



BEA AquaLogic Service Bus™ and Cyclone Interchange®

BEA EDI Utilities Sample

AquaLogic Service Bus Version 2.1
BEA EDI Utilities Version 1.0
Document Date: April 2006

Copyright

Copyright © 2006 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software is protected by copyright, and may be protected by patent laws. No copying or other use of this software is permitted unless you have entered into a license agreement with BEA authorizing such use. This document is protected by copyright and may not be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA SYSTEMS DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks and Service Marks

BEA, BEA JRockit, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Communications Platform, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic Log Central, BEA WebLogic Network Gatekeeper, BEA WebLogic Personalization Server, BEA WebLogic Personal Messaging API, BEA WebLogic Platform, BEA WebLogic Portlets for Groupware Integration, BEA WebLogic Server Process Edition, BEA WebLogic SIP Server, BEA WebLogic WorkGroup Edition, Dev2Dev, Liquid Computing, and Think Liquid are trademarks of BEA Systems, Inc. BEA Mission Critical Support, BEA Mission Critical Support Continuum, and BEA SOA Self Assessment are service marks of BEA Systems, Inc.

All other names and marks are property of their respective owners.

Contents

BEA EDI Utilities Sample

Sample Description	1
Using the Sample	1
JMS Queues in WLS	2
Configuration Using the sbconfig.jar File	2
Outbound Scenario	3
Inbound Scenario	3
Cyclone Interchange Configuration	4
Integration Pickup Exchange	4
Integration Delivery Exchange	4
Setting the Environment	4
Using Web Services Transport	5
Outbound Web Services Transport	5
Inbound Web Services Transport	6
Using JMS Transport	7
Outbound JMS Transport	7
Inbound JMS Transport	9

BEA EDI Utilities Sample

Sample Description

This sample provides the capability to send and receive XML or EDI documents using either web service or jms transports. There are two primary scenarios when exchanging XML or EDI documents, these are.

- Outbound Scenario – The JMS or web service client sends the XML document to AquaLogic Service Bus™, which transforms the XML document in to an EDI document and then sends it to Cyclone® Interchange.
- Inbound Scenario – Cyclone Interchange sends the EDI document to AquaLogic Service Bus, which transforms the EDI document to XML message and then sends it to the back-end system or client.

Using the Sample

The sample is provided with an ant build file (`build.xml`) that contains targets for building the clients. It is also used for building and deploying (including redeploying and undeploying) the web service used as a backend system/client in an inbound scenario. After installation, the sample is located in `WL_HOME\samples\edi\examples\B2BEDISample`, where `WL_HOME` is the WebLogic Server® 9.1 installation directory. The sample directory consists of the following files and directories.

- `sbconfig.jar` – Contains the sample AquaLogic Service Bus configuration for inbound and outbound scenarios using JMS and web service transports.

- `bin` – Directory that contains the scripts for running the JMS and web service clients on windows and UNIX environments.
- `src` – Directory where the source files are located
- `wsdl_files` – Directory where the WSDL files are located
- `output` – Directory where the web service artifacts jar and class files are located
- `build.xml` – The ant build file that provides targets for building clients and the web service

In order to use the sample, create an AquaLogic Service Bus domain and extend the domain to include the EDI functionality. For more information, see section “Getting Started with AquaLogic Service Bus” in Chapter 4, “Designing the EDI Solution” of the *EDI Solution Guide*.

The following sections contain sample specific information that depicts the inbound and outbound scenarios for JMS and web service transports.

JMS Queues in WLS

In order for this sample to work, four JMS queues with the following JNDI names need to be created.

- `b2b.ci.jms.EDIQueue`
- `b2b.ci.jms.EDIQueue_Receive`
- `b2b.sb.jms.EDIQueue`
- `b2b.sb.jms.EDIQueue_Receive`

For more information on configuring JMS Queues, refer WebLogic Server documentation available at.

<http://e-docs.bea.com/wls/docs90/>

Configuration Using the sbconfig.jar File

The `sbconfig.jar` file contains the sample AquaLogic Service Bus configuration for inbound and outbound scenarios using both web services and JMS transports. For each scenario, there is a combination of proxy and business services that are configured. These services are described in this section, categorized by the scenario.

Outbound Scenario

For web services, the proxy and business service descriptions are as follows.

- **CycloneWebService** – This business service is configured to `http://localhost:5080/services/MessageService`, where the Cyclone Interchange web service provider is running. The Cyclone Interchange Integration Pick-up Exchange needs to be configured with the location specified above.
- **CIProxy** – This is a proxy service for the `CycloneWebService` business service, which listens at `http://localhost:7001/CIProxy`. Once a message is received by proxy, it is processed and sent to the `CycloneWebService` business service.

For JMS, the proxy and business service descriptions are as follows.

- **CycloneJMSService** – This business service is configured to the `b2b.ci.jms.EDIQueue` queue, where the Cyclone Interchange's Integration Pick-up Exchange should have been configured.
- **CIJMSProxy** – This is proxy service for the `CycloneJMSService` business service, and is configured to the `b2b.sb.jms.EDIQueue` queue. Once the client sends a message to this queue, the message is processed and sent to the `CycloneJMSService` business service.

Inbound Scenario

For web services, the proxy and business service descriptions are as follows.

- **ClientWebService** – This business service is configured to `http://localhost:7001/B2BSample/MessageService`, so that the web service can process incoming messages and save it to the file system.
- **ClientProxy** – This proxy service is for the business service `ClientWebService` and listens at `http://localhost:7001/ClientProxy` for the incoming messages. When a message is received, it is processed and sent to `ClientWebService`. Cyclone Interchange's Integration Delivery Exchange should be configured to this URL.

For JMS, the proxy and business service descriptions are as follows.

- **ClientJMSService** – This business service is configured to the `b2b.sb.jms.EDIQueue_Receive` queue, so that the back-end system or client can receive the messages.
- **ClientJMSProxy** – This proxy service is for the business service `ClientJMSService` and is configured to listen on the `b2b.ci.jms.EDIQueue_Receive` queue. Once a message is

received, it is processed and sent to the business service `ClientJMSService`. The Cyclone Interchange Delivery Exchange should be configured to this queue.

To import the configuration file (`sbconfig.jar`) and to configure the proxy and business services, refer AquaLogic Service Bus documentation available at.

<http://e-docs.bea.com/alsb/docs21/>

Cyclone Interchange Configuration

Before using Cyclone Interchange, you need to configure two integration pick-up exchanges (one each for JMS & web services), and two integration delivery exchanges (one each for JMS & web services.)

Integration Pickup Exchange

Configure the web services integration pickup exchange to

`http://localhost:5080/services/MessageService` and the JMS integration pickup exchange to the `b2b.ci.jms.EDIQueue JMS Queue`.

Integration Delivery Exchange

Configure the web services integration delivery exchange to

`http://localhost:7001/ClientProxy` and the JMS integration delivery exchange to the `b2b.ci.jms.EDIQueue_Receive JMS queue`.

Setting the Environment

Open a command window and set your WebLogic Server environment by executing the following platform specific scripts.

Windows: `setDomainEnv.cmd`

Unix: `setDomainEnv.sh`

These scripts are located in the `bin` directory under your domain directory. The default location of a WebLogic Server domain is.

`BEA_HOME/user_projects/domains/domainName`

where `BEA_HOME` is the top-level installation directory of BEA products, and `domainName` is the name of your domain.

Using Web Services Transport

This section discusses the outbound and inbound scenarios for web services transport.

Outbound Web Services Transport

In an outbound scenario, a web service client (based on a WebLogic stack) is used to send the XML document to AquaLogic Service Bus. The client uses the interface provided by `MessageService_Cyclone.WSDL` (located in `wSDL_files` directory) to send the XML document to AquaLogic Service Bus. Following are steps for building and running the web services client.

1. Open a command window and navigate to the samples directory shown below.

```
WL_HOME\samples\edi\examples\B2BEDISample
```

2. Build the client based web services on the WebLogic stack using the `build-ws-client` ant target, as shown below.

```
ant build-ws-client
```

By default, the client sends the XML document to the following location.

```
http://localhost:7001/CIProxy
```

Where, the AquaLogic Service Bus proxy is listening for any incoming message. If the AquaLogic Service Bus proxy is configured to a different URL, then update the `location` attribute in `wSDLsoap:address` element of the `MessageService_Cyclone.WSDL` file.

3. Use the `client-gen` command to generate files with the updated location, as shown below:

```
ant client-gen
```

Once the files are generated, you have to build them.

4. Build the newly generated or updated files created in the previous step.
5. After successfully completing the build, change to the `bin` directory in the command window. This directory contains all the platform specific scripts required to send XML documents to AquaLogic Service Bus.

Windows: `run-ws-client.bat`

Unix: `run-ws-client.sh`

The following table lists the parameters that are required to execute these scripts.

Parameter	Description
<XML File Path>	Complete path and file name of the XML file to be sent to AquaLogic Service Bus.
<isContent>	Whether the XML file is sent as an URL or as part of a SOAP message. If set to <code>true</code> , the XML file content will be sent as part of the SOAP message.
<Document Name>	Name of the EDI document to which this XML file belongs. For example: 850, 855, ORDERS and so on.
<Document Version>	Specify the version to which this EDI document belongs. For example: 4020, 4030, D98B and so on.
<Document Class>	The standard to which this EDI document belongs. For example: X12 or EDIFACT.

- Use the `clean-client` target to clean the client files, as shown below.

```
ant clean-client
```

Inbound Web Services Transport

In the inbound scenario, AquaLogic Service Bus sends an XML document to the web service hosted on a WebLogic Server. The web service implements the `MessageService.WSDL`, which can be found in the `wsdl_files` directory. By default, this is hosted at the `http://localhost:7001/B2BSample/MessageService` location. This URL needs to be configured in AquaLogic Service Bus, so that AquaLogic Service Bus always sends to this web service.

The functionality of a web service is that whatever the XML document it receives, it saves it in the file system depending on the location specified in the `MessageService.properties` file. This properties file can be found in the following directory:

```
WL_HOME\samples\edi\examples\B2BEDISample\src\com\bea\wli\b2b\sample
```

where `WL_HOME` is the WebLogic Server[®] 9.1 installation directory.

The steps for building and deploying/undeploying the web service are described below.

- Open a command window and navigate to the samples directory shown below.

```
WL_HOME\samples\edi\examples\B2BEDISample
```

2. Use the `deploy` target to build and deploy the web service, as shown below.

```
ant deploy
```

This target deploys the `B2BSampleEar` web service on the `AdminServer`, using the following values.

Username: `weblogic`

Password: `weblogic`

3. To check whether the web service is deployed correctly or not, open a browser window and type the following URL.

```
http://localhost:7001/B2BSample/MessageService?WSDL
```

This should display the WSDL implemented by this service.

4. Use the `redeploy` target to rebuild and redeploy the web service if any modifications have been implemented.

```
ant redeploy
```

5. Use the `undeploy` target to undeploy the web service from the WebLogic Server.

```
ant undeploy
```

6. Use `clean-all` target to clean this web service build, as shown below.

```
ant clean-all
```

Using JMS Transport

Like web services, there are two scenarios for the JMS transport, inbound and outbound.

Outbound JMS Transport

In the outbound scenario, a JMS client (based on WebLogic stack) is used to send the XML document to AquaLogic Service Bus. This client posts the XML document to the `b2b.sb.jms.EdIQueue` queue and the AquaLogic Service Bus proxy should be configured on the same queue.

The steps for building and running the JMS client are as follows:

1. Open the command window and navigate to the `samples` directory shown below.

`WL_HOME\samples\edi\examples\B2BEDISample`

2. Use the `build-jms-client` ant target is used to build the JMS client based on the WebLogic stack.

```
ant build-jms-client
```

By default this client sends the XML document to the `b2b.sb.jms.EDIQueue` queue, where the AquaLogic Service Bus proxy is listening for the incoming message. If the AquaLogic Service Bus proxy is configured to a different queue, then modify the `JMSQueueSender.java` file (located in `src\com\bea\wli\b2b\sample\jmsclient` directory) and rebuild.

3. Once the build is complete, change to the `bin` directory, where you will find the batch and shell script files required for running the client.
4. Send the XML documents to AquaLogic Service Bus using either one of the platform specific executable files.

Windows: `run-jms-sender.bat`

Unix: `run-jms-sender.sh`

These files expect parameters defined in the following table.

Parameter	Description
<code><XML File Path></code>	Complete path and file name of the XML file to be sent to AquaLogic Service Bus.
<code><Document Name></code>	Name of the EDI document to which this XML file belongs. For example: 850, 855, ORDERS and so on.
<code><Document Version></code>	Specify the version to which this EDI document belongs. For example: 4020, 4030, D98B and so on.
<code><Document Class></code>	Specify the standard to which this EDI document belongs. For example: X12 or EDIFACT.

5. Use the `clean-client` target to clean the client files, as shown below.

```
ant clean-client
```

Inbound JMS Transport

In the inbound scenario, AquaLogic Service Bus sends the XML messages to JMS queue configured in AquaLogic Service Bus. This sample is configured such that, by default, this message will be sent to the JMS Queue namely `b2b.sb.jms.EDIQueue_Receive`. The functionality of this client is that it saves whatever the JMS messages it receives to the file system location specified as part of the startup.

The steps for building and running the JMS client are as follows.

1. Open the command window and navigate to the samples directory shown below.

```
WL_HOME\samples\edi\examples\B2BEDISample
```

2. Build the JMS client based on the WebLogic stack, using the `build-jms-client` ant target, as shown below.

```
ant build-jms-client
```

By default, this client receives the XML document from the `b2b.sb.jms.EDIQueue_Receive` queue, where AquaLogic Service Bus sends the XML message. If the AquaLogic Service Bus business service is configured to a different queue, modify the `JMSQueueReceiver.java` file (located in the `src\com\bea\wli\b2b\sample\jmsclient` directory) and rebuild.

3. Once the build is complete, change to the `bin` directory, where you will find the batch and shell script files required for running the client.
4. Receive the XML documents from AquaLogic Service Bus using either one of the platform specific executable files.

Windows: `run-jms-receiver.bat`

Unix: `run-jms-receiver.sh`

These files expect parameter defined in the following table.

Parameter	Description
<Directory Path>	Complete directory path and name to save the XML files obtained from AquaLogic Service Bus.

5. Use the `clean-client` target to clean the client files, as shown below.

```
ant clean-client
```

