Accessing Data Services Via AquaLogic Service Bus 2.5

AquaLogic Data Services Platform can be accessed from AquaLogic Service Bus. Thus an AquaLogic Service Bus client can make full use of data services. This approach allows a more efficient and flexible approach to accessing data services than exposing them as web services via WebLogic Workshop and Java Web Services (JWS).

Introduction

The following table identifies the release compatibilities between AquaLogic Data Services Platform and Service Bus:

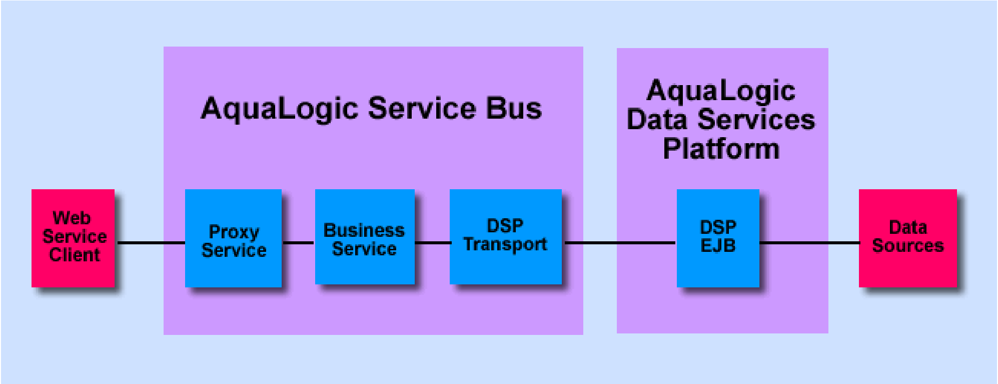
AquaLogic Data Services Platform Version	AquaLogic Service Bus Version
Version 2.5	Version 2.5 Version 2.6

To make an AquaLogic Data Service Platform data service available to an AquaLogic Service Bus client, you need to do the following:

- Deploy the AquaLogic Data Service Platform Transport to AquaLogic Service Bus as an application.
- Generate the WSDL file for the data service of interest and import it into the AquaLogic Service Bus.
- Generate a business service based on the WSDL
- Generate a proxy service based on the business service.

Your client application is then able to access the data service as an AquaLogic Service Bus client.

Figure 1 AquaLogic Interoperability Architecture



Accessing AquaLogic Data Services Platform from AquaLogic Service Bus

Note: The following assumes that you are running AquaLogic Service Bus 2.5 under WebLogic Server 9.2 and AquaLogic Data Service Platform under WebLogic Server 8.1.

The following sections describe the steps necessary to access a data service from AquaLogic Service Bus.

Step 1: Start Your Server

1. Start the WebLogic Server 9.2 for the AquaLogic Service Bus application needing access to your WebLogic Server 8.1 data service.
2. Start the WebLogic Server 8.1 for the data service.

For example, suppose that the MortgageBroker application sample in AquaLogic Service Bus needs access to a data service. For the purpose of this discussion, the sample RTLApp application provided with AquaLogic Data Service Platform is used.

First, start the server for the AquaLogic Service Bus Mortgage Broker examples. On Windows you can do this by selecting:

Start → All Programs → BEA Products → Examples → AquaLogic Service Bus → Start Examples Server

Then, start the server for the RTL demo. On Windows you can do this by selecting:

Start → All Programs → BEA WebLogic Platform 8.1 → BEA AquaLogic Data ServicesPlatform2.5 → Examples → RTL Demo → Launch RTL Demo Server

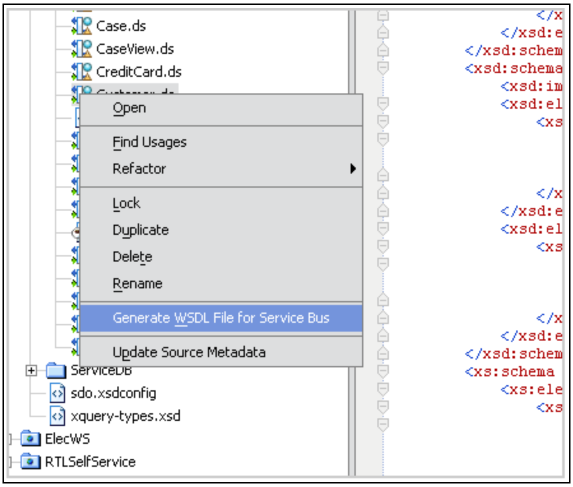
Step 2: Generate a WSDL for the Data Service

You can generate a WSDL for your data service in two different ways. Each is described below.

A. Generating a WSDL File Using WebLogic Workshop 8.1

1. Launch WebLogic Workshop 8.1.
2. Navigate in the Application (generally the left-most pane) to your data service (`.ds` file). Select it.
3. Right-click to select the Generate WSDL File for Service Bus option. A WSDL file for the data service will be generated in the same directory where the data service is located.

Figure 2 Generate WSDL for Service Bus Dialog



For example, if you had navigated to the `Customer.ds` file in the RTLApp, `Customer.wsdl` will be generated.

B. Export the WSDL Using the AquaLogic Data Service Console

1. Launch the AquaLogic Data Services Console.

   On Windows you can do this by selecting:

Start → All Programs → BEA WebLogic Platform 8.1 → BEA AquaLogic Data Services Platform 2.5 → Examples → RTL Demo → AquaLogic Data Services Console

   You can also do this by typing `http://localhost:7001/ldconsole` in your web browser.

2. In the project navigator on the left, select ldplatform, then your application underneath it.
3. Next navigate to Data Services, then to the data service for your application.

   For instance, if your application were the RTLApp example application, then you would navigate from ldplatform to RTLApp to Data Services to RTL Services.

4. Pick the particular service that you want to export.

   If you were exporting a service from the RTLApp example, you might pick Customer.

5. Click Export WSDL in the far right column of the data service you want to export. In this case, you can save your WSDL file to any directory.

Step 3: Deploy the Data Services Transport

1. Launch the AquaLogic Service Bus console, then select the WebLogic Server Console.

   The WebLogic Server Console will appear in a new window. Click on Deployments under Your Deployed Resources.

2. When the Summary of Deployments panel appears, click Lock & Edit under Change Center on the left.
3. The Install button then becomes active. Click it.
4. In the Install Assistant, navigate to the deployment EAR file, `dsp_transport.ear`, in WebLogic Server 8.1.

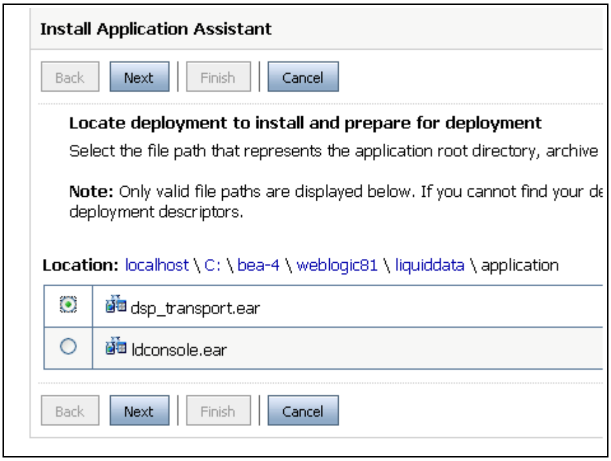
   In Windows, this is located at:

   ```
   <bea_home>\weblogic81\liquiddata\application
   ```

5. Then select `dsp_transport.ear` and click Next.

   The option dsp_transport will appear under Deployments.

Figure 3 Install Application Assistant Dialog



Install Application Assistant

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive

Note: Only valid file paths are displayed below. If you cannot find your de
deployment descriptors.

Location: localhost \ C: \ bea-4 \ weblogic81 \ liquiddata \ application

●	dsp_transport.ear
○	ldconsole.ear

6. Click Activate, then check the box in front of `dsp_transport.ear` and select Servicing All Requests from the Start drop-down menu. This will complete activation.

Step 4: Import the WSDL for the Data Service

1. Return to the service bus console and select Project Explorer.
2. Navigate to your project folder and click either Create or Edit.

   Note: There will be either a Create or an Edit button in the Change Center, depending on whether you are creating a new session or editing an existing one.

3. Select the project folder or an existing subfolder, or create a new one. For the latter, type a name in Enter New Folder Name under Folders and click Add. The new folder will be added to the tree structure.
4. Next import the WSDL file that you generated in Step 2. Do this by selecting WSDL under Interface in the Create Resource drop-down menu.
5. On the panel that appears next, give the resource a name and a description (optional), then click Browse to locate the WSDL file that you generated in Step 2. Select it and click Save. It will then appear as a resource in the Resources table.

For more information, see Adding a WSDL in “Using the AquaLogic Service Bus Console.”

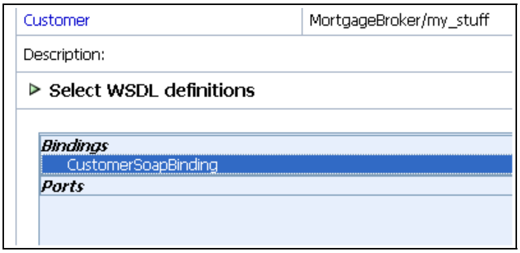
Step 5: Create the Business Service

1. In Create Resources under Resources, select Business Service under Service. In the panel that appears, enter a name for the service in Service Name. For instance, if you had created `Customer.wsdl` for the sample RTLApp, then you could name your business service CustomerBS. Enter an optional description of the service.

2. Below, in the same panel, select WSDL Web Service and click Browse to locate the WSDL file that you imported in Step 4. Click on it. A Select WSDL Definitions panel will appear. Select a binding under Bindings.

   If you were running the sample application and had imported `Customer.wsdl` as described above, you would select CustomerSoapBinding (Setting SOAP Binding for WSDL).

Figure 4 Setting SOAP Binding for WSDL



Customer MortgageBroker/my_stuff

Description:

▷ Select WSDL definitions

Bindings

CustomerSoapBinding

Ports

3. Click Submit. Then click Next in the main panel.

4. Under Protocol, select `dsp` from the drop-down menu. Then enter the Endpoint URI. For Endpoint URI, enter:

   ```
   t3://<host:port>/<application_name>
   ```

   If you were running the RTLApp as described above in the default setup, you would enter:

   ```
   t3://localhost:7001/RTLApp
   ```

5. Click Add to add the Endpoint URI, then click Next.

6. Click Next again to accept the defaults.

6. On the Create a Business Service - Summary screen that appears, click Save.

   The business service should then appear under the Resources table. If you were creating `CustomerBS`, that name appears appear in the table.

For more information, see Adding a Business Service in "Using the AquaLogic Service Bus Console."

Step 6: Create the Proxy Service

The procedure for creating the proxy service is similar to that for creating the business service.

1. Under Create Resources, select Proxy Service under Service in the drop-down menu.

2. In the next panel that appears, give your proxy service a name.

   If you were working with the example, you might want to name it CustomerPS.

2. Provide a description (optional) for the service in Description.

2. Select Business Service under Create from Existing Service and click Browse.

2. In the screen that appears, select the name of the business service that you created in the previous step. If you had created CustomerBS, as in the example, select CustomerBS. Click Submit, then click Next.

3. In the next panel, select http in the drop-down menu for Protocol and type in a name for the Endpoint URI. For Endpoint URI you can give it any name you want. Click Next.

4. In the next two screens click Next to accept defaults, or fill in other values or select other choices.

5. You will then see a summary screen allowing you to edit your entries. When you are satisfied with your entries, click Save. The proxy service should then appear in the table of resources. If you were working with the example, you should see CustomerPS in the table.

6. Click Activate, then click Submit.

For more information, see Adding a Proxy Service in "Using the AquaLogic Service Bus Console."

Note: The AquaLogic Data Service Platform transport by default uses the utf-8 character set. When using a multibyte character set (such as Japanese), you will want to switch to a more appropriate character set in order to avoid unexpected characters appearing in editing fields.

Step 7: Test Your Setup

1. Reselect the folder where your resources are located, then, under Resources, locate the proxy service you created.

2. Select Launch Test Console under Actions. The test console should appear. Pick a method under Available Options in the drop-down menu and click Execute. For the RTLApp example, you might select the getCustomer() method. In Response Document, you should see a list of customers.

For more information, see Test Console in "Using the AquaLogic Service Bus Console."

Additional Information

For more information, see the following:

- BEA WebLogic Workshop 8.1 documentation

  `http://edocs.bea.com/workshop/docs81/index.html`

- Using the AquaLogic Service Bus Console

  `http://edocs.bea.com/alsb/docs25/consolehelp/index.html`

- Using the AquaLogic Data Services Platform Console

  `http://edocs.bea.com/aldsp/docs25/admin/ldconsole.html`