The server-config-wsdd file is generated.
- **Test:** this provides various options for testing the service, such as using the Web Service Explorer or sample JSP files
- The **Server** option displays the default server. To deploy your service to a different server, click **Server** and specify the server that you want to use.
- The **Web service runtime** displays the default runtime. To deploy your service to a different runtime, click **Web service runtime** and specify the runtime that you want to use.
- The **Service project** option displays the project containing the project selected in your workspace. To specify a different project and EAR file, click the **Service project link**. Ensure that the project selected as the **Client Web Project** is different than the **Service Web Project**, or the service will be overwritten by the client's generated artifacts.
- If you want to create a client, select the type of proxy to be generated from the **Client type** list, and repeat the above slider steps for the client.
- Select the **Publish the Web service check box** if you want to launch the Web Services Explorer to publish your Web service to a UDDI registry.
- Select the **Monitor the Web service check box** to send the Web service traffic through the TCP/IP monitor, which allows you to watch the SOAP traffic generated by the Web service, and to test this traffic for WS-I compliance of the specified Service project. As an alternative, you can manually set up a TCP/IP monitor.
- Either click **Finish** to create the Web service or click **Next** to configure advanced options.

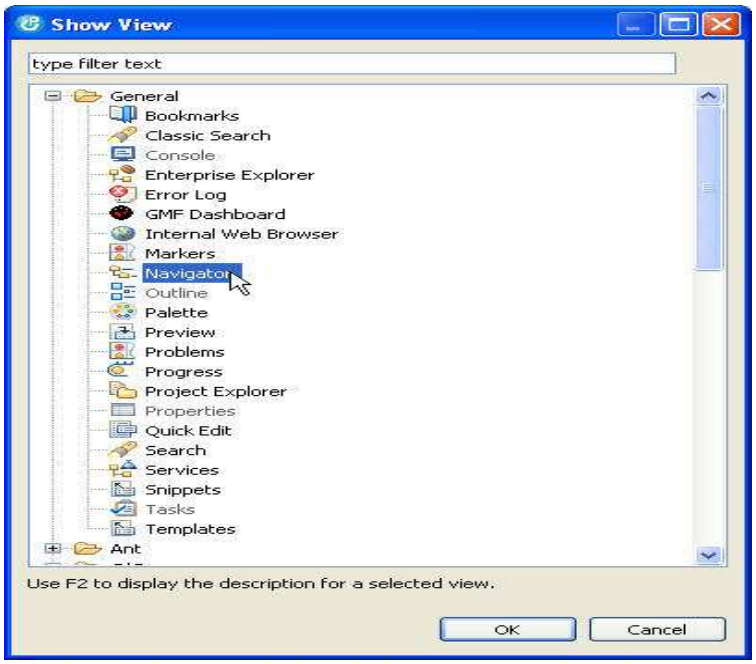



**Click the Finish button for create the Web service. The Window will close once it creates successfully. IF found any error follow the steps again**

# Web Service Created – Running/Testing the Web Service Showing in the Project

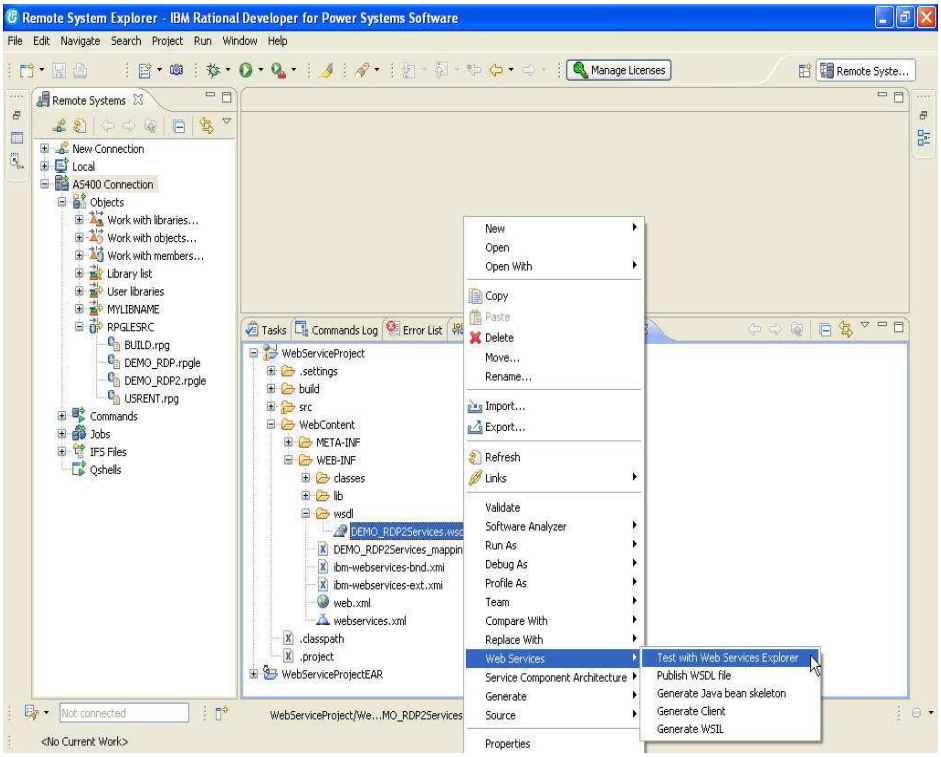


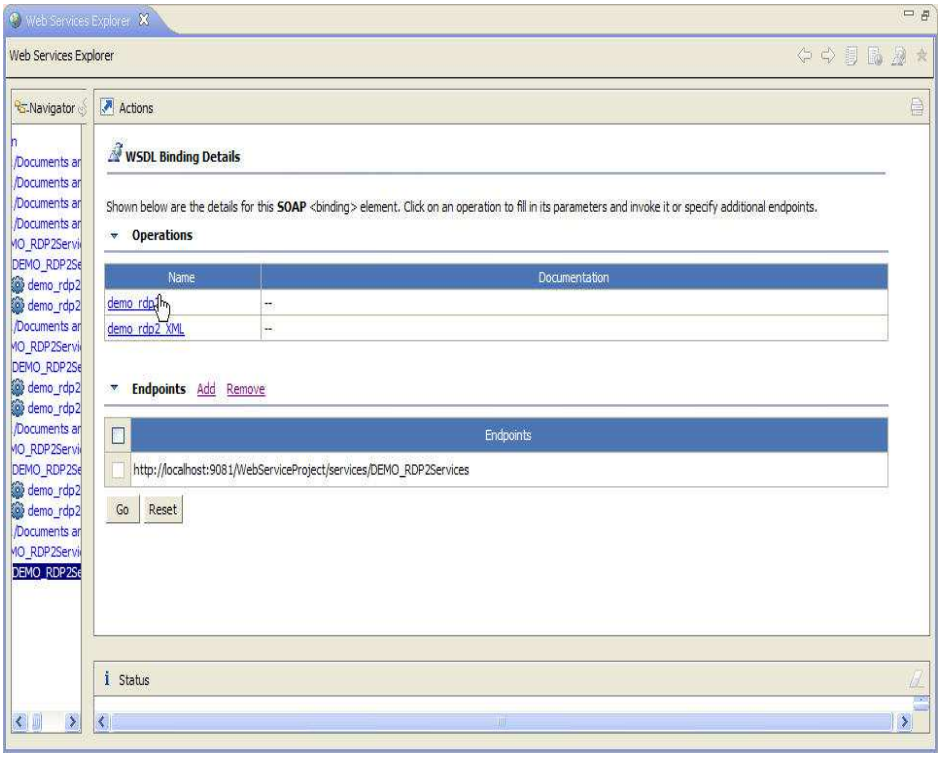

## Testing/Running the Web Service




Click on the Navigator as shown in the Image

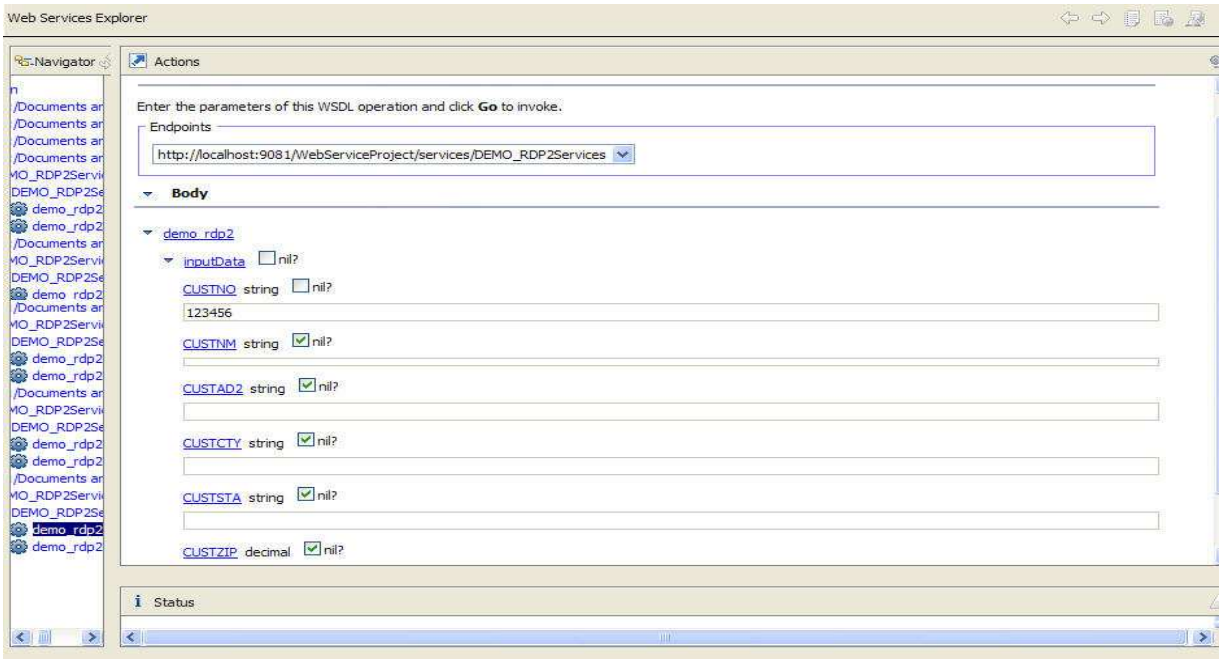
For running the web service.



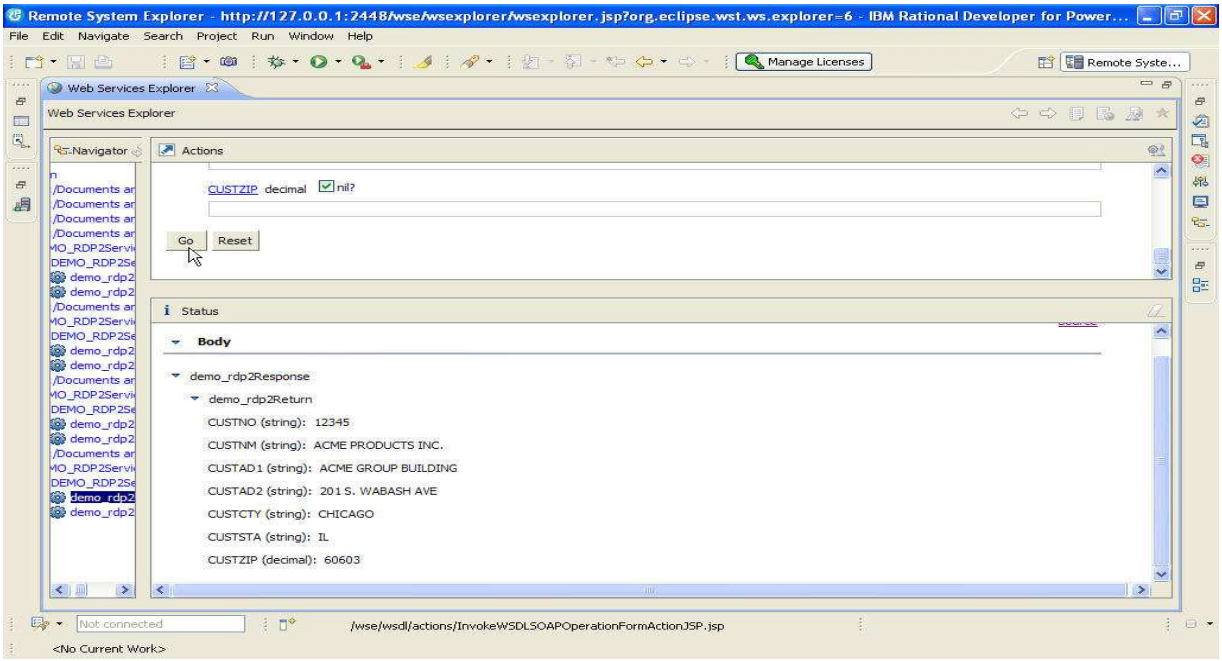




- There are showing two options. When you create the Web service through wizard.
- The wizard creates two files. One Java beans and other the XML format. Try to test both you will get the idea of each option.
- Both options generate the same output. But when you run from option one its generate
- the form base output and other in the XML base.

## Web Service Input




## Web Service Input/Output




Click on the **Go** button to invoke the web service to fetch the data from the IBM I server and display the output.

In the below image the output is generated from the IBM I Server