BEA AquaLogic Service Bus™

Tutorials

Version: 2.5
Document Revised: October 2006

Copyright

Restricted Rights Legend

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

Contents

Introduction to the BEA AquaLogic Service Bus Tutorials

What is BEA AquaLogic Service Bus? 1-1

AquaLogic Service Bus Concepts 1-2

Overview of the Tutorials 1-4

Focus of the Tutorials 1-5

How to Use the Tutorials 1-5

Getting Started with the BEA AquaLogic Service Bus Tutorials

About the Scenarios 2-1

Setting Up the Tutorials 2-2

Step 1: Install AquaLogic Service Bus 2-3

Step 2: Change the Configuration Settings 2-3

Step 3: Start BEA AquaLogic Service Bus 2-7

Step 4: Build and Deploy the Business Services 2-7

Step 5: Log in to the AquaLogic Service Bus Console 2-8

Where to Go from Here 2-9

Tutorial 1. Routing a Loan Application

Prerequisites 3-1

Tutorial Objectives 3-1

Features Highlighted 3-2

Definition of the Scenario 3-2

Overview of the Run Time Process in AquaLogic Service Bus 3-3

Required Resources 3-3

Steps in This Tutorial 3-4

Step 1: Prepare the Environment 3-4

Step 2: Create a Session and Set up a Project 3-4

Step 3: Create the WSDL Resources 3-7

Step 4: Create a Proxy Service. 3-9

Step 5: Create the Business Services 3-16

Step 6: Configure the LoanGateway1 Proxy Service 3-21

Step 7: Test Your Loan Application Routing Configuration 3-27

Tutorial 2. Transforming a Loan Application

Prerequisites 4-1

Tutorial Objectives 4-1

Definition of the Scenario 4-2

Overview of the Run Time Process in AquaLogic Service Bus 4-2

Required Resources 4-3

Steps in This Tutorial 4-3

Step 1: Prepare the Environment 4-4

Step 2: Create a Session and Set up a Project 4-4

Step 3: Create the Resources 4-5

Step 4: Configure the Routing for the LoanGateway2 Proxy Service. 4-10

Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service 4-13

Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service 4-21

Step 7: Add a Default Routing Case 4-23

Step 8: Test the Loan Application Routing Configuration. 4-24

Best Practices 4-27

Tutorial 3. Validating a Loan Application

Prerequisites. 5-2

Tutorial Objectives 5-2

Features Highlighted 5-2

Definition of the Scenario 5-2

Overview of the Process in AquaLogic Service Bus 5-3

Required Resources 5-4

Steps in This Tutorial. 5-4

Step 1: Prepare Your Environment. 5-5

Step 2: Create a Session and Set up a Project 5-5

Step 3: Create the Resources 5-5

Step 4: Configure the LoanGateway3 Proxy Service. 5-6

Step 5: Add a Default Routing Case. 5-16

Step 6: Test the Loan Application Routing Configuration. 5-18

Step 7: Using Reporting and Monitoring 5-21

CHAPTER 1

Introduction to the BEA AquaLogic Service Bus Tutorials

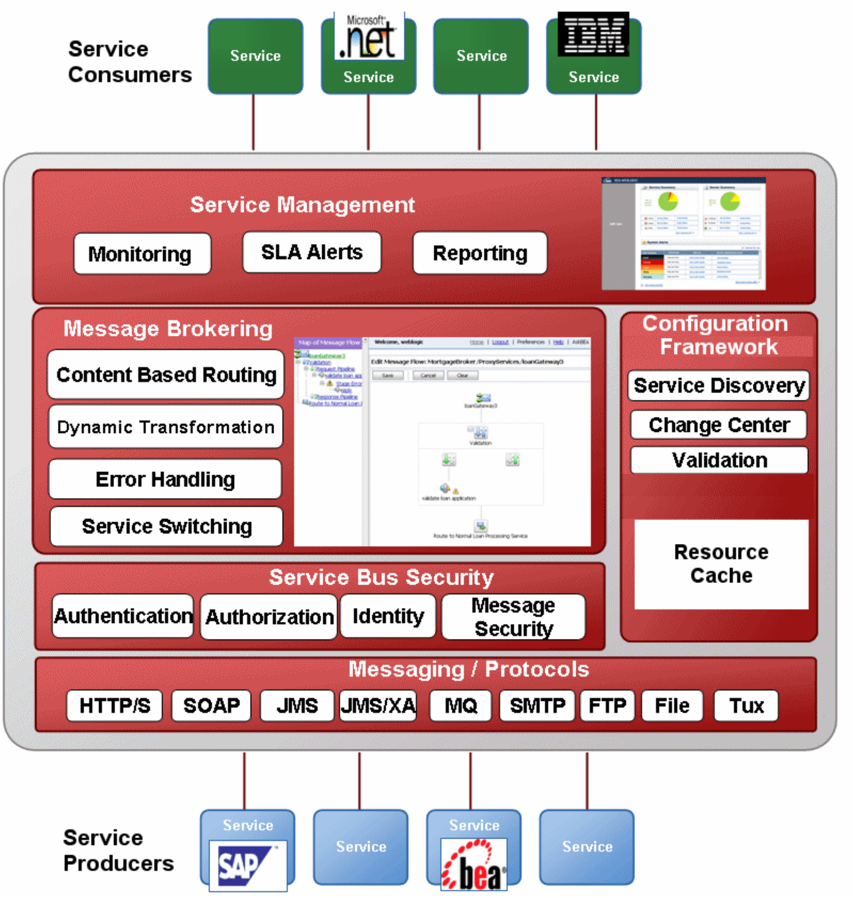
What is BEA AquaLogic Service Bus?

BEA AquaLogic Service Bus™ is a configuration-based, policy-driven Enterprise Service Bus (ESB). It is targeted for service-oriented integration, managing Web Services, and providing conventional message brokering across heterogeneous IT environments. AquaLogic Service Bus manages the routing and transformation of messages in an enterprise system to promote seamless application integration.

The AquaLogic Service Bus Console enables you to configure a proxy service to suit your needs and to manage Web Services by controlling the service and policy configurations and by monitoring system and operations tasks. AquaLogic Service Bus relies on WebLogic Server® run-time facilities. For more information, see *BEA AquaLogic Service Bus Concepts and Architecture.*

Figure 1-1 illustrates the intermediary function of AquaLogic Service Bus.

Figure 1-1 AquaLogic Service Bus Architecture



AquaLogic Service Bus Concepts

AquaLogic Service Bus allows you to manage Web Services and deliver authentic message brokering through the configuration of proxy services in the AquaLogic Service Bus Console. It is used to resolve differences in requirements between business services and service clients. The concepts underlying AquaLogic Service Bus are described in this section. For more information about AquaLogic Service Bus, see *BEA AquaLogic Service Bus Concepts and Architecture.*

This section describes the benefits of using AquaLogic Service Bus as an intermediary between clients and business services. It focuses on the configuration of the proxy service. AquaLogic Service Bus is also used to manage Web Services. The configuration functions are separated from the management functions in the AquaLogic Service Bus Console. You can therefore implement AquaLogic Service Bus independently and dynamically, as driven by your business needs without costly infrastructure development efforts.

AquaLogic Service Bus provides intelligent message brokering between business services (such as enterprise services and databases) and service clients (such as presentation applications or other business services) through proxy services that you configure using AquaLogic Service Bus Console. **Proxy services** are AquaLogic Service Bus definitions of intermediary Web Services that AquaLogic Service Bus implements locally on WebLogic Server. With AquaLogic Service Bus message brokering, service clients exchange messages with an intermediary proxy service rather than working directly with a business service.

A proxy service can route messages to multiple business services; you can choose to configure a proxy service with an interface that is independent of the business services with which the proxy service communicates. In such cases, you can configure a message flow definition to route a message to the appropriate business service and map the message data into the format required by the business service's interface.

Business services are remote services or external end points that are typically not implemented by AquaLogic Service Bus. They are definitions of the enterprise services with which you can exchange messages. For more information on business services, see Business Services in *Using the AquaLogic Service Bus Console*.

In AquaLogic Service Bus, a **Message Flow** is the implementation of a proxy service. You configure the logic for the manipulation of messages using proxy service message flow definitions. This logic includes such activities as transformation, publishing, and reporting, which are implemented as individual actions within the stages of a pipeline. Pipelines are one-way processing paths that include no branching.

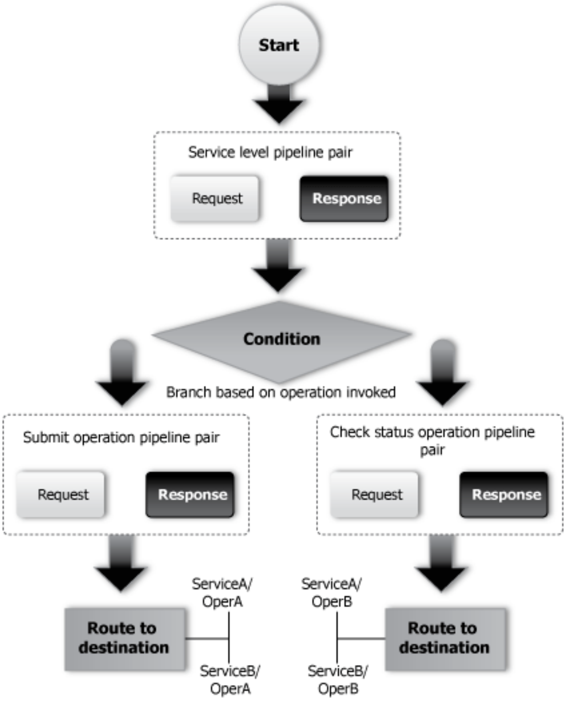
Each pipeline is a sequence of stages containing actions. A **stage** is a user-configured processing step. Messages fed into the pipelines are accompanied by a set of message context variables that contain the message contents. They can be accessed or modified by actions in the pipeline stages.

Pipeline pairs are request and response pipelines. The request pipeline definition specifies the actions that AquaLogic Service Bus performs on request messages to the proxy service before invoking a business service or another proxy service. The response pipeline definition specifies the processing that AquaLogic Service Bus performs on responses from the business or proxy service that the proxy service invokes before returning a response to a client.

Each pipeline consists of a sequence of stages, each stage containing actions. However, a single service-level request pipeline might optionally branch out into operational pipelines (you can configure one default operational pipeline at most one per operation). The determination of the operation is done through user-selected criteria. The response processing starts with the relevant operation pipeline which then joins into a single service-level response pipeline.

Figure 1-2 shows an example of operation pipelines in a proxy service.

Figure 1-2 Example of Operation Pipelines in a Proxy Service



For more information, see "Pipeline Pairs" in *BEA AquaLogic Service Bus Concepts and Architecture*. For information about how to use the WebLogic Service Bus Console to configure proxy services and other resources, see *Using the AquaLogic Service Bus Console.*

Overview of the Tutorials

These tutorials show how AquaLogic Service Bus is used to route a loan application to appropriate business services based on the configurations set in the proxy service. The

AquaLogic Service Bus Console comprises JSP-based portlets that support the configuration and design of the proxy service message flows and the associated resources.

Focus of the Tutorials

These tutorials highlight specific features available in AquaLogic Service Bus. In the tutorials, you will use the console to create the proxy service that will interact with the service client and business services. As you work through the tutorial, you will:

- Learn how to configure proxy services to process the input and output messages in a message flow.
- Create, read, and update configuration objects (such as proxy services and WSDLs).
- See how AquaLogic Service Bus uses session management to allow configuration changes locally, and then instantiate the changes in the core environment.
- Configure resources such as WSDLs, Service Callouts, XQueries, and business services.
- Route messages according to XQuery-based policies or callouts to external Web Services.
- See how BEA AquaLogic Service Bus manages synchronous and asynchronous messages.
- Define a transformation on an XML message, based on XQuery, supporting a callout to a Web Service to gather additional data for the transformation.
- See how message processing by AquaLogic Service Bus is driven by metadata specified as the *message flow definition* for a proxy service in the AquaLogic Service Bus Console.

How to Use the Tutorials

The tutorials represent typical use case scenarios for BEA AquaLogic Service Bus. You will learn how to configure and use BEA AquaLogic Service Bus to resolve the business scenarios presented in the specified use cases. The tutorials will:

- Provide you an understanding of the concepts underlying BEA AquaLogic Service Bus.
- Show you how to navigate through the AquaLogic Service Bus Console and demonstrate the capabilities of the console.
- Provide the basic steps that will help you to configure specific scenarios.

Complete the first tutorial, Routing a loan application, to become familiar with BEA AquaLogic Service Bus, then proceed to the other tutorials provided as part of the tutorial suite:

- Tutorial 1. Routing a Loan Application is based on scenarios that demonstrate how AquaLogic Service Bus facilitates the routing of messages within an enterprise.
- Tutorial 2. Transforming a Loan Application describes how a proxy service is configured with a routing node, transformations, and a Web Service callout (Service Callout) to allow content-based routing and message enrichment through AquaLogic Service Bus.
- Tutorial 3. Validating a Loan Application describes how a BEA AquaLogic Service Bus proxy service can be configured to validate a message passing from a client to a business service.

Document Conventions

The following conventions are used throughout this document:

- The instructions in the tutorial documentation are for Windows command shells. If you are using a UNIX-based shell, substitute / for \ in path names.
- `BEA_HOME` represents the directory in which you installed your BEA product; for example, c:\bea.

Related Documents

- *Introducing BEA AquaLogic Service Bus Examples*
- *BEA Products Installation Guide*
- *BEA AquaLogic Service Bus Concepts and Architecture*
- *AquaLogic Service Bus Console Online Help*

Getting Started with the BEA AquaLogic Service Bus Tutorials

In this chapter, you learn to:

- Set up the environment
- Configure resources
- Start BEA AquaLogic Service Bus
- Create the solutions that implement the business cases contained in the tutorial

Complete the steps in this section before you start the tutorials.

The AquaLogic Service Bus tutorials guide you through designing and configuring AquaLogic Service Bus resources and deploying a configuration. These tutorials use the AquaLogic Service Bus example suite as a basis for instruction. (See *BEA AquaLogic Service Bus Examples Guide*)

About the Scenarios

The tutorials describe a typical mortgage broker scenario for a loan application request. A primary mortgage company uses AquaLogic Service Bus to route loan applications to appropriate business services. The loan applications are routed to different business services depending on qualifying criteria, such as the requested interest rate and the requested principal amount. The applicant's credit rating information is required to complete the loan request when the principal amount is greater than US$ 25 million.

Complete the first tutorial to become familiar with AquaLogic Service Bus. The subsequent tutorials build on what you learn in the first tutorial. These tutorials explore the basic functionality of AquaLogic Service Bus, including message routing and enrichment.

Setting Up the Tutorials

The files (including pre-built business services) that support the building of the tutorial solutions are located in the following directory:

BEA_HOME\weblogic92\samples\servicebus\examples

The tutorial files contain properties that you must set to run the tutorials. In the tutorial, you can use four different business services. Each business service also has a set of files associated with it. The files structure for each of the business services is the same—the directory structure for one of the business services, the normal service, is described in Table 2-1. You can explore the other directories as an exercise.

Table 2-1 Sample Files Provided in Support of the Tutorials

Filename/Location	Description
`/src/setEnv.cmd`	This file sets the environment properties.
`/src/examples.properties`	This file contains business service build and deployment properties.
For each business service:	
BEA_HOME`/weblogic92/samples/servicebus/examples/src/examples/webservices/jws_basic/normal`	The files contained in this folder are those required by the normal business service. The same files structure exists for the other three business services used in this tutorial (`manager`, `large`, and `credit`).
`SimpleBean.java`	The jws-181 Web Service definition.
`LoanStruct.java`	The definition of the message.
`build.xml`	The build script that is used to build each of the services using `ant`.
`*.wsdl`	A Web Service Definition Language (WSDL) file defines each of the services.
`/client/Main.java`	The Java client with which each of the business services interacts.

You must execute the following steps to set up and run the tutorials:

- Step 1: Install AquaLogic Service Bus
- Step 2: Change the Configuration Settings
- Step 3: Start BEA AquaLogic Service Bus
- Step 4: Build and Deploy the Business Services
- Step 5: Log in to the AquaLogic Service Bus Console

Step 1: Install AquaLogic Service Bus

1. Install AquaLogic Service Bus. For instructions, see *AquaLogic Service Bus Installation Guide.*

2. Create a new AquaLogic Service Bus domain in which to develop the tutorials. The Configuration Wizard guides you through the process of creating a new domain using the configuration templates.

   For this tutorial, in the Configuration Wizard, ensure that you:

   – Select the **AquaLogic Service Bus** check box when you select the domain source.

   – Select **Development Mode** and the default PointBase database.

   – Create a domain named `ServiceBusTutorial`.

   For instructions on how to use the configuration wizard, see *Creating WebLogic Domains Using the Configuration Wizard*.

Step 2: Change the Configuration Settings

To develop and run the AquaLogic Service Bus tutorials you must modify some of the configuration properties. The associated files are located in the following directory:

BEA_HOME`\weblogic92\samples\servicebus\examples`

There are two subdirectories: \build and \src. Make the configuration changes to the files in the \src directory.

Table 2-2 Configuration Settings

Folder Name	Description
\build	This folder contains the files required to run the pre-configured AquaLogic Service Bus examples provided in the AquaLogic Service Bus Examples domain.
\src	This folder contains the files that support the development and execution of the AquaLogic Service Bus tutorial. While developing the tutorial you will modify some of the files contained in this folder, such as examples.properties.

Edit the examples.properties File

To deploy the business services on your running tutorial domain, you must modify the properties that specify the WebLogic Server behavior. You can modify the properties in the *WebLogic configuration* section of the following file:

BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties

To Modify the examples.properties File

1. Change the hostname to the name of the machine on which the ServiceBusTutorial domain is running.
2. Change the port number to the port number of the tutorial domain.
3. Enter the username that you specified when creating the domain.
4. Enter the password that you specified when creating the domain.
5. Change the server name to the name you specified when you created the domain (if you specified a name other than the default name (AdminServer) assigned by the Configuration Wizard).

Listing 2-1 Fragment of examples.properties File

```
### WebLogic configuration ###

-- Configure per your WLS installation.

wls.hostname=localhost
wls.port=7021

wls.username=weblogic

wls.password=weblogic

wls.server.name=xbusServer
```

Edit and Run the setEnv Script

Edit the `setEnv` script (`setEnv.cmd` on Windows or `setEnv.sh` on UNIX systems) in the following directory to replace the default path with the path to your tutorial domain's `setDomainEnv` script.

BEA_HOME`\weblogic92\samples\servicebus\examples\src`

The new path is:

BEA_HOME`\user_projects\domains\ServiceBusTutorial\bin\setDomainEnv.cmd`.

where *BEA_HOME*`\user_projects` is the default directory in which you created your `ServiceBusTutorial` domain. (Use `setDomainEnv.cmd` or `setDomainEnv.sh`, as appropriate for your operating system.)

To Run the setEnv Script

On Windows:

1. Open a command window.
2. Change to the following directory:

   BEA_HOME`\weblogic92\samples\servicebus\examples\src\examples`
3. In the examples directory, on the command line, run the following command:

   `..\setEnv.cmd`

On UNIX:

```
% cd BEA_HOME/weblogic92/samples/servicebus/examples/src/examples

% ../setEnv
```

Where:

- *BEA_HOME* is the directory in which you installed AquaLogic Service Bus.

Edit the setDomainEnv.cmd File—Windows 2000 Users Only

This step is only for Windows 2000 users. On Windows 2000 machines, a class path length problem prevents you from running ant tasks and executing the tutorial. This work around uses the `subst` command to shorten the classpath passed by ant. Follow these steps to shorten the classpath:

1. Edit the `setDomainEnv.cmd` file located in the Service Bus domain installation location, `*BEA_HOME*\user_projects\domains\ServiceBusTutorial\bin\`

2. In the `setDomainEnv.cmd` file search for the instruction that sets the `PRE_CLASSPATH`. The instruction looks like:

   ```
   set
   PRE_CLASSPATH=%PRE_CLASSPATH%;%WL_HOME%\servicebus\lib\sb-public.jar;%
   ```

3. Identify three unused drive letters (for example, X, Y, and Z) and add the following code to the beginning of the file:

   ```
   subst X: %WL_HOME%\servicebus\lib
   subst Y: %WL_HOME%\integration\common\lib
   subst Z: %WL_HOME%\server\lib
   ```

4. Edit the `PRE_CLASSPATH` entry, replacing the path with the appropriate drive letter. For the example:

   ```
   set
   PRE_CLASSPATH=%PRE_CLASSPATH%;X:\sb-public.jar;X:\sb-internal.jar;Y:\wl
   icommon.jar;Y:\qs_p13n_system.jar;X:\xbus-core.jar;Z:\wlxbean.jar;Z:\xq
   uery.jar;Z:\apache_xbean.jar;Z:\binxml.jar
   ```

5. Save the changes and close the `setDomainEnv.cmd` file.

6. Run the `setDomainEnv.cmd` script by entering the following commands:

   a. `cd` *BEA_HOME*`\user_projects\domains\ServiceBusTutorial\bin\`

   b. `run setDomainEnv.cmd`

Step 3: Start BEA AquaLogic Service Bus

Start AquaLogic Service Bus using one of the following methods:

- From the Windows Start menu, select:

  Start Programs→BEA Products→User Projects→`domain_name`**→Start Server for AquaLogic Service Bus Domain**

  Where:

  - `domain_name` is the name you assigned your AquaLogic Service Bus domain when you created it.

- On Windows systems, invoke `startWebLogic.cmd` from the root of the AquaLogic Service Bus domain.

- On UNIX systems, invoke `startWebLogic.sh` from the root of the AquaLogic Service Bus domain.

  A server command console window, displays status information about Weblogic Server.

Note: If you created a production mode domain, enter a valid username and password to start the server. For more information about domains, see *Creating WebLogic Domains Using the Configuration Wizard*.

Step 4: Build and Deploy the Business Services

You must start the server before building the Web Services. See "Step 3: Start BEA AquaLogic Service Bus" on page 2-7. After the server is started, you must run a build script to:

- Build and deploy Web Services
- Create a Java client to test the Web Services
- Execute the Java client using `ant run`

To Build and Deploy the Business Services

1. In the command window, ensure that you have run the `setEnv` script, as described in "To Run the setEnv Script" on page 2-5.

2. Change to the `webservices\jws_basic` directory.

```
cd
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
s\jws_basic
```

The directory contains four business service folders. Complete step 3 to step 5 to build and run each of the business services.

3. Change to the normal directory:
   `cd normal`

   Note: To build the other business services (credit, large, and manager), change to the appropriate business service folder.

4. To deploy the Web Service, enter `ant`, then press **Enter**.

   A message indicating whether the command was successfully executed is displayed on the console.

5. To test the deployed Web Service, Enter `ant run`, then press **Enter.**

   A message from the business service is displayed at the command line. The message indicates whether the Web Service was run successfully.

6. To build the remaining Web Services (credit, large, and manager) that you will use in this tutorial, repeat step 3 through step 5 of these build and deploy instructions for each of the folders listed in the `\jws_basic` directory.

Step 5: Log in to the AquaLogic Service Bus Console

To log in to the AquaLogic Service Bus Console, open a browser window and enter the following URL in your browser:

```
http://host:port/sbconsole
```

where `host:port` represents the name and port number of the machine on which WebLogic Server is running. For example, if WebLogic Server is running on your local machine using the default port configuration, enter the following URL in your browser:

```
http://localhost:7021/sbconsole
```

The AquaLogic Service Bus Console login page is displayed. Enter the username and password that you specified when you created the AquaLogic Service Bus domain.

Where to Go from Here

After you complete the steps required to set up the tutorials, you can proceed to "Tutorial 1. Routing a Loan Application" on page 3-1, which describes how you can configure BEA AquaLogic Service Bus with the resources required for the routing scenario.

Each of the tutorials include instructions to design and configure the AquaLogic Service Bus resources, followed by test sections that you can use to test the completed configurations.

Tutorial 1. Routing a Loan Application

BEA AquaLogic Service Bus enables message routing between business services in the enterprise environment. Messages can be routed from a client through the AquaLogic Service Bus intermediary, to the appropriate business service. BEA AquaLogic Service Bus routes messages to one or more destinations based on the actions taken on the messages. Configure those actions using the AquaLogic Service Bus Console. Routing provides an efficient alternative to building a multitude of point-to-point Web Services connections between pairs of systems.

This section includes the following topics:

- Prerequisites
- Tutorial Objectives
- Definition of the Scenario
- Steps in This Tutorial

Prerequisites

Complete "Setting Up the Tutorials" on page 2-2.

Tutorial Objectives

The objective of the tutorial is to create and test a routing scenario developed using the graphical environment provided in AquaLogic Service Bus Console. It includes the following tasks:

- Importing Web Service Definition Language (WSDL) files

- Registering business services
- Creating a proxy service
- Configuring content-based routing

Features Highlighted

This tutorial introduces you to the following features of BEA AquaLogic Service Bus:

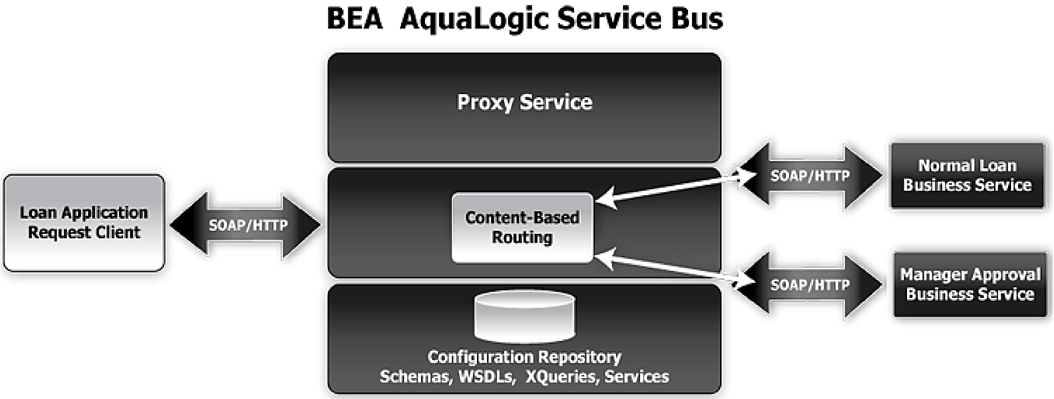
- The AquaLogic Service Bus Console
- Implementation details of proxy services: message flows and actions
- Editing XQuery expressions using the XQuery Expression Editor
- Working with context variables to manipulate the content of the message

Definition of the Scenario

A primary mortgage company uses BEA AquaLogic Service Bus to route loan applications to appropriate business services based on the interest rate requested. An application containing a request for a rate less than 5% requires management approval and is routed to an appropriate business service for processing. All other loan applications are routed to another business service for processing. The target business service responds, indicating whether the loan application is approved or rejected.

Figure 3-1 summarizes the logical architecture to support this scenario. It illustrates how BEA AquaLogic Service Bus mediates the messaging between the clients and the business services in your enterprise.

Figure 3-1 Expose a Loan Application Request Web Service via AquaLogic Service Bus


BEA AquaLogic Service Bus
Proxy Service
Loan Application Request Client
SOAP/HTTP
Content-Based Routing
SOAP/HTTP
Normal Loan Business Service
SOAP/HTTP
Manager Approval Business Service
Configuration Repository
Schemas, WSDLs, XQueries, Services

Overview of the Run Time Process in AquaLogic Service Bus

A client sends a loan application to a proxy service named **LoanGateway1**. The default proxy service has a conditional routing stage that checks the value of the requested interest rate in the loan application document. If the interest rate is less than 5%, the loan request is routed to the **managerLoanReviewService** business service, otherwise it is routed to the **NormalLoan** business service. The target business service returns a response message similar to the following message:

```
Loan Application Response: APPROVED BY THE <i><b>MANAGER</b></i> LOAN
APPLICATION REVIEW SERVICE.
```

Note: `MANAGER` is replaced with `NORMAL` when the **normalLoan** business service is used at run time.

Required Resources

Table 3-1 lists the resources required to develop and run the *Routing a Loan Application* tutorial.

Table 3-1 Routing Loan Application Tutorial Resources

Resource Name	Description
normalLoan	This is the WSDL resource

Table 3-1 Routing Loan Application Tutorial Resources

NormalLoan	This is an external business service used by AquaLogic Service Bus
managerLoanReviewService	This is an external business service used by AquaLogic Service Bus
LoanGateway1	This is the AquaLogic Service Bus proxy service

Steps in This Tutorial

In this tutorial, BEA AquaLogic Service Bus is used to route a loan application within a mortgage company to a target a Web Service depending on the interest rate requested.

Complete these steps to design and configure a proxy service and the associated resources in AquaLogic Service Bus to resolve this user case scenario:

- Step 1: Prepare the Environment
- Step 2: Create a Session and Set up a Project
- Step 3: Create the WSDL Resources
- Step 4: Create a Proxy Service
- Step 5: Create the Business Services
- Step 6: Configure the LoanGateway1 Proxy Service
- Step 7: Test Your Loan Application Routing Configuration

Step 1: Prepare the Environment

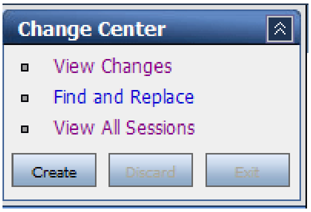
Ensure that AquaLogic Service Bus is running in the domain that you have created for the tutorial and that you have completed the steps described in "Setting Up the Tutorials" on page 2-2.

Step 2: Create a Session and Set up a Project

You must first start a session in the AquaLogic Service Bus Console. You can view resources and configurations in the AquaLogic Service Bus Console without being in a session, but cannot edit, change, or delete items or properties in a project, unless you have an active session. The **Change**

Center manages sessions in the console as in Figure 3-2. All the sessions are saved as temporary files until the session is activated.

Figure 3-2 Manage Sessions Using Change Center



The Change Center has several functions that are summarized in Table 3-2.

Table 3-2 Description of Functions of the Change Center

Click…	To…
Create	Create a new session. You must create a session to make changes to the configuration. You can exit this session using the Exit function, then re-enter it by clicking Edit. Once in the session, you can continue to modify the resources.
Edit	Enter a session that you previously created and exited. You must activate a session before you can use Edit.
Exit	Discontinue the session and save the session state. Click Edit to re-enter the discontinued session.
Discard	End the current session without saving the changes.
Activate	Save the current session's configuration to the run time. When you click Activate, the Activate Session page is displayed. In this page, the user and session name of the active session are displayed. If required, you can add a description. Click Submit to save the current session's configuration to the run time.

Note: It is a good practice to save your configuration changes incrementally.

BEA recommends that you click **Create** before making any changes whenever you change a configuration in the console and that you click **Activate** when you have completed your changes to the configuration.

Create the MortgageBroker Project and File Folders

This section describes how you create a MortgageBroker project under the Projects folder in the AquaLogic Service Bus Console. For this scenario, you will also create three folders in the MortgageBroker project—to hold each of the resource types to be defined for the project as follows:

- ProxyService
- BusinessServices
- WSDL

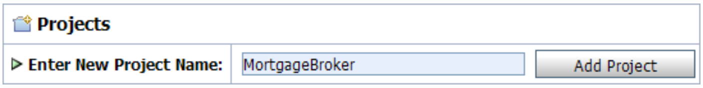
Note: In the AquaLogic Service Bus Console, you can add new projects and navigate the project trees using the **Project Explorer**. These Project folders can be collapsed and expanded as required. All project configurations are performed in the project page. Selecting a project or one of its sub-folders from the Project Explorer displays the information for that entity on the project page.

To Start a Session and Create a Project

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation pane, select **Project Explorer.**

   The Project Explorer opens in the navigation pane and a project page is displayed on the console.
3. In the **Enter New Project Name** field in the **Projects** section, type **MortgageBroker** as in Figure 3-3.

Figure 3-3 Create a Project



Projects

▷ Enter New Project Name: MortgageBroker

4. Click **Add Project**.

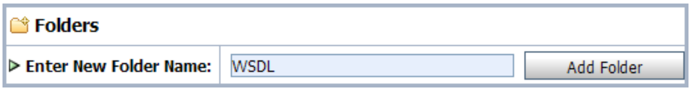
The MortgageBroker project is created and listed in Project Explorer under Projects.

You should create the WSDL folder first, and add a WSDL resource to it. The WSDLs are the basis on which you create the business services and the proxy service. You subsequently create other folders and resources for this scenario.

To Create a Project Folder

1. In Project Explorer, click the **MortgageBroker** project to open the associated project page.
2. On the Project Page, in **Folders**, enter the folder name in the field provided. In this case, enter `WSDL` as displayed in Figure 3-4.

Figure 3-4 Create a Folder



3. Click **Add Folder**.

   The WSDL folder is displayed in the list of project folders for the MortgageBroker project.

4. Repeat step 2 and step 3 to create the following additional folders: **ProxyService** and **BusinessService**.
5. When all three folders are created, click **Activate**, then **Submit** to save the project directory structure (see Table 3-2).

Step 3: Create the WSDL Resources

AquaLogic Service Bus resources are configured using configuration wizards. Each configuration wizard includes a sequence of pages that prompt you for information about the resource and provide property configuration options.

Create the normalLoan Service WSDL Resource

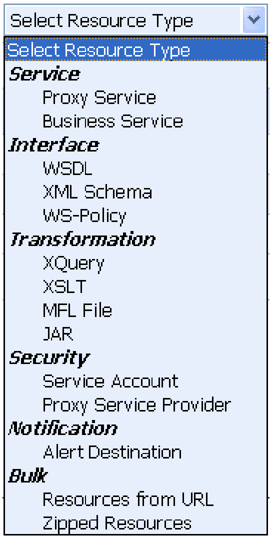
A WSDL defines the public contract (interface specification) between a client and a service, whether the service is a proxy service or business service. It is the formal description of a Web Service. A WSDL is used to describe what a Web Service's interface is, where it resides, and how to invoke it. You create the WSDL resource first as subsequent service registration steps depend on it. The WSDL is subsequently used to register the business service to create the proxy service.

To Import a WSDL

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation pane, select **Project Explorer.**
3. In Project Explorer, click **MortgageBroker** to expand the folder, then click **WSDL** to open the WSDL folder.

4. In the **Resources** pane, from the **Select Resource Type** drop-down list, select **WSDL**.

Figure 3-5 Select a Resource Type



A page that allows you to create a new WSDL resource is displayed as in Figure 3-6.

5. Enter the following information in the Create a New WSDL Resource page:

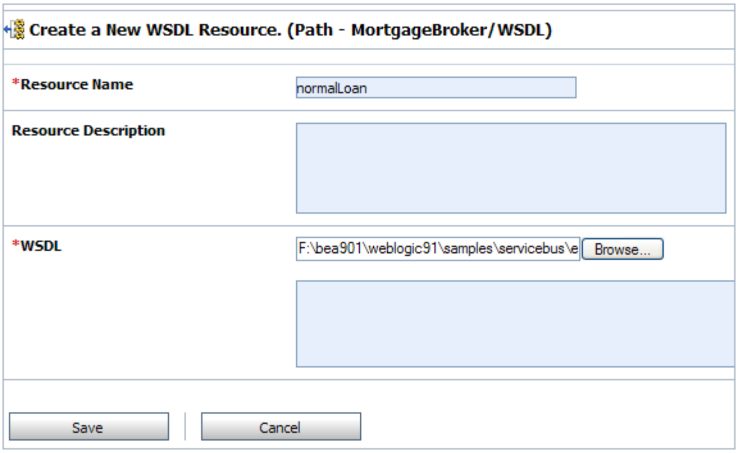
   a. Enter the resource name **normalLoan**.

   b. Click **Browse...** to locate the WSDL associated with the normal loan process:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
   ices\jws_basic\normal\NormalLoanApprovalService.wsdl
   ```

   c. You can optionally enter information in the Resource Description text box to document the resource.

Figure 3-6 Create a WSDL Resource



Create a New WSDL Resource. (Path - MortgageBroker/WSDL)

*Resource Name: normalLoan

Resource Description:

*WSDL: F:\bea901\weblogic91\samples\servicebus\e

d. Click **Save** to create the WSDL resource.

The resource is now displayed in the resources pane.

This step completes the creation of a WSDL resource, **normalLoan**, in the WSDL folder.

Create the ManagerApproval Service WSDL Resource

Import the WSDL associated with the ManagerApproval Service. Use the procedure "To Import a WSDL" on page 3-7 to do this using the following parameters in Table 3-3.

Table 3-3 ManagerApproval Service WSDL

Configuration Parameter	Value
Resource Name	`managerApproval`
ManagerApprovalService WSDL	`BEA_HOME\weblogic92\servicebus\examples\src\examples\webservices\jws_basic\manager\ManagerApprovalService.wsdl`

Step 4: Create a Proxy Service

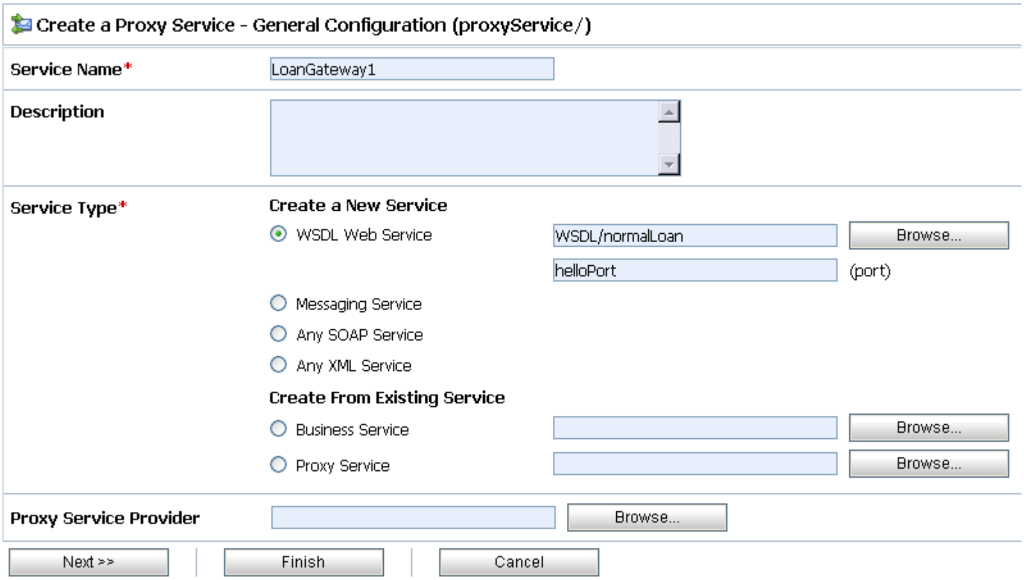
In this step, you will create a proxy service. The proxy service is used to route the loan application to the appropriate business service.

To Create the Proxy Service

1. In Project Explorer, select the **MortgageBroker** project. The project folder is expanded to show the directory structure of the project.
2. Select the **ProxyService** folder. The MortgageBroker/ProxyService page is displayed.
3. In the **Select Resource Type** drop-down list, select **Proxy Service**.

   The Create a Proxy Service - General Configuration page is displayed as in Figure 3-7.

Figure 3-7 Create a Proxy Service



Create a Proxy Service - General Configuration (proxyService/)

Service Name* LoanGateway1

Description

Service Type*

Create a New Service

● WSDL Web Service WSDL/normalLoan

helloPort (port)

○ Messaging Service

○ Any SOAP Service

○ Any XML Service

Create From Existing Service

○ Business Service

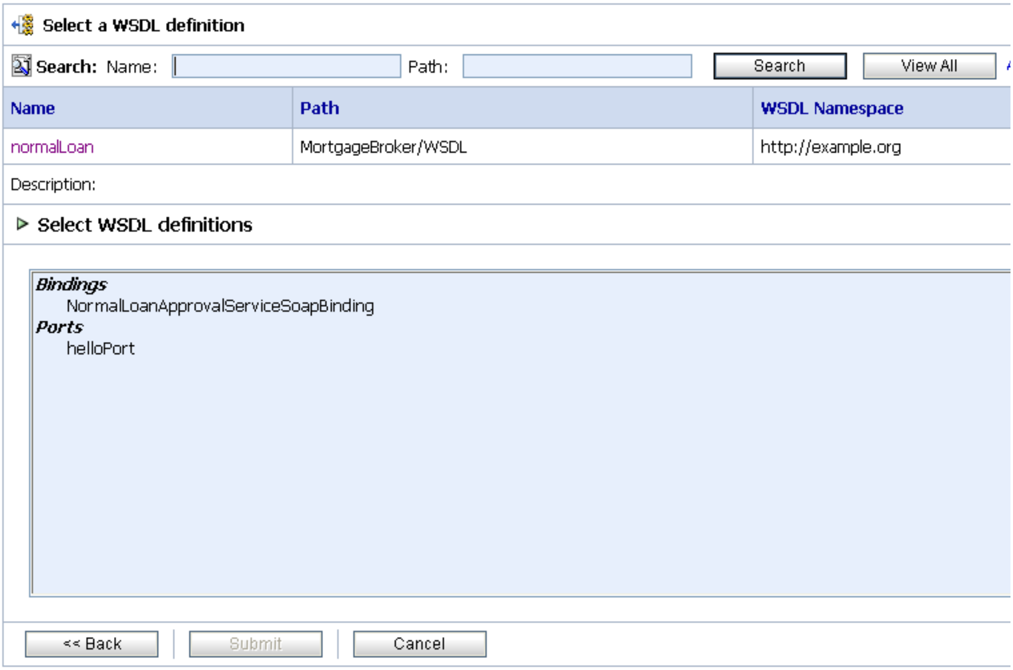
○ Proxy Service

Proxy Service Provider

4. Name the proxy service, **LoanGateway1**.
5. In Service Type, select **WSDL Web Service**, then click **Browse...**

   The Select a WSDL browser window is displayed. The proxy service is based on the WSDL resource that you originally created, hence you must reference the resource here.
6. In the Select a WSDL definition window as in Figure 3-8, select the **normalLoan** WSDL. The **WSDL Definitions** pane is populated with the content categories of the WSDL.

Figure 3-8 Select WSDL Definition



Select a WSDL definition

Name	Path	WSDL Namespace
normalLoan	MortgageBroker/WSDL	http://example.org

Description:

Select WSDL definitions

Bindings
NormalLoanApprovalServiceSoapBinding
Ports
helloPort

a. In the Definitions pane, from the Ports category, select the WSDL port for this WSDL (`helloPort`).

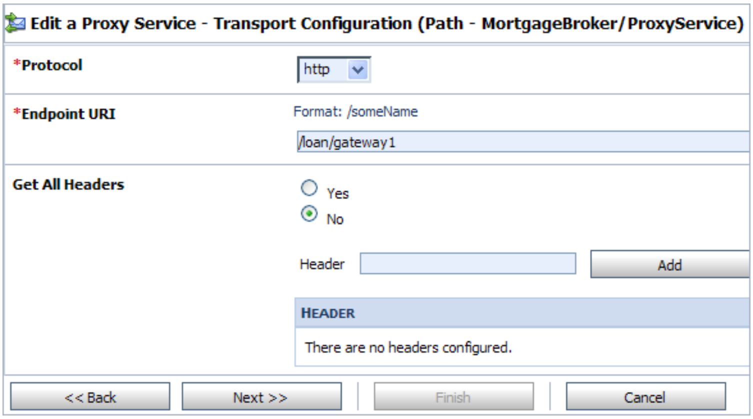
b. Click **Submit** to complete your selection.

The WSDL port text field is populated with the port name as in Figure 3-7.

7. Click **Next** to continue configuring the proxy service.

The Create a Proxy Service - Transport Configuration page is displayed as in Figure 3-9.

Figure 3-9 Transport Configuration of Proxy Service



Edit a Proxy Service - Transport Configuration (Path - MortgageBroker/ProxyService)

*Protocol	http
*Endpoint URI	Format: /someName /loan/gateway1
Get All Headers	○ Yes ● No Header [ ] Add

HEADER
There are no headers configured.

8. Accept the default protocol, **http**.

   Note: The protocol you select on this page determines the format for the Endpoint URI that you will specify in the next step. **http** is the default protocol since it is assumed by most services.

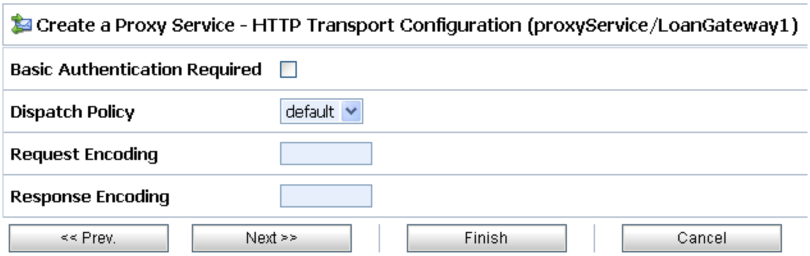
9. Set the Endpoint URI to `/loan/gateway1`. This is the URI to which the client will send its messages.

10. Accept the default for the **Get All Headers** option (**No**).

11. Click **Next**.

    The Create a Proxy Service - HTTP Transport Configuration page is displayed as in Figure 3-10.

Figure 3-10 HTTP Transport Configuration of Proxy Service



Create a Proxy Service - HTTP Transport Configuration (proxyService/LoanGateway1)

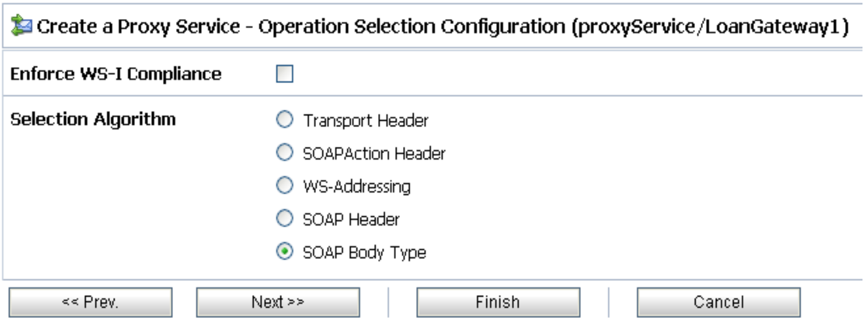
Basic Authentication Required	□
Dispatch Policy	default
Request Encoding	
Response Encoding	

You do not need to change configuration settings on this page as a security policy has not been defined for the Web Service.

12. Click **Next** to accept the default settings for the HTTP transport configuration.

    The Create a Proxy Service - Operation Selection Configuration page is displayed as in Figure 3-11.

Figure 3-11 Operation Selection Configuration of Proxy Service



Create a Proxy Service - Operation Selection Configuration (proxyService/LoanGateway1)

Enforce WS-I Compliance □

Selection Algorithm
○ Transport Header
○ SOAPAction Header
○ WS-Addressing
○ SOAP Header
● SOAP Body Type

13. Accept the default selection algorithm (**SOAP Body Type**).

    Note: On this page you are selecting the metadata accompanying the message that identifies which Web Service operation to invoke.

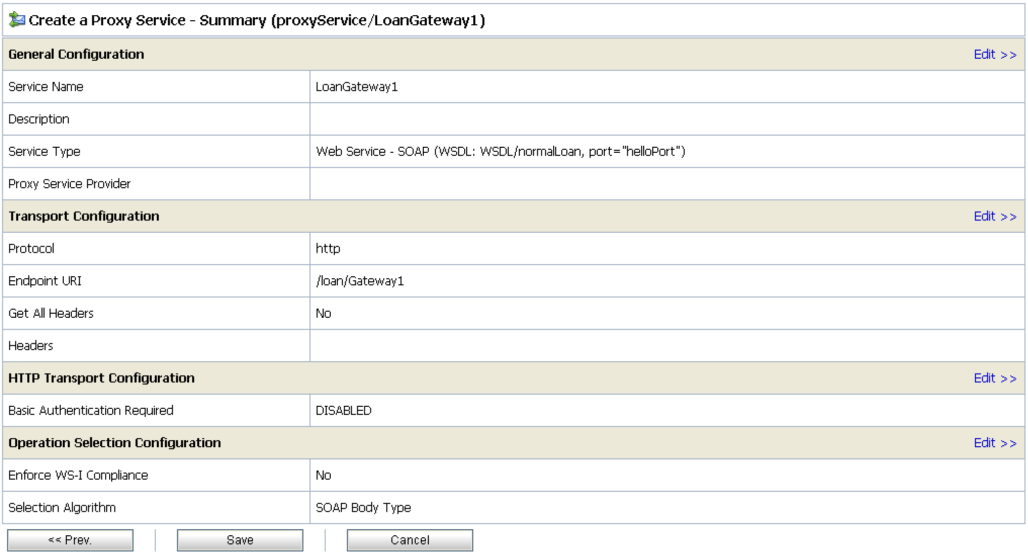
    The default service is an RPC Web Service. As a result, the SOAP body contains the service operation selection information. An operation in a Java Web Service WSDL corresponds to a public method in the Java Web Service. If more than one method is accessible to a client (that is, there are multiple public methods), then each public method type will have an operation definition in the WSDL. For a SOAP based RPC service the SOAP body indicates the operation (method) selected by the client.

    Note: The SOAP header can also specify the operation. However, by convention, the SOAP body defines this.

14. Click **Next**.

    A summary of the configuration settings for the proxy service is displayed in Figure 3-12.

Figure 3-12 Proxy Service - Configuration Settings Summary



Create a Proxy Service - Summary (proxyService/LoanGateway1)

General Configuration	Edit >>
Service Name	LoanGateway1
Description	
Service Type	Web Service - SOAP (WSDL: WSDL/normalLoan, port="helloPort")
Proxy Service Provider	
Transport Configuration	Edit >>
Protocol	http
Endpoint URI	/loan/Gateway1
Get All Headers	No
Headers	
HTTP Transport Configuration	Edit >>
Basic Authentication Required	DISABLED
Operation Selection Configuration	Edit >>
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type

You can review the configuration settings and change them, if necessary, before registering the proxy service.

15. When satisfied with the proxy service configuration settings, click **Save** to register the service.

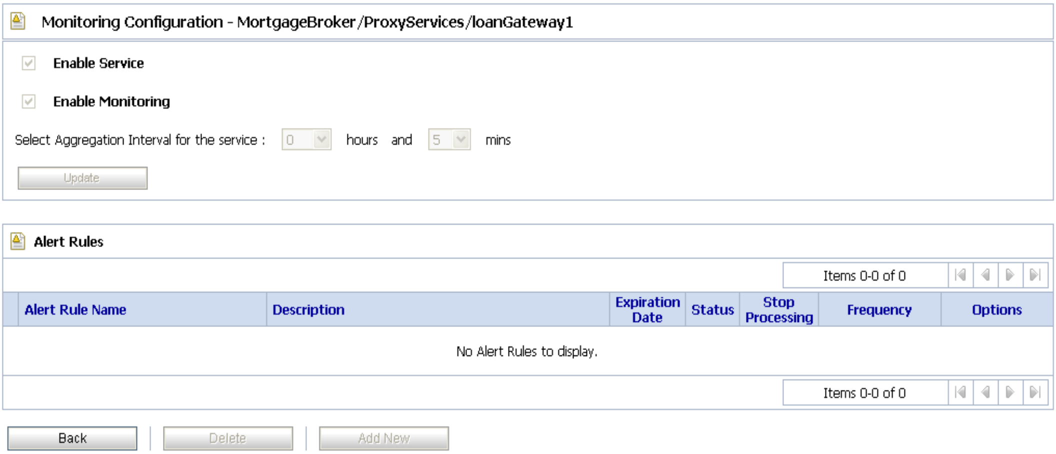
    Note: You must always select **Save** prior to activating a session configuration in the Change Center.

    The **LoanGateway1** proxy service is displayed in the Resources pane in the **ProxyService** folder of the **MortgageBroker** project.

16. To turn on monitoring for the LoanGateway1 proxy service, click the Manage Monitoring icon associated with the **LoanGateway1** proxy service in the Resources pane.

The Monitoring Configuration page, which you can use to enable monitoring for this service, is displayed as in Figure 3-13.

Figure 3-13 Monitoring Configuration of Proxy Service



Monitoring Configuration - MortgageBroker/ProxyServices/loanGateway1

☑ Enable Service

☑ Enable Monitoring

Select Aggregation Interval for the service : 0 hours and 5 mins

Alert Rules

Items 0-0 of 0

Alert Rule Name	Description	Expiration Date	Status	Stop Processing	Frequency	Options
No Alert Rules to display.						

Items 0-0 of 0

To enable monitoring:

a. Select the **Enable Monitoring** check box.

b. Enter an aggregation interval for the service. This interval is the period over which aggregated statistics related to the service are computed for display in the dashboard of the AquaLogic Service Bus Console.

For more information about using the AquaLogic Service Bus Dashboard and monitoring services, see:
- Monitoring in *AquaLogic Service Bus Console Online Help*
- Monitoring in *BEA AquaLogic Service Bus User Guide*

c. Click **Update**.

d. Click **Back** in the Monitoring Configuration page, to return your browser to the **MortgageBroker** project, **ProxyService** folder page.

This step completes the configuration of the **LoanGateway1** proxy service in the AquaLogic Service Bus Console.

Step 5: Create the Business Services

In the routing a loan scenario, when the interest rate requested on a loan application is greater than or equal to 5%, then the loan application is routed to the normal loan processing service for approval. If the interest rate requested on a loan application is less than 5%, then the loan application must be approved by a manager, therefore it is routed to a manager approval service.

Complete this step to:

- Create the NormalLoan Business Service
- Create the managerLoanReviewService Business Service

Create the NormalLoan Business Service

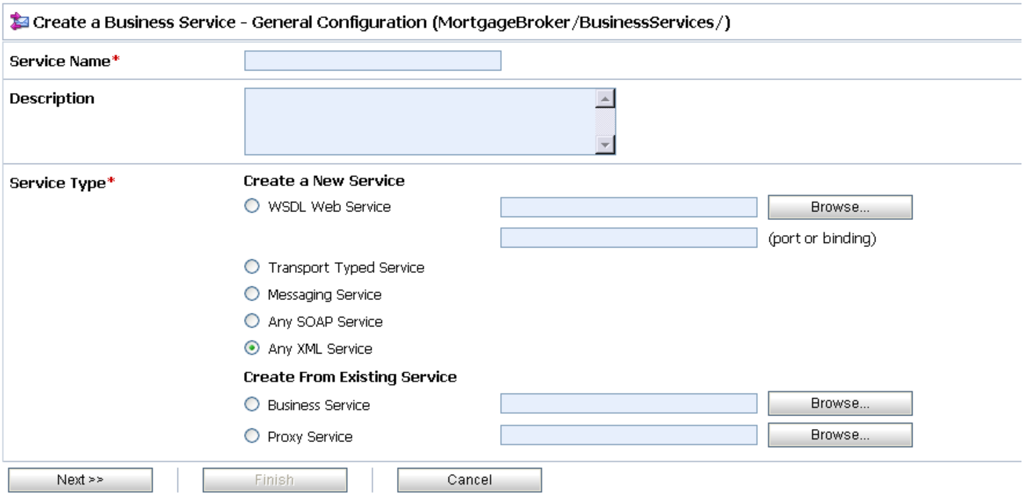
1. In Project Explorer, select the BusinesService folder from the MortgageBroker project tree.

   Note: If the folders are not displayed in the project tree, click **Activate**, then **Submit** to register the changes in the session. The folders that you created for the MortgageBroker project are displayed.
   Click **Create** to make changes to the session.

2. In the **Select Resource Type** drop-down list, select **Business Service**.

   The Create a Business Service - General Configuration page is displayed as in Figure 3-14.

Figure 3-14 Create Business Service



Create a Business Service - General Configuration (MortgageBroker/BusinessServices/)

Service Name*

Description

Service Type*

Create a New Service

○ WSDL Web Service (port or binding)

○ Transport Typed Service

○ Messaging Service

○ Any SOAP Service

● Any XML Service

Create From Existing Service

○ Business Service

○ Proxy Service

3. Enter a name for the service. In this case, enter `NormalLoan`.
4. For Service Type, select **WSDL Web Service**, then click **Browse...** to locate the WSDL and port on which to base the business service:

   a. In the WSDL Browser select the `normalLoan` WSDL

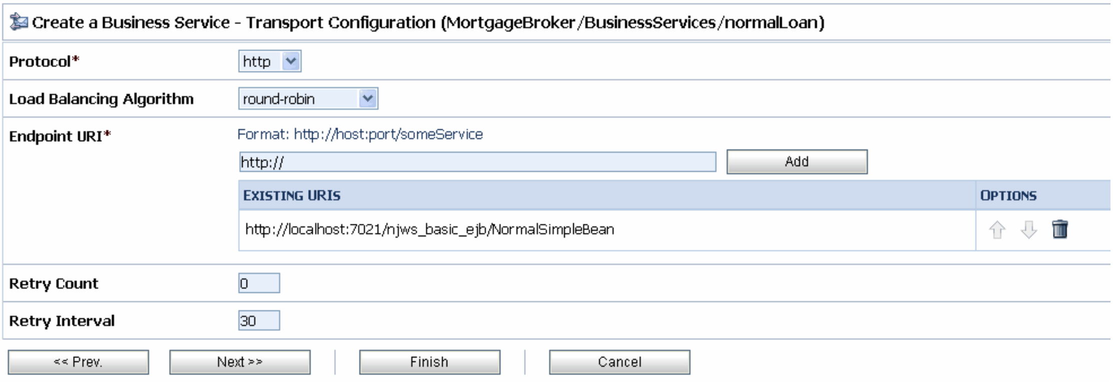
   b. Select the port (`helloPort`) for the WSDL.

   c. Click **Submit**.

   The text field for the WSDL port service type is populated with the value selected on the Create a Business Service - General Configuration page.
5. Click **Next**.

   The Create a Business Service - Transport Configuration page is displayed as in Figure 3-15.

Figure 3-15 Transport Configuration of a Business Service



Create a Business Service - Transport Configuration (MortgageBroker/BusinessServices/normalLoan)

Protocol*	http
Load Balancing Algorithm	round-robin
Endpoint URI*	Format: http://host:port/someService http://
Retry Count	0
Retry Interval	30

EXISTING URIS	OPTIONS
http://localhost:7021/njws_basic_ejb/NormalSimpleBean	⇧ ⇩ 🗑

6. Accept the default Protocol, **http**.
7. Select **none** for the Load balancing Algorithm, since load balancing is relevant only when a service specifies multiple end points.

   When you specify multiple end points and a load balancing algorithm, if one end point is overloaded or not available at run time, the message can be dispatched to the next service in the list of end point URIs. In this case there is only one service, therefore accepting the default behavior or selecting **none** does not effect the behavior of the proxy service at run time.

8. The End point URI of the business service is the end point URI on the server on which the service is deployed. Ensure that the pre-populated value for the End point URI in the **Existing URIs** field is `http://<host:port>/njws_basic_ejb/NormalSimpleBean`

   Where:

   `<host:port>` represents the machine and port on which your AquaLogic Service Bus server is running. The host and port values must be the same as the values specified in `BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties`

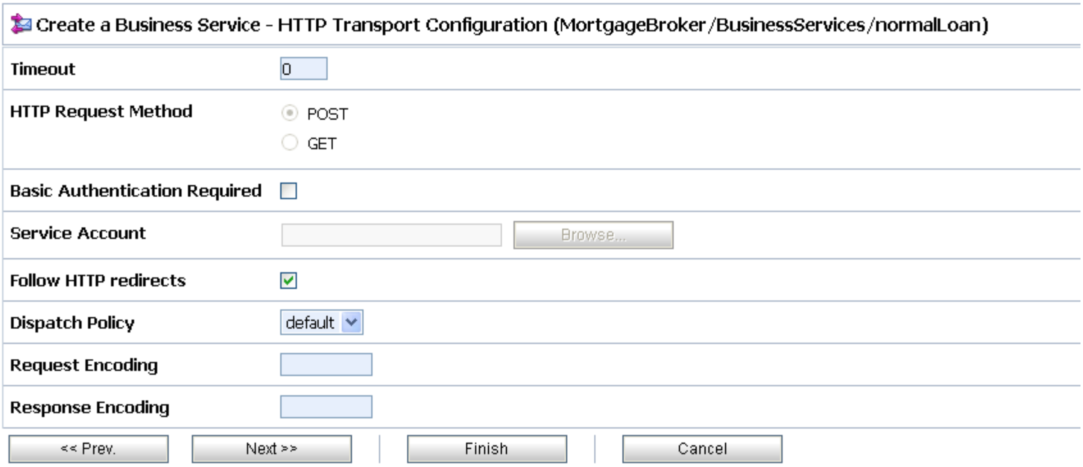
   For more information, see "Edit the examples.properties File" on page 2-4.

   Note: Delete any invalid end point URI in the existing URI list.

9. Accept the default settings for the remaining options on the page and click **Next**.

   The Create a Business Service-HTTP Transport Configuration page is displayed as in Figure 3-16.

Figure 3-16 HTTP Transport Configuration of Business Service



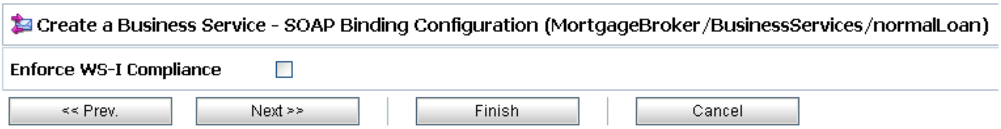
Create a Business Service - HTTP Transport Configuration (MortgageBroker/BusinessServices/normalLoan)

Timeout	0
HTTP Request Method	● POST ○ GET
Basic Authentication Required	☐
Service Account	Browse...
Follow HTTP redirects	☑
Dispatch Policy	default
Request Encoding	
Response Encoding	

10. Accept the default settings on this page, then click **Next**.

    The Create a Business Service-SOAP Binding Configuration page is displayed as in Figure 3-17.

Figure 3-17 SOAP Binding Configuration of Business Service



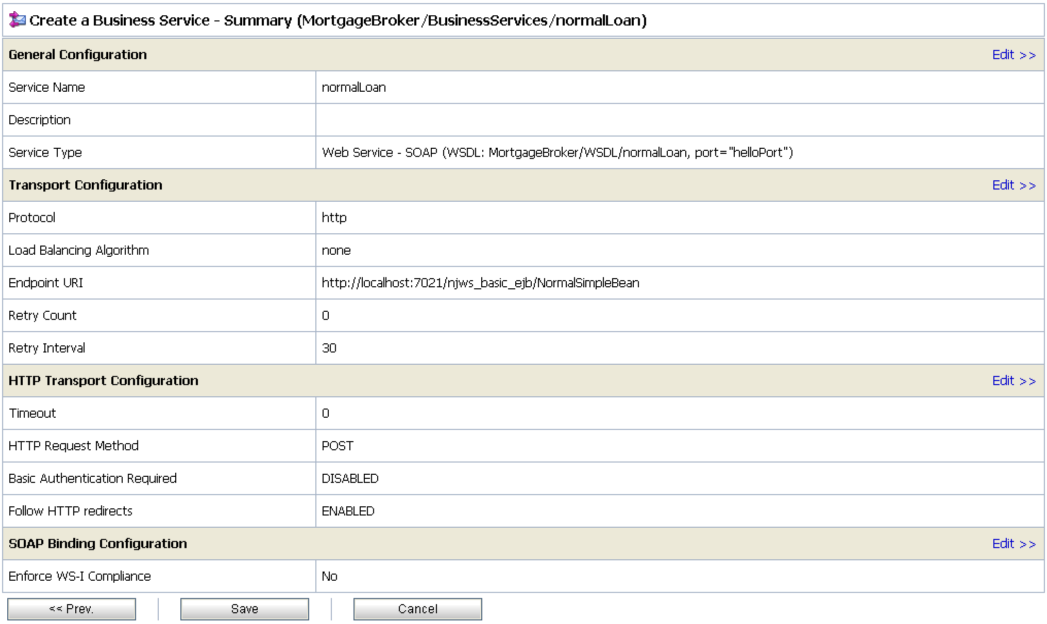
Create a Business Service - SOAP Binding Configuration (MortgageBroker/BusinessServices/normalLoan)

Enforce WS-I Compliance □

11. Accept the default settings on this page, then click **Next**.

    The Create a Business Service-Summary page is displayed as in Figure 3-18. You can now check if all the settings of the business service have been implemented correctly.

Figure 3-18 Business Service - Configuration Settings Summary



Create a Business Service - Summary (MortgageBroker/BusinessServices/normalLoan)

General Configuration	Edit >>
Service Name	normalLoan
Description	
Service Type	Web Service - SOAP (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Transport Configuration	Edit >>
Protocol	http
Load Balancing Algorithm	none
Endpoint URI	http://localhost:7021/njws_basic_ejb/NormalSimpleBean
Retry Count	0
Retry Interval	30
HTTP Transport Configuration	Edit >>
Timeout	0
HTTP Request Method	POST
Basic Authentication Required	DISABLED
Follow HTTP redirects	ENABLED
SOAP Binding Configuration	Edit >>
Enforce WS-I Compliance	No

12. Click **Save** to accept the configuration settings.

13. To turn on monitoring for the **NormalLoan** business service, click the Manage Monitoring icon associated with the **NormalLoan** business service in the Resources pane.

    The Monitoring Configuration page, which you can use to enable monitoring for this service, is displayed.

To enable monitoring:

a. Select the **Enable Monitoring** check box.

b. Enter an aggregation interval—the period over which aggregated statistics related to the service are computed for display in the AquaLogic Service Bus Console, specifically the Dashboard.

   For more information about using the AquaLogic Service Bus Dashboard and monitoring services, see:
   - Monitoring in *AquaLogic Service Bus Console Online Help*

   - Monitoring in *BEA AquaLogic Service User Guide*

c. Click **Update**.

d. Click **Back** in the Monitoring Configuration page to return your browser to the **MortgageBroker** project, **BusinessService** folder page.

You have completed the configuration of the NormalLoan business service on AquaLogic Service Bus Console.

Create the managerLoanReviewService Business Service

To create the business service to which a loan application is routed when the interest rate requested in that loan application is less than 5%.

Follow the same steps as described in the preceding section: "Create the NormalLoan Business Service" on page 3-16, but use the configuration parameters in the following table.

Table 3-4 ManagerLoanReviewService Business Service

Configuration Parameter	Value
Business Service Name	`managerLoanReviewService`
ManagerApprovalService WSDL	Select the managerApproval WSDL you created based on: `*BEA_HOME*\weblogic92\servicebus\examples\src\examples\webservices\jws_basic\manager\ManagerApprovalService.wsdl` **Note:** Select the **`helloPort`** port.
EndpointURI	`http://*<host:port>*/mjws_basic_ejb/ManagerSimpleBean`

Summary

When you complete "Step 1: Prepare the Environment" on page 3-4 to "Step 5: Create the Business Services" on page 3-16, you have created two business services (`NormalLoan` and `managerLoanReviewService`), a proxy service (`LoanGateway1`) and a WSDL (`normalLoan`) that are the resources required for the tutorial.

Once you have configured the proxy service with a base configuration you can proceed to the next step (Step 6: Configure the LoanGateway1 Proxy Service) to complete the configuration of the proxy service by adding the routing behavior for the loan application.

Step 6: Configure the LoanGateway1 Proxy Service

AquaLogic Service Bus Message Flows define the implementation of proxy services. Message flows can include zero or more pipeline pairs: request and response pipelines for the proxy service (or for the operations on the service) and error handler pipelines that can be defined for stages, pipelines, and proxy services. Pipelines can include one or more stages, which in turn include actions. To configure the routing behavior of the **LoanGateway1** proxy service, you must complete the following for the message flow:

- Configure the Routing Behavior of the Message Flow
- Add a Default (Else) Condition to the Routing Table

Configure the Routing Behavior of the Message Flow

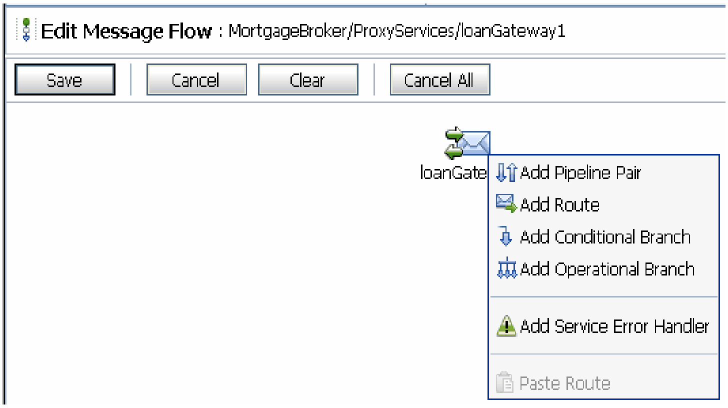
1. In Project Explorer, select the ProxyService folder from the MortgageBroker project tree.

   Note: You must be in a session to edit resources. If you have not already done so, click **Create** in the Change Center to begin a session to configure the proxy service and edit the message flow.

2. In the **Actions** column associated with the **LoanGateway1** proxy service, click the **Edit Message Flow** icon .

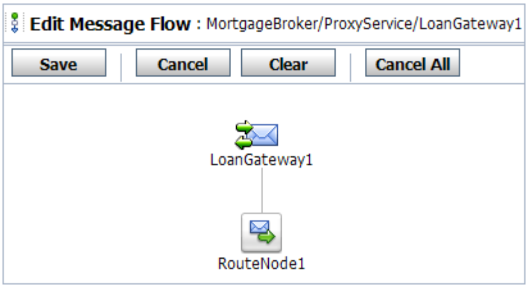
   The Edit Message Flow page for the proxy service **LoanGateway1** is displayed as in Figure 3-19.

Figure 3-19 Configure Message Flow for LoanGateway1 Proxy Service



3. Click **LoanGateway1**, then select **Add Route Node**. `RouteNode1` is added in the configuration dialog as in Figure 3-20.

Figure 3-20 Add Route Node

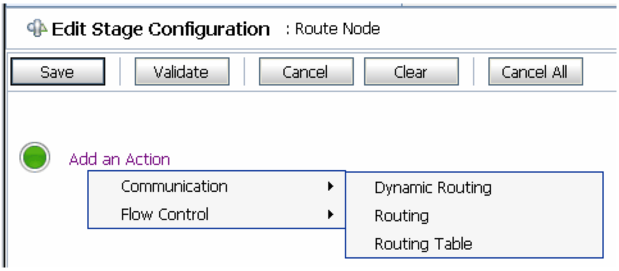

Edit Message Flow : MortgageBroker/ProxyService/LoanGateway1
Save
Cancel
Clear
Cancel All
LoanGateway1
RouteNode1

4. Click **RouteNode1** and from the pop-up menu, select **Edit→Route Node**.

   The Edit Stage Configuration: Route Node page is displayed. The page contains a single link—**Add an Action** as in Figure 3-21.

   Note: A stage is an element of a pipeline and also a container for actions defined in a pipeline. Actions are the elements of a pipeline stage that define the handling of messages as they flow through a proxy service at run time.

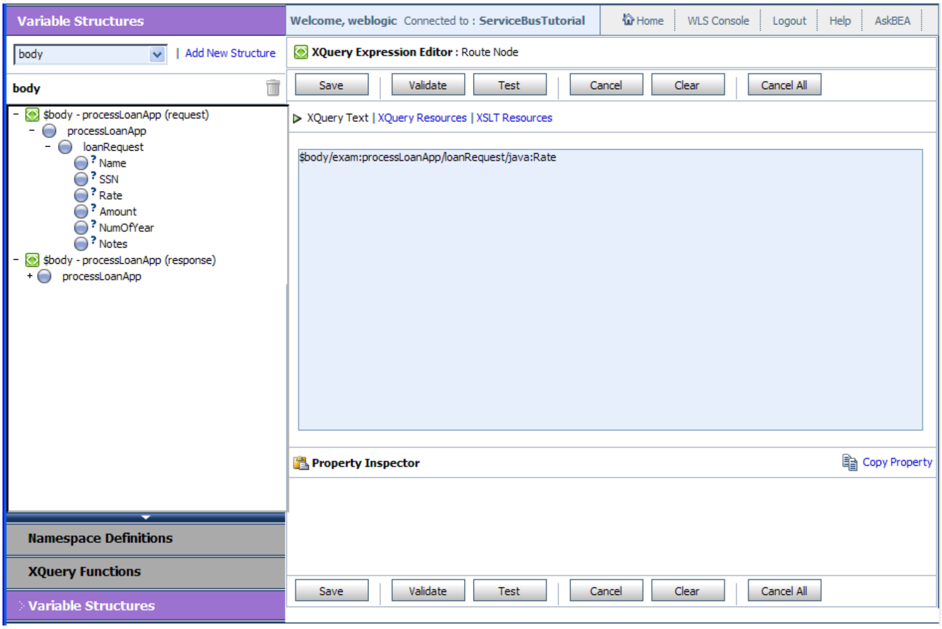
Figure 3-21 Add an Action



5. Click **Add an Action**, then select **Communication→Routing Table**.
6. The Edit Stage Configuration page changes to display routing table configuration page.
7. You must configure the routing table to route messages to business services based on evaluating the interest rate element of the incoming message. You must create an XQuery expression, using the XQuery Expression Editor.
8. In the Routing table, click <Expression>. The XQuery Expression Editor page is displayed as in Figure 3-22.
9. Accept the default selection of XQuery Text in the page.
10. Click Variable Structures. The Variable Structures pane is displayed.
11. Select body from the Select Structure drop-down list in the Variable Structures pane.
12. A structural representation of the body element is displayed in the Variable Structures pane.
13. Click + beside the processLoanApp element to expand the element.
14. The processLoanApp element is within the $body - processLoanApp (request) element.
15. Click + to expand the loan request element.
16. A graphical representation of the structure of the loan request document is displayed.
17. At run time, the proxy service makes its routing decision based on the value in the rate element of the message.
18. Select the circle icon associated with the Rate node element, drag and drop it in the XQuery Expression text box.
19. The XQuery expression $body/exam:processLoanApp/loanRequest/java:Rate is written in the text box.

Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the rate element in the Variable Structures pane. The expression is displayed in the Property Inspector palette. Place the cursor in the XQuery expression text box and click Copy Property. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XQuery Expression text box.

Figure 3-22 XQuery Expression Editor



```
$body/exam:processLoanApp/loanRequest/java:Rate
```

20. Click **Validate** to validate the XQuery.

    It is a good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the **Validate** button. In this case, the expression is valid.

21. Click **Save**.

    The routing table is displayed on the Edit Stage Configuration page. `<Expression>` is replaced by the expression that returns the value of the **rate** element in the message.

22. In the **Operator** drop-down list, select **<**.

23. In the associated text field enter the number **5**.

The routing table now contains an expression that determines the routing behavior. If the value in the rate element is less than 5, you must route according to the routing table configuration.

24. Click the **Service** link to define the service to which you want to route messages when the rate is less than five. The Select Service browser window is displayed.

    a. Select the **managerLoanReviewService** business service.

    b. Click **Submit**.

25. In the Routing Table Operation drop-down list, select the **processLoanApp** operation. This is the operation on the **managerLoanReviewService** business service that is invoked at run time if the rate requested in the loan request is less than 5.

You have now defined the case to route the loan application to the **managerLoanReviewService** business service.

If the rate specified in the loan application is equal to or greater than five percent, then the message is routed to the **normalLoan** business service. The next section describes how you can add a condition to the routing table to account for this case (which is the default case).

Add a Default (Else) Condition to the Routing Table

1. Click the **Case** icon, then select **Insert Default Case**. The else condition (default case) is added to the routing table.

2. In the else condition, click the **Service** link. The Select Service browser window is displayed.

    a. Select the **NormalLoan** business service.

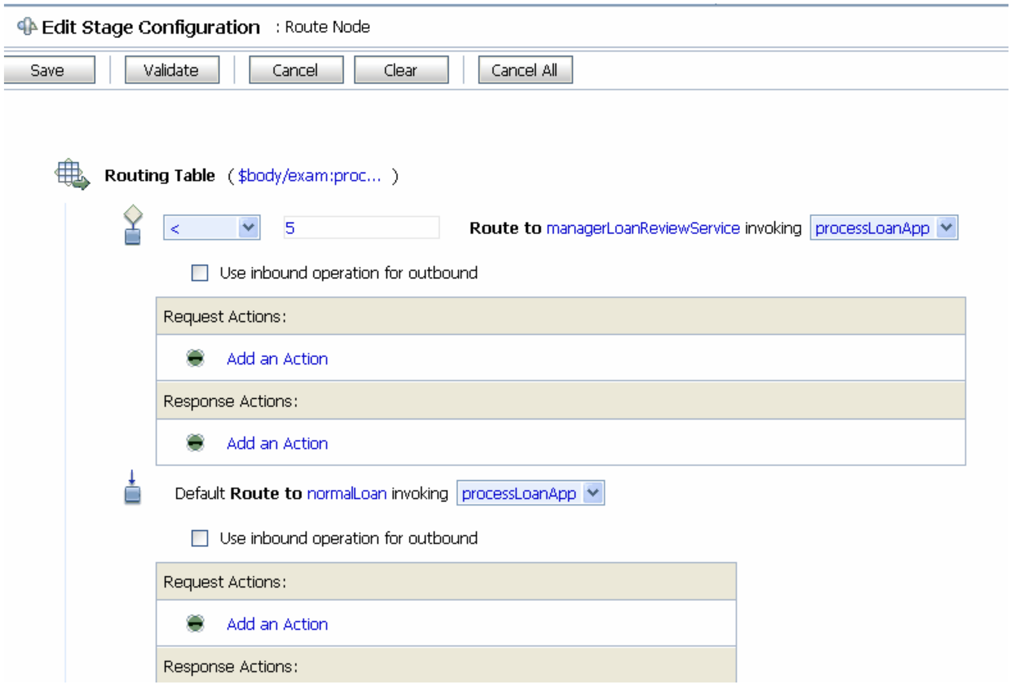
    b. Click **Submit**.

3. In the Default Case operation drop-down list, select the **processLoanApp** operation for the service.

    This step completes the configuration of the message flow in the route node: **RouteNode1**.

    After you have completed these steps, the Edit Stage Configuration page is displayed as in Figure 3-23.

Figure 3-23 Routing Behavior of the Message Flow



Edit Stage Configuration : Route Node

Routing Table ($body/exam:proc...)

< 5 Route to managerLoanReviewService invoking processLoanApp

□ Use inbound operation for outbound

Request Actions:

Response Actions:

Default Route to normalLoan invoking processLoanApp

□ Use inbound operation for outbound

Request Actions:

Response Actions:

4. Click **Validate** to validate the routing configuration.

   Note: It is a good practice to do this before you save the configuration.

5. Click **Save** on the Edit Stage Configuration page to save the configuration and return to the Edit Message Flow page.

6. Click **Save** on the Edit Message Flow page and return to the MortgageBroker/ProxyServices page.

7. In the Change Center, click **Activate**, then **Submit** to save the session and to instantiate all the configuration changes made during the session.

8. Click **Validate** to validate the routing configuration.

   Note: It is a good practice to do this before you save the configuration.

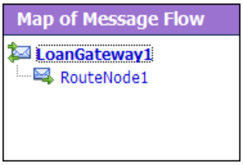
9. Click **Save** on the Edit Stage Configuration page to save the configuration and return to the Edit Message Flow page.

10. Click **Save** on the Edit Message Flow page and return to the MortgageBroker/ProxyServices page.

11. In the Change Center, click **Activate**, then **Submit** to save the session and to instantiate all the configuration changes made during the session.

You can view the map of the message flow of the proxy service as in Figure 3-24.

Figure 3-24 Map of Message Flow of the LoanGateway1 Proxy Service



Map of Message Flow

LoanGateway1

RouteNode1

Summary

"Step 6: Configure the LoanGateway1 Proxy Service" on page 3-21 completes the configuration of the logic in the Routing Table that determines the run time routing behavior for messages. A message is routed to the **managerLoanReviewService** business service if the rate specified in the loan application is less than 5, otherwise the message is routed to the **normalLoan** business service.

Step 7: Test Your Loan Application Routing Configuration

Now that you have configured AquaLogic Service Bus to work with the client and the target business services, you can test the configuration. Testing is done from the command line by changing the value of the interest rate field in the `build.xml` file and then running `ant`. The routing behavior for the proxy service in this scenario is based on the interest rate requested in the loan application message. A change in the value of the interest rate results in a change in the routing behavior as follows:

- A value of 4.9 or less causes the message to be routed to the **managerLoanReviewService** business service.

- A value of 5.0 or greater causes the message to be routed to the `normalLoan` service.

This section describes how you can test the proxy service for both cases. The message returned in the command window after running a test indicates the success or failure of the test and identifies the business service that processed the request.

This section includes the following test scenarios:

- Testing the managerLoanReviewService Business Service
- Testing the normalLoan Business Service

To run this test correctly, ensure that WebLogic Server is started in the AquaLogic Service Bus 2.5 (ServiceBusTutorial) domain.

Testing the managerLoanReviewService Business Service

1. Open a command window.
2. Run the `setEnv.cmd` to set up your environment as follows:
   a. Change to the following directory:

      ```
      BEA_HOME\weblogic92\samples\servicebus\examples\src\
      ```

   b. In this directory, on the command line, run the following command:

      ```
      setEnv.cmd
      ```

   Note: If you are testing the configuration on a UNIX system, run the `setEnv.sh` script at your command prompt.

3. Change to the following directory:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
   s\jws_basic\normal
   ```

4. To test the routing behavior of the proxy service, edit the `build.xml` file **run** target as follows:
   a. Open `build.xml`.
   b. Scroll to the end of the `build.xml` file looking for the string:

      ```
      <arg line="http://$...
      ```

   c. Replace the string found at this location with the following string and save the file.

      ```
      <arg line="http://${wls.hostname}:${wls.port}/loan/gateway1 steve 555
      2500 20 4.9 notes" />
      ```

   This changes the URI to that of the proxy service running in the tutorial domain, and specifies a value of `4.9` as the interest rate.

The remainder of the input parameters specified on the `arg line` are described in Table 3-5.

Table 3-5 Input Parameters

Parameter	Description
steve	A single string representing the client name
555	The client ID
2500	The requested loan amount
20	The duration of the loan
4.9	The requested interest rate
notes	A string that can contain miscellaneous notes

5. Ensure that the business services are deployed as follows:

   a. In the command window, change to the following directory:

   ```
   cd
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
   ices\jws_basic\normal
   ```

   b. Enter `ant`, then press **Enter**. The `BUILD SUCCESSFUL` message appears as output.

6. To run the test, at the command prompt, enter:

   ```
   ant run
   ```

   This tests the routing logic of the proxy service. A response message from the Web Service is displayed at the command prompt.

   ```
   [java] Loan Application Response: APPROVED BY THE <i><b>MANAGER</b></i>
   LOAN APPLICATION REVIEW SERVICE

        [java] Rate: 4.9

   BUILD SUCCESSFUL
   ```

The response message indicates that the Manager Loan Application Processing Service processed the request because the interest rate requested was 4.9 (less than 5).

Testing the normalLoan Business Service

1. Follow the procedure in the Testing the managerLoanReviewService Business Service section. Edit the `build.xml` file in the following directory:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
   s\jws_basic\normal
   ```

   Note: Enter 5.1 as the interest rate in the `build.xml` file (to test the **normalLoan** business service), and then save the changes.

2. Deploy the business services if you have not already done so as follows:

   a. In a command window, change to the following directory:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
   ices\jws_basic\normal
   ```

   b. Enter `ant`, then press **Enter**. The `BUILD SUCCESSFUL` message appears as output.

3. To test the routing logic of the proxy service, on the command line, enter:

   ```
   ant run
   ```

   A response message from the Web Service is displayed at the command prompt.

   ```
   [java] Loan Application Response: APPROVED BY THE <i><b>NORMAL</i></b>
   LOAN APPLICATION PROCESSING SERVICE

        [java] Rate: 5.1

   BUILD SUCCESSFUL
   ```

The response message indicates that the *Normal Loan Application Processing Service* processed the request because the interest rate requested was 5.1.

CHAPTER 4

Tutorial 2. Transforming a Loan Application

Data transformation is the mapping of data from one format to another. This section describes how BEA AquaLogic Service Bus routes and transforms messages, when necessary, according to specific configurations for a proxy service.

This section includes the following topics:

- Prerequisites
- Tutorial Objectives
- Definition of the Scenario

Prerequisites

Complete "Tutorial 1. Routing a Loan Application" on page 3-1.

Tutorial Objectives

The tutorial provides the steps to create and test a routing and transformation scenario developed using the graphical environment provided in AquaLogic Service Bus Console.

Using the AquaLogic Service Bus Console you will build on what you learned in Chapter 3, "Tutorial 1. Routing a Loan Application," to:

- Configure content-based routing.
- Use an XPath expression to change the content in the message body.

Definition of the Scenario

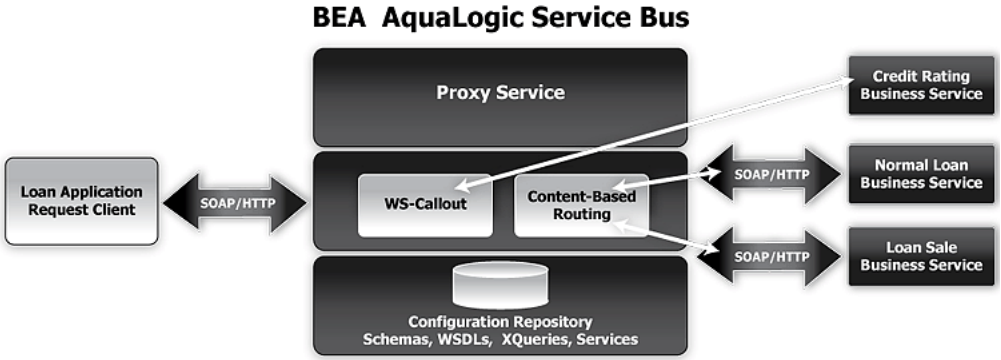
A primary mortgage company uses AquaLogic Service Bus to identify and re-route loan applications that can be sold to secondary loan companies. Loan applications with a principal request of greater than US$ 25 million are candidates for sale to a secondary loan company. When AquaLogic Service Bus receives a loan application meeting these criteria, the applicant's credit rating information is retrieved (by making a callout to a Web Service). The credit rating information is added to the loan application, then the application is forwarded to the secondary mortgage company Web Service to be processed.

Loan applications with a principal request equal to or less than US$ 25 million are routed to a different business service for processing.

The target business services respond indicating whether the loan application is approved or rejected.

Figure 4-1 illustrates where AquaLogic Service Bus fits in your enterprise to mediate the messaging between the enterprise services and the business services.

Figure 4-1 Expose a Loan Application Processing Web Service via AquaLogic Service Bus


BEA AquaLogic Service Bus
Proxy Service
Loan Application Request Client
SOAP/HTTP
WS-Callout
Content-Based Routing
Configuration Repository
Schemas, WSDLs, XQueries, Services
SOAP/HTTP
SOAP/HTTP
Credit Rating Business Service
Normal Loan Business Service
Loan Sale Business Service

Overview of the Run Time Process in AquaLogic Service Bus

A primary mortgage company receives a loan application. It is routed through the AquaLogic Service Bus proxy service, **LoanGateway2**, to determine the target business service to process the application. If the loan amount is greater than US$ 25 million, the application is routed to the **LoanSaleProcessor** business service. If the amount is less than or equal to US$ 25 million, the application is routed to the **NormalLoan** business service. When the loan amount is more than US$ 25 million, the request pipeline makes a service callout to the **CreditRatingService** and receives the credit rating of the applicant using the `$creditRating` variable. To fulfill the interface requirements of the secondary loan company service, the message body is transformed by adding the credit rating details. The transformed message (`$body`) is routed to a business service that handles applications for large loan amounts. The service returns a response similar to the following:

```
Loan Application Response: CREDIT RATING: AA: LOAN PURCHASED BY THE
<i><b>LARGE</b></i> LOANS SERVICE
```

Required Resources

You will use the Project folder, **MortgageBroker**, and the directory structure you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in Table 4-1.

Table 4-1 Routing a Loan Application Tutorial Resources

Resource Name	Description
CreditRatingService, LoanSale	These are the WSDL resources
NormalLoan, LoanSaleProcessor, CreditRatingService	These are the external business services used by AquaLogic Service Bus
LoanGateway2	This is the AquaLogic Service Bus proxy service

Steps in This Tutorial

In this tutorial, you will:

- Import additional WSDL resources
- Register new business and proxy services
- Configure the routing behavior for the proxy service. This includes:
  - Adding a Service Callout to get a credit rating
  - Changing the namespace and inserting a new **CreditRating** element in the outbound (request) message
  - Reverting the namespace change and removing the **CreditRating** element on the response message.

Complete the following steps to design and configure the proxy service and the associated resources in AquaLogic Service Bus and resolve this user case scenario:

- Step 1: Prepare the Environment
- Step 2: Create a Session and Set up a Project
- Step 3: Create the Resources
- Step 4: Configure the Routing for the LoanGateway2 Proxy Service
- Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service
- Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service
- Step 7: Add a Default Routing Case
- Step 8: Test the Loan Application Routing Configuration

Step 1: Prepare the Environment

Ensure that AquaLogic Service Bus is running in the domain that you created for the tutorial and that you have completed the steps described in Chapter 3, "Tutorial 1. Routing a Loan Application".

Step 2: Create a Session and Set up a Project

For this tutorial, you need to use the **MortgageBroker** project folder, and the directory structure that you created in Chapter 3, "Tutorial 1. Routing a Loan Application" to hold the project artifacts.

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation pane, select **Project Explorer.**

   The Project Explorer pane is opened in the navigation pane and a project page is displayed in the console.
3. In Project Explorer, expand the **MortgageBroker** project tree to expose the sub folders containing the project artifacts:
   - ProxyService
   - BusinessServices

- WSDL

Step 3: Create the Resources

In this step, you will import the WSDL resource, and create the proxy service and the business services required for the tutorial:

- Create the WSDL Resources
- Create a Proxy Service
- Create a Business Service

Create the WSDL Resources

You must create the WSDL resources (**LoanSale** and **CreditRatingService**) before creating the other resources required for this scenario since WSDLs are the base on which you create the business and proxy services.

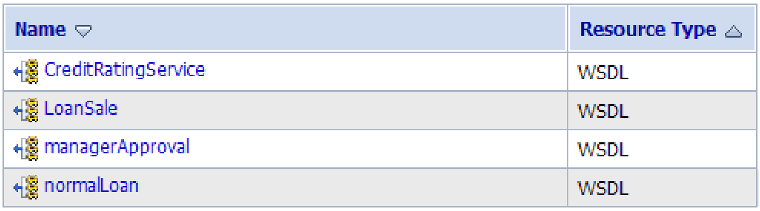
To import the appropriate WSDLs and create the WSDL resources, follow the steps described in "To Import a WSDL" on page 3-7, but for this instance, name your resources and base them on the WSDLs as described in Table 4-2.

Table 4-2 WSDL Configuration Settings

Resource Name	Location of the WSDL
LoanSale	`*BEA_HOME*\weblogic92\servicebus\examples\src\examples\webservices\jws_basic\large\LargeLoanPurchasingService.wsdl`
CreditRatingService	`*BEA_HOME*\weblogic92\servicebus\examples\src\examples\webservices\jws_basic\credit\CreditLoanApprovalService.wsdl`

When you complete this step, the `MortgageBroker/WSDL` folder contains the WSDL resources created in this tutorial and in Chapter 3, "Tutorial 1. Routing a Loan Application" as in Figure 4-2.

Figure 4-2 WSDL Resources



Name ▽	Resource Type △
CreditRatingService	WSDL
LoanSale	WSDL
managerApproval	WSDL
normalLoan	WSDL

Create a Proxy Service

In this step, you will create a proxy service. The proxy service is used to route the loan application to the appropriate business service. It also calls a look up service to obtain the credit rating of the requestor if the loan amount requested is greater than US$ 25 million.

Note: No units are assigned to the loan amount, but units can be any currency such as US Dollars.

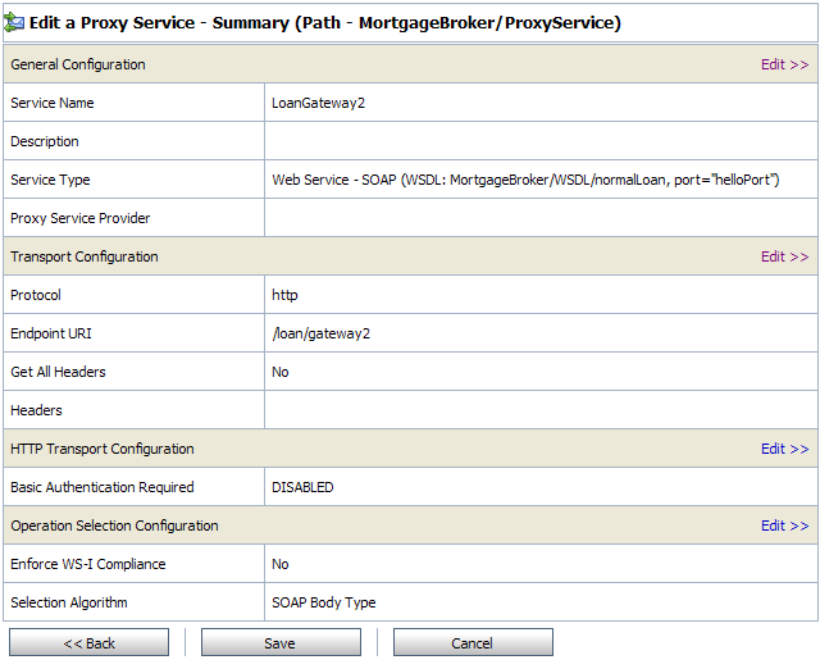
To create a new proxy service, follow the steps described in "To Create the Proxy Service" on page 3-10, but for this instance, use the proxy service name and the Endpoint URI described in Table 4-3.

Table 4-3 Proxy Service Configuration Settings

Proxy Service Name	`LoanGateway2`
Service Type	Select the **WSDL Web Service** (port or binding) option. Select the **normalLoan** WSDL in the Select a WSDL browser window. Select the **helloPort** port in the Select a WSDL Definition page.
Endpoint URI	Click Next to proceed to the Create a Proxy Service - Transport Configuration page. Enter **/loan/gateway2** as the endpoint URI.

After you complete this step, a summary of the configuration settings for the proxy service is displayed as in the following Figure 4-3. Review the configuration settings prior to registering the proxy service.

Figure 4-3 LoadGateway2 Proxy Service - Configuration Settings Summary

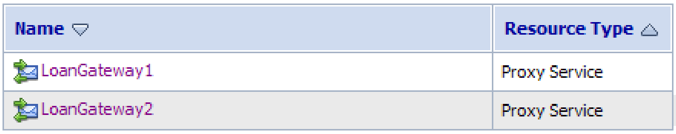


Edit a Proxy Service - Summary (Path - MortgageBroker/ProxyService)

General Configuration	Edit >>
Service Name	LoanGateway2
Description	
Service Type	Web Service - SOAP (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Proxy Service Provider	
Transport Configuration	Edit >>
Protocol	http
Endpoint URI	/loan/gateway2
Get All Headers	No
Headers	
HTTP Transport Configuration	Edit >>
Basic Authentication Required	DISABLED
Operation Selection Configuration	Edit >>
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type

When you complete this step, the `MortgageBroker/ProxyService` folder contains the proxy services that you created in this tutorial and in Chapter 3, "Tutorial 1. Routing a Loan Application" as in Figure 4-4.

Figure 4-4 Proxy Service Resources



Name ▽	Resource Type △
LoanGateway1	Proxy Service
LoanGateway2	Proxy Service

Create a Business Service

In this scenario, the proxy service is configured to route to one of several different business services, depending on the business requirements as follows:

- **CreditRatingService**—returns the customer's credit rating when a loan application meeting specified criteria is received. This is implemented using a Web Service callout (Service Callout).
- **NormalLoan**—is the secondary mortgage company's business service. This service is invoked when a loan request is made for amounts of US$ 25 million or less.
- **LoanSaleProcessor** is the secondary mortgage company's business service. This service is invoked when a loan request is made for amounts greater than US$ 25 million.

You created the Normal Loan service in Chapter 3, "Tutorial 1. Routing a Loan Application". To create the **LoanSaleProcessor** and **CreditRatingService** business services for this scenario, follow the steps described in "Create the NormalLoan Business Service" on page 3-16. However, in this case, configure your services using the names, service types, and endpoint URIs described in Table 4-4 and Table 4-5:

Table 4-4 Configuration Settings for the LoanSaleProcessor Business Service

Business Service Name	`LoanSaleProcessor`
Service Type	Select the **WSDL port** option. Select the **LoanSale** WSDL in the Select a WSDL browser window. Select the **helloPort** port in the Select a WSDL Definition page.
Endpoint URI	Ensure that the pre-populated value for the Endpoint URI in the **Existing URIs** field is `http://<host:port>/ljws_basic_ejb/LargeSimpleBean` where `<host:port>` represents the machine and port on which your AquaLogic Service Bus server is running. The host and port values must be the same as the values specified in `BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties` For more information, see "Edit the examples.properties File" on page 2-4.

Table 4-5 Configuration Settings for the CreditRatingService Business Service

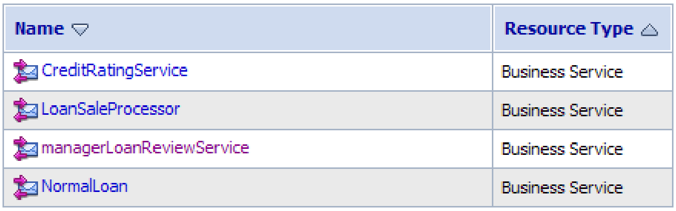
Business Service Name	`CreditRatingService`

Table 4-5 Configuration Settings for the CreditRatingService Business Service

Service Type	Select the **WSDL port** option. Select the **creditRatingService** WSDL in the Select a WSDL browser window. Select the **helloPort** port in the Select a WSDL Definition page.
Endpoint URI	Ensure that the pre-populated value for the endpoint URI in the **Existing URIs** field is `http://<host:port>/crejws_basic_ejb/CreditSimpleBean` where `<host:port>` represents the machine and port on which your AquaLogic Service Bus server is running. The host and port values must be the same as the values specified in `BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties` For more information, see "Edit the examples.properties File" on page 2-4.

When you complete this step, the **MortgageBroker/BusinessService** folder contains the business services that you created in this tutorial and in Chapter 3, "Tutorial 1. Routing a Loan Application" as in Figure 4-5.

Figure 4-5 Business Service Resources



Name ▽	Resource Type △
CreditRatingService	Business Service
LoanSaleProcessor	Business Service
managerLoanReviewService	Business Service
NormalLoan	Business Service

Summary

The resources required for this scenario are created when you complete Step 1: Prepare the Environment to Step 3: Create the Resources. You have also configured the proxy service with a base configuration. Step 4: Configure the Routing for the LoanGateway2 Proxy Service to Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service describe how to complete the configuration of the proxy service to add the routing, transformation, and Service Callout behavior for the loan application messages.

Step 4: Configure the Routing for the LoanGateway2 Proxy Service

A proxy service is implemented in AquaLogic Service Bus as a message flow, which includes request and response pipelines. This step includes the following tasks:

- Create a Routing Table
- Configure the Routing Expression

Create a Routing Table

1. In the AquaLogic Service Bus Console navigation pane, select **Resource Browser**.

   The Resource Browser pane is opened in the navigation pane and the Summary of Proxy Services page is displayed in the console.

   Note: You must be in a session to edit resources.

2. In the **Actions** column associated with the **LoanGateway2** proxy service, click the **Edit Message Flow** icon .

   The Edit Message Flow page for the proxy service **LoanGateway2** is displayed.

3. Click **LoanGateway2**, then select **Add Route Node**.

   In the configuration dialog, **RouteNode1** is added.

4. Click **RouteNode1** and select **Edit Route** from the pop-up menu.

   The Edit Stage Configuration page is displayed and contains a single link, **Add an Action**.

5. Click **Add an Action**, then select **Communucation→Routing Table** from the pop-up menu.

   The Edit Stage Configuration page changes to display the routing table configuration information.

   To configure the routing table to route messages to business services based on evaluating the **amount** element of the incoming message, you must create an XQuery expression using the XQuery Expression Editor.

Configure the Routing Expression

1. In the Routing table, click the `<Expression>` link. The XQuery Expression Editor page is displayed.

2. Accept the default selection of **XQuery Text** in the page.
3. Click **Variable Structures** to display the Variable Structures pane.
4. Select **body** from the **Select Structure** drop-down list in the Variable Structures pane. A structural representation of the **body** element is displayed in the Variable Structures pane.

   a. Click + beside the **processLoanApp** element to expand the element. The **processLoanApp** element is within the **$body - processLoanApp** (request) element.

   b. Click + to expand the **loan request** element.

      The structure of the loan request document is displayed.

   Note: At run time, the proxy service makes its routing decision based on the value in the `amount` element of the message.

   c. Click the circle icon ● associated with the **Amount** node element, drag and drop it in the XQuery Expression text box.

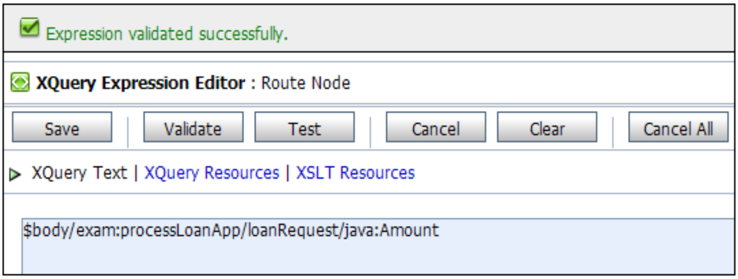
      The `$body/exam:processLoanApp/loanRequest/java:Amount` XQuery expression is written in the text box.

      Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **Amount** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XQuery expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XQuery Expression text box.

5. Click **Validate** to validate the XQuery.

   It is good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button. In this case, the expression is valid as in Figure 4-6.

Figure 4-6 Validate Routing Expression



Expression validated successfully.

XQuery Expression Editor : Route Node

```
$body/exam:processLoanApp/loanRequest/java:Amount
```

6. Click **Save**.

   The routing table is displayed on the Edit Stage Configuration page. `<Expression>` is now replaced by the expression that returns the value of the `amount` element in the message.

7. In the Operator drop-down list, select **>** and in the associated text field, enter the number **25000000** (without commas).

   The routing table now contains an expression that determines the routing behavior. If the value in the **Amount** field is greater than US$ 25 million, then messages are routed according to the routing table configuration.

8. Click the **Service** link to define the service to which you want to route when the amount is greater than US$ 25 million. The Select Service browser window is displayed.

   a. Select the **LoanSaleProcessor** business service

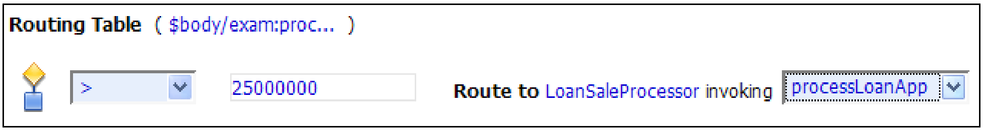
   b. Click **Submi**t.

9. In the Operation drop-down list, select the **processLoanApp** operation.

   This is the operation on the **LoanSaleProcessor** service business service that is invoked at run time if the amount of the loan requested in the loan application is greater than US$ 25 million.

You have now defined the case for routing the loan application to the **LoanSaleProcessor** business service as in Figure 4-7.

Figure 4-7 Routing the Loan Application to LoanSaleProcessor



Routing Table ($body/exam:proc...)

> 25000000 Route to LoanSaleProcessor invoking processLoanApp

When a loan application with the amount of the loan greater than US$ 25 million is identified, a Web Service callout (Service Callout) is performed to retrieve the customer's credit rating. The credit rating information is added to the loan application and the application is then forwarded to the secondary mortgage company's business service to be processed. Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service describes how to configure the proxy service to do the Service Callout and transform the message appropriately for the target service.

Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service

Execute the following steps to configure the Routing Table Request actions:

- Configure the Service Callout Input Parameter
- Add a Service Callout Action
- Configure the Message Transformation

Configure the Service Callout Input Parameter

A Service Callout is used to send the loan application to the **CreditRatingService** business service, which returns the credit rating of an applicant. Before configuring the Service Callout action, you must assign an input parameter for the Service Callout action as follows:

1. Open the Edit Stage Configuration page for the routing table that you created in “Step 4: Configure the Routing for the LoanGateway2 Proxy Service” on page 4-10 in the AquaLogic Service Bus Console.

2. In the Request Actions pane of the Routing Table, click **Add an Action**, then select **Message Processing→Assign**.

   The **Assign** action is added to the Request Action pipeline in the Routing Table.

3. Click the `<Expression>` link. The XQuery Expression Editor page is displayed.

   a. Accept the default selection of **XQuery Text** in the page.

   a. Select **body** from the Variables Structures drop-down list in the Variables Structures pane, then expand the element list and drag the **loanRequest** element and drop it into the XQuery expression text box. The XQuery expression is written in the text box:

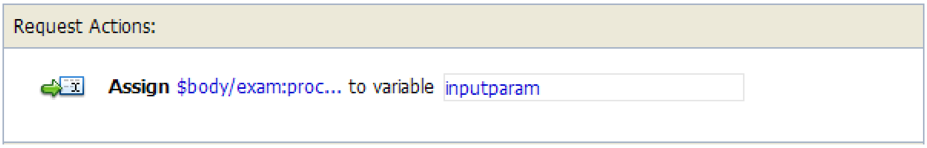
   ```
   $body/exam:processLoanApp/loanRequest
   ```

b. Click **Validate** to validate the XQuery, then **Save**.

The routing table is displayed on the Edit Stage Configuration page. `<Expression>` is now replaced by the expression that returns the content of the message to be sent to the Service Callout service.

4. Enter `inputparam` in the variable text box as in Figure 4-8.

Figure 4-8 Configure Service Callout Input Parameter



Request Actions:

Assign $body/exam:proc... to variable inputparam

This step completes the assignment of the value returned by the XQuery expression (`$body/exam:processLoanApp/loanRequest`) to the `inputparam` variable.

Add a Service Callout Action

This section describes how you can configure the Service Callout action to send the loan application to the **CreditRatingService** business service, which returns the credit rating of an applicant.

1. In the Request Actions pane of the Routing Table, click ⇨, then select **Add an Action→Communication→Service Callout**.

   The Service Callout action is added to the Request Action pipeline in the Routing Table.

2. Click the Service Callout **Service** link. The Select Service browser window is displayed.

   a. In the Select Service window, select **CreditRatingService**.

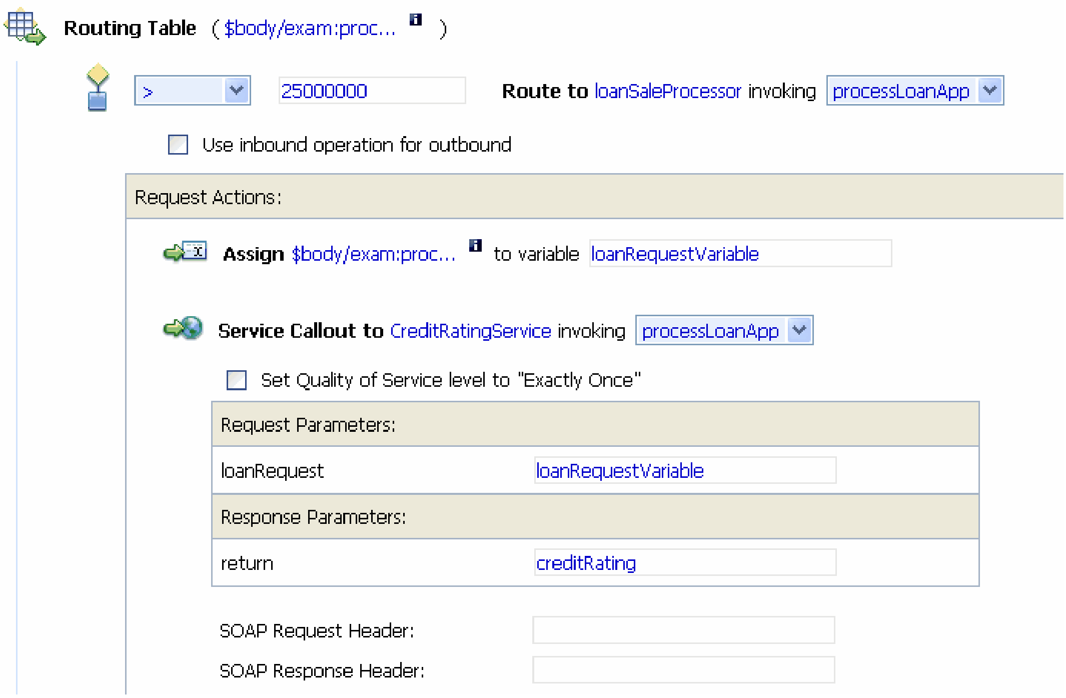
   b. Click **Submit**.

3. From the Operation drop-down list, select **processLoanApp**. The fields that allow you to configure the request and response parameters for the Service Callout are displayed:

   a. In the **Service Callout Request Parameters** section, set the Loan request parameter to **inputparam** (the message context variable you defined in the preceding section).

   b. In the **Service Callout Response Parameters** section, set the return parameter name to **creditRating** as in Figure 4-9.

Figure 4-9 Configure Service Callout Action



Routing Table ($body/exam:proc...)

> 25000000 Route to loanSaleProcessor invoking processLoanApp

□ Use inbound operation for outbound

Request Actions:

Assign $body/exam:proc... to variable loanRequestVariable

Service Callout to CreditRatingService invoking processLoanApp

□ Set Quality of Service level to "Exactly Once"

Request Parameters:	
loanRequest	loanRequestVariable
Response Parameters:	
return	creditRating

SOAP Request Header:

SOAP Response Header:

This step completes the configuration of the Service Callout to the **creditRatingService** Web Service. The credit rating returned by the Web Service is assigned to the `creditRating` context variable.

Configure the Message Transformation

In this section, you will configure the message transformation to match the public contract (interface requirements) of the **loanSaleProcessor** service. You will configure the proxy service to:

- Rename the namespace for the message to the namespace required by the target service using XPath expressions as described in “Rename the Namespace” on page 4-16.
- Insert an new element into the message by adding an element action in the configuration as described in “Insert a New Element in the Message” on page 4-16.

Rename the Namespace

1. Click the Service Callout icon , then select **Add an Action→Message Processing→Rename**.

   The Rename action configuration fields are added to the Request Actions pane.

2. In the **Rename** statement, click the **<XPath>** link. The XPath Expression Editor is displayed.

   a. In the XPath Expression text box, enter the following XPath expression:

   ```
   .//java:*
   ```

   This expression identifies every instance of a namespace with a prefix of `java`.

   b. Click **Save** to save the XPath expression and return to the Edit Stage Configuration page.

3. In the variable text box, enter `body`.

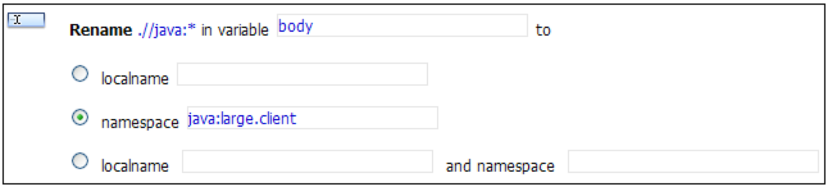
   The **body** context variable contains the body of the message. You have completed creating a condition in which the XPath expression finds all the namespaces with the `java` prefix in the **body** context variable. The next step specifies the namespace with which to replace the namespaces identified by the XPath expression.

4. Select the namespace option and enter the fully qualified namespace to be substituted in the message. The new namespace is `java:large.client`.

   Note: The namespace that you are replacing in this case is `java:normal.client`.

The Rename action should now appear as in Figure 4-10.

Figure 4-10 Rename Namespace



Rename ./java:* in variable body to

○ localname

● namespace java:large.client

○ localname and namespace

Insert a New Element in the Message

1. Click the Rename icon , then select **Add an Action→Message Processing→Insert**.

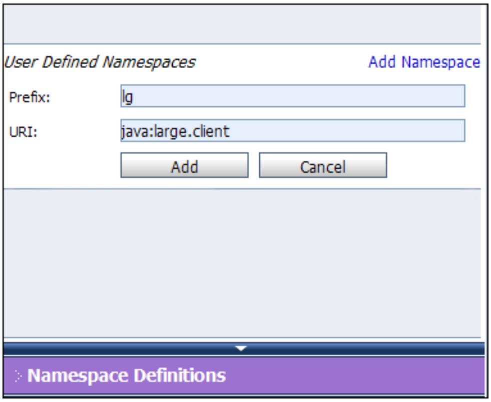
   The Insert action configuration fields are added to the Request Actions pane.

2. In the Insert action, click `Expression`.

The XQuery Expression Editor page is displayed. You will add a new namespace on this page in the subsequent steps.

3. In the **Namespace Definitions** palette, locate to the **User Defined Namespaces** area.

Figure 4-11 User Defined Namespace



User Defined Namespaces

Prefix: lg

URI: java:large.client

a. Click **Add Namespace**.

b. In the Prefix field, enter `lg`.

c. In the URI field, enter `java:large.client`

d. Click **Add**.

The new namespace is displayed in the **User Defined Namespaces** area in the **Namespace Definitions** palette as in Figure 4-11.

4. In the XQuery Expression text box, enter the following expression to specify how the new element is constructed:

```
<lg:CreditRating>{data($creditRating)}</lg:CreditRating>
```

Where:

– `lg:` is namespace of the element that you add to the message must be associated with the `java:large.client` namespace.

– `{}` indicates to the XQuery engine that the content between the `{}` is not XML and must be interpreted.

At run time, the `$creditRating` variable is assigned a credit rating value by the credit rating service (to which we configured a Service Callout action in "Add a Service Callout Action" on page 4-14).

5. Click **Validate**, then **Save**.

   Note: It is good practice is to validate the expression before you submit it.

6. In the Insert statement, select **after** from the drop-down list.

7. Click the **<XPath>** link. The XPath Expression Editor page is displayed.

   a. Click **Variable Structures**. The Variable Structures pane is displayed.

   b. In the **Select Structure** drop-down list in the Variables Structures pane, select **body**.

   c. Expand the element list and drag the **Notes** element and drop it into the XPath Expression text box.

   The following XPath expression is written to the text box

   ```
   ./exam:processLoanApp/loanRequest/java:Notes
   ```

   Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **Notes** element in the Variable Structures pane. The XPath expression is displayed in the **Property Inspector** palette. Place the cursor in the XPath expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XPath Expression text box.

8. In this statement, replace the namespace **java** with the namespace **lg:**

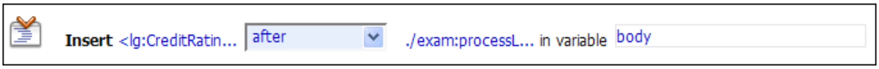
   ```
   ./exam:processLoanApp/loanRequest/lg:Notes
   ```

9. Click **Validate**, then **Save** to return to the Edit Stage Configuration page.

10. Enter `body` in the Variable text field (the last field in the expression). This is the context variable into which the new **<CreditRating>** element is inserted at run time.

The Insert action should now look like Figure 4-12.

Figure 4-12 Insert New Element into Message

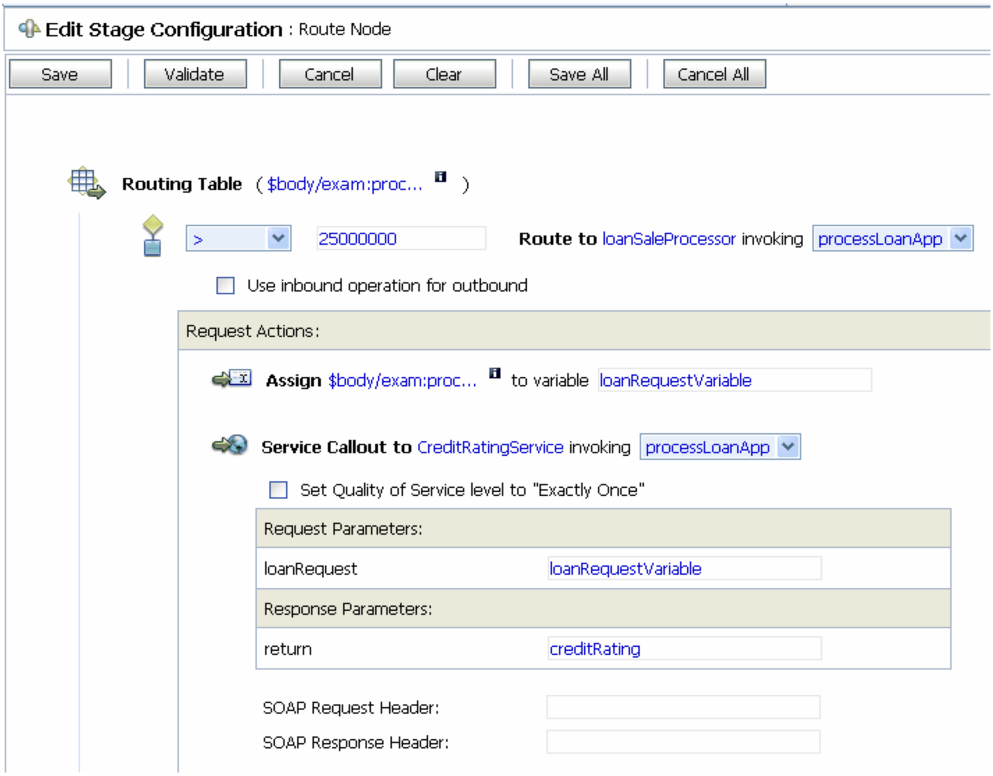


Insert <lg:CreditRatin... after ./exam:processL... in variable body

This step completes the configuration of the outbound message. You added a credit rating element to the message and changed the namespace so that the message complies with the public contract (interface) of the target service.

The configuration of the outbound message should look like Figure 4-13.

Figure 4-13 Configuration of Outbound Message



Edit Stage Configuration : Route Node

Routing Table ($body/exam:proc...)

> 25000000 **Route to** loanSaleProcessor invoking processLoanApp

□ Use inbound operation for outbound

Request Actions:

Assign $body/exam:proc... to variable loanRequestVariable

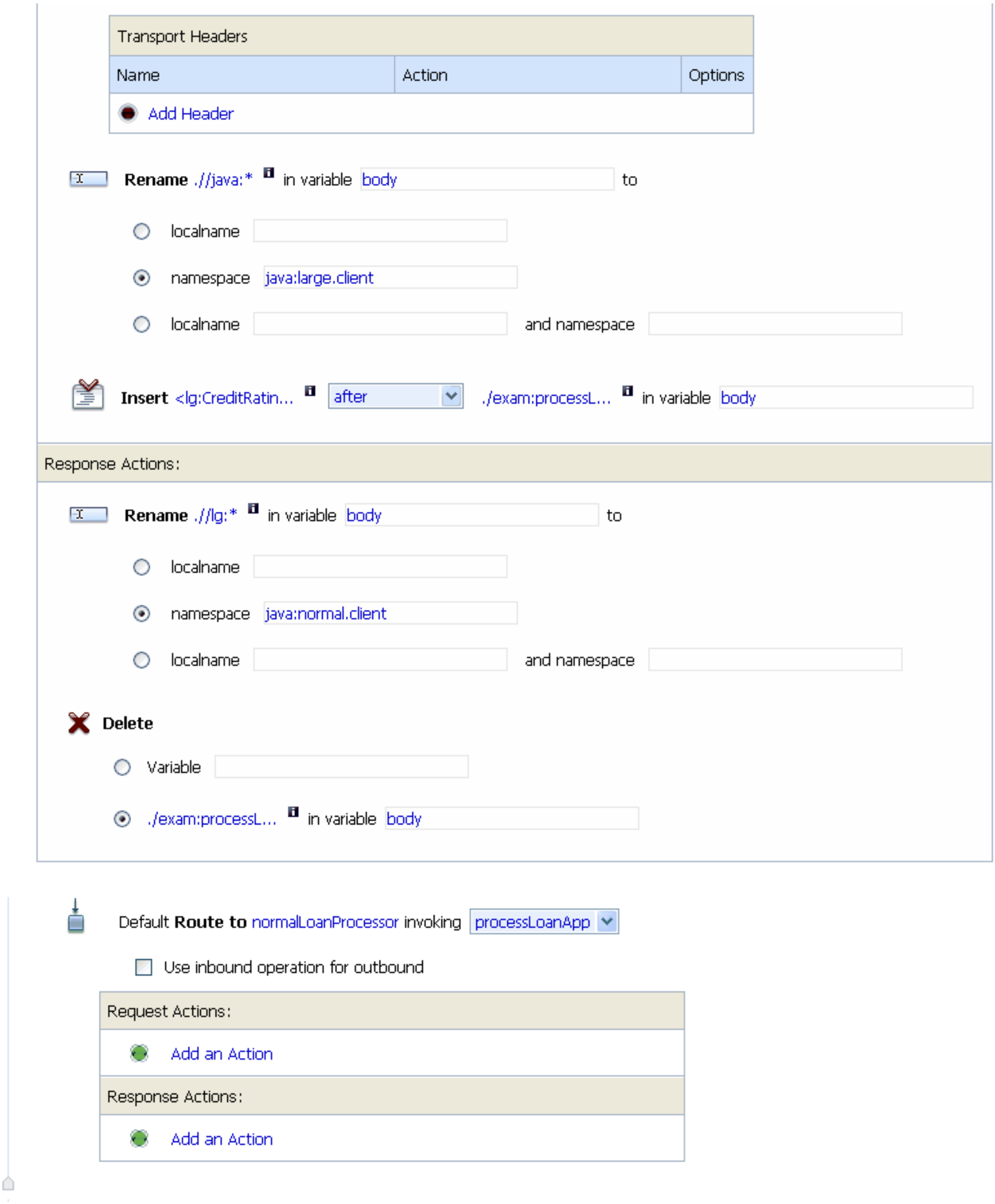
Service Callout to CreditRatingService invoking processLoanApp

□ Set Quality of Service level to "Exactly Once"

Request Parameters:	
loanRequest	loanRequestVariable
Response Parameters:	
return	creditRating

SOAP Request Header:

SOAP Response Header:



Transport Headers		
Name	Action	Options
Add Header		

Rename .//java:* in variable body to

○ localname

● namespace java:large.client

○ localname and namespace

Insert <lg:CreditRatin... after ./exam:processL... in variable body

Response Actions:

Rename .//lg:* in variable body to

○ localname

● namespace java:normal.client

○ localname and namespace

Delete

○ Variable

● ./exam:processL... in variable body

Default **Route to** normalLoanProcessor invoking processLoanApp

☐ Use inbound operation for outbound

Request Actions:

Add an Action

Response Actions:

Add an Action

The next step describes how you can configure the response actions for the **LoanGateway2** proxy service.

Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service

This step describes how you can configure the response actions in the Routing Table so that the message that is returned by the proxy service to the client complies with the client's public contract (WSDL). You can configure the proxy service to:

- Remove the CreditRating element from the response message as described in "Delete an Element" on page 4-21.
- Revert the namespace to the namespace in the original message: `java:normal.client` as described in "Add a Rename Action" on page 4-22.

Delete an Element

1. Ensure that the Edit Stage Configuration page for the routing table that you created and configured in the preceding steps is open in the AquaLogic Service Bus Console.
2. In the Routing Table **Response Actions** pane, select **Add an Action→Message Processing→Delete**.

   The **Delete** action configuration fields are added to the Response Actions in the Routing Table.
3. Select the **< XPath> in variable** option, then click **<XPath>**. The XPath Expression Editor page is displayed.
   a. In the XPath Expression text box, enter the following expression:

      ```
      ./exam:<processLoanAppResponse>/return/lg:CreditRating
      ```

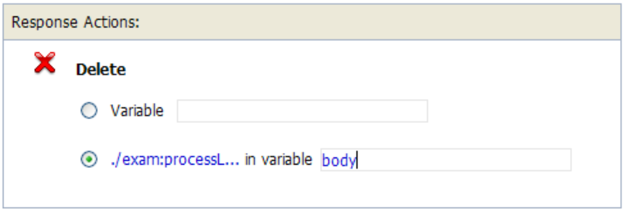
      where `<processLoanAppResponse>` is the WSDL operation that was invoked with the string **Response** appended to it, and **return** is the WSDL part name.

   b. Click **Validate**, then **Save**. You are returned to the Edit Stage Configuration page.
4. Enter `body` in the variable text field in the Delete action.

   The configuration for the Delete action is therefore:

   Delete `./exam:processLoanAppResponse/return/lg:CreditRating` in variable `body` as in Figure 4-14.

Figure 4-14 Delete Element



Response Actions:

✖ **Delete**

○ Variable

● ./exam:processL... in variable body

You have specified that AquaLogic Service Bus removes the credit rating element from the response message as the message is processed in the response pipeline.

Add a Rename Action

This section describes how you can rename the namespace to the namespace that the client requires, that is `java:normal.client`.

Note: Recall that you configured the request message to the **loanSaleProcessor** business service to change the namespace to that required by that service. (See "Rename the Namespace" on page 4-16.

1. Click the **Delete** icon ✖, then **Add an Action→Message Processing→Rename**.

   The Rename action configuration fields are added to the Response Actions pane.

2. In the **Rename** statement, click **<XPath>**. The XPath Expression Editor is displayed.

   a. In the XPath Expression text box, enter the following expression:

   ```
   .//lg:*
   ```

   This XPath expression finds every namespace with a prefix of `lg:`

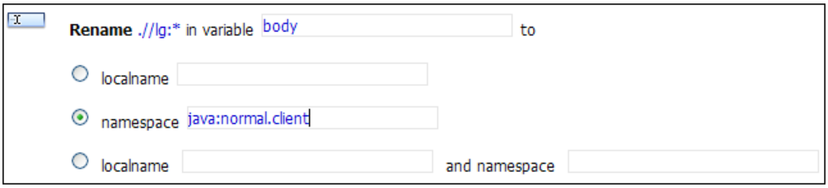
   b. Click **Save**. The Edit Stage Configuration page is displayed.

3. In the Rename action's **variable** text field enter `body`.

4. Select the **namespace** option, and in the text field associated with the namespace, enter `java:normal.client`.

   When these steps are completed, you have specified that AquaLogic Service Bus replaces all namespaces with the prefix `lg:` in the response message with `java:normal.client` as in Figure 4-15.

Figure 4-15 Add Rename Action



Rename .//lg:* in variable body to

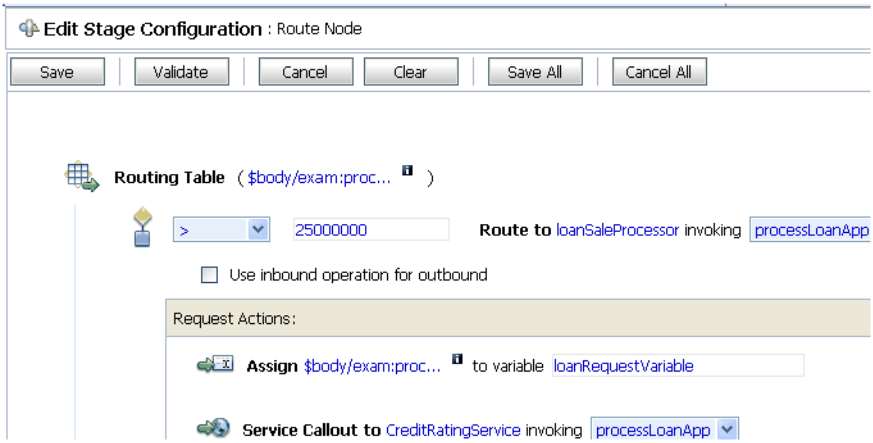
○ localname

● namespace java:normal.client

○ localname and namespace

After these steps are complete, the configuration of the response action of the proxy service appears in Figure 4-16.

Figure 4-16 Configuration of Response Actions of Proxy Service



Edit Stage Configuration : Route Node

Routing Table ($body/exam:proc...)

> 25000000 **Route to** loanSaleProcessor invoking processLoanApp

□ Use inbound operation for outbound

Request Actions:

Assign $body/exam:proc... to variable loanRequestVariable

Service Callout to CreditRatingService invoking processLoanApp

The last step in the configuration of the routing table for the **LoanGateway2** proxy service is to add a default routing case (an else condition) to the case expression.

Step 7: Add a Default Routing Case

This step describes how you can configure a default routing case for the routing table of the LoanGateway2 proxy service. At run time, if none of the conditions configured in the Routing Table are met, AquaLogic Service Bus routes messages according to the configuration of this default case.

1. Click the routing table Case icon , then select **Insert Default Case**.

   The default case is added after the routing table on the Edit Stage Configuration page.

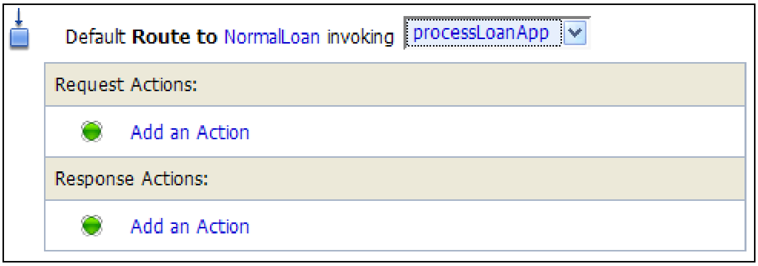
2. Click <**Service**> in the default case to display the Select Services window.

   a. Select the **NormalLoan** business service.

   b. Click **Submit**.

3. In the default case configuration operation drop-down box, select the operation **processLoanApp** as in Figure 4-17.

Figure 4-17 Add Default Routing Case



Default **Route to** NormalLoan invoking processLoanApp

Request Actions:

Add an Action

Response Actions:

Add an Action

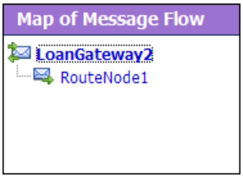
4. Click **Save** to save the route node configuration and return to the Edit Message Flow page for the **LoanGateway2** proxy service.

5. Click **Save** on each configuration page until you return to the Summary of Proxy Services page.

6. Click **Activate**, and then **Submit** to save the session configuration information.

You can view the map of the message flow of the proxy service in Figure 4-18.

Figure 4-18 Map of Message Flow of the LoanGateway2 Proxy Service



Map of Message Flow

LoanGateway2

RouteNode1

Step 8: Test the Loan Application Routing Configuration

Now that you have configured AquaLogic Service Bus to work with the client and the target business services, you can test your configuration. The routing behavior of the proxy service in this scenario is based on the value of the loan amount in the loan application message. Changing the value of the loan amount changes the routing behavior.

- A value equal to or less than US$ 25 million causes the message to be routed to the **normalLoan** business service.
- A value greater than US$ 25 million causes the message to be routed to the **LoanSaleProcessor** business service.

This section describes how you can test the proxy service for both cases. The message returned in the command window after running a test indicates the success or failure of the test and also identifies the business service that processed the request.

This section includes the following test scenarios:

- Test the LoanSaleProcessor Business Service
- Test the NormalLoan Business Service

Test the LoanSaleProcessor Business Service

1. Open a command window.
2. If you have not run the `setEnv.cmd` to set up your environment, you must do so now.
   a. Change to the following directory:
   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\
   ```
   b. In this directory, on the command line, run the following command:
   ```
   setEnv.cmd
   ```
   Note: If you are testing the configuration on a UNIX system, run the `setEnv.sh` script at your command prompt.
3. Change to the following directory:
   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
   s\jws_basic\normal
   ```
4. To change the routing behavior of AquaLogic Service Bus, you must edit the `build.xml` file run target in this directory:
   a. Open `build.xml`.
   b. Scroll to the end of the `build.xml` file looking for the string:
   ```
   <arg line="http://$...
   ```

c. Replace the string found at this location with the following string and save the changes.

```
<arg line="http://${wls.hostname}:${wls.port}/loan/gateway2 steve 555
25000001 20 5.1 notes" />
```

This changes the URI of the proxy service to that of the deployed Web Service (**gateway2**) running in the tutorial domain, and specifies US$ **25000001** as the loan amount. For more information about the input parameters for the test, see "Step 7: Test Your Loan Application Routing Configuration" on page 3-27.

5. If you have not already done so, deploy the business services as follows:

a. In a command window, change to the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
ices\jws_basic\normal
```

a. Enter `ant`, then press **Enter**.

6. To test the build, on the command prompt, enter:

```
ant run
```

This tests the routing logic of the proxy service. The following message from the Web Service is displayed at the command prompt:

```
[java] Loan Application Response: CREDIT RATING: AA: LOAN PURCHASED BY
THE <i><b>LARGE</b></i> LOANS SERVICE

      [java] Rate: 5.1

BUILD SUCCESSFUL
```

The response message indicates that the *Large Loan Service* processed the request because the amount of the loan requested was greater than US$ 25 million.

Test the NormalLoan Business Service

1. Follow the procedure "Test the LoanSaleProcessor Business Service" on page 4-25 to edit the `build.xml` file in the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
s\jws_basic\normal
```

In this case (to test the **normalLoan** business service), in the `build.xml` file, enter a value of `25000000` for the loan amount (see "Step 7: Test Your Loan Application Routing Configuration" on page 3-27), then save the changes.

2. If you have not already done so, deploy the business services as follows:

a. In a Command window, change to the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
ices\jws_basic\normal
```

a. Enter `ant`, then press **Enter**.

3. To test the routing logic of the proxy service, at the command prompt, enter:

```
ant run
```

A response message from the Web Service is displayed at the command prompt.

```
[java] Loan Application Response: APPROVED BY THE <i><b>NORMAL</b></i>
LOAN APPLICATION PROCESSING SERVICE

     [java] Rate: 5.1

BUILD SUCCESSFUL
```

The response message indicates that the Normal Loan Application Processing Service processed the request because the amount of the loan requested was US$ 25 million.

Best Practices

In this tutorial, you configured the transformation and replaced the namespaces in the Route node. Alternatively, it is possible to configure the same logic in the request pipeline in a pipeline pair. The work you must do to accomplish the transformation and namespace replacement is the same regardless of the model that you choose.

In the case of the tutorial scenario, it is easier to add the actions to the request and response actions of the existing route node rather than create a separate pipeline pair, add a stage, and then add the transformation actions to that stage.

The following scenario describes a situation in which the choice of the implementation is important:
A route node routes messages to five different services. Each service requires the same transformation. In this scenario, it is a good practice to add a single transformation to a request pipeline rather than configure five identical sets of transformations for each routing configuration of every route node. The run time execution cost is the same, but the cost of maintaining, configuring, and understanding the latter implementation is higher and requires greater effort.

Tutorial 3. Validating a Loan Application

This section describes how a BEA AquaLogic Service Bus proxy service can be configured to validate a message passing from a client to a business service. Validation can occur at any stage in the flow, though it usually happens as early as possible to prevent unnecessary processing of the message if an invalid result occurs. Validation can occur in several ways.

You can:

- Validate elements selected by an **XPath** expression against a top level XML schema element or WSDL resource. For more information about the Validate action, see “Adding an Action” in Proxy Services: Actions in the *AquaLogic Service Bus Console Online Help*.
- Validate the message and report errors. For more information, see “Adding an Action” in Proxy Services: Error Handlers in *AquaLogic Service Bus Console Online Help*.

AquaLogic Service Bus does not automatically validate the message sent or received against the service interface definition, whether this is a WSDL definition or a messaging interface definition. However, you can configure a Validate action and use XQuery conditional expressions to perform explicit validate checks in the message flow. This reduces the possibility of run time errors as a result of invalid configurations.

This section includes the following topics:

- Prerequisites
- Tutorial Objectives
- Definition of the Scenario
- Steps in This Tutorial

Prerequisites

Complete "Tutorial 1. Routing a Loan Application" on page 3-1 and "Tutorial 2. Transforming a Loan Application" on page 4-1.

Tutorial Objectives

The objective of this tutorial is to provide steps to configure and test BEA AquaLogic Service Bus and demonstrate how message validation and error handling is managed using the graphical environment in the AquaLogic Service Bus Console. It includes the following:

- Demonstrates a common message validation and error handling pattern using an AquaLogic Service Bus proxy service.
- Validates a request pipeline message against a schema type definition in a WSDL.

Features Highlighted

This tutorial exposes you to the following features of AquaLogic Service Bus:

- Configure a request response pipeline
- Use the validate action to raise a validation exception
- Illustrate how a stage error handler can trap an error and return a customized error message

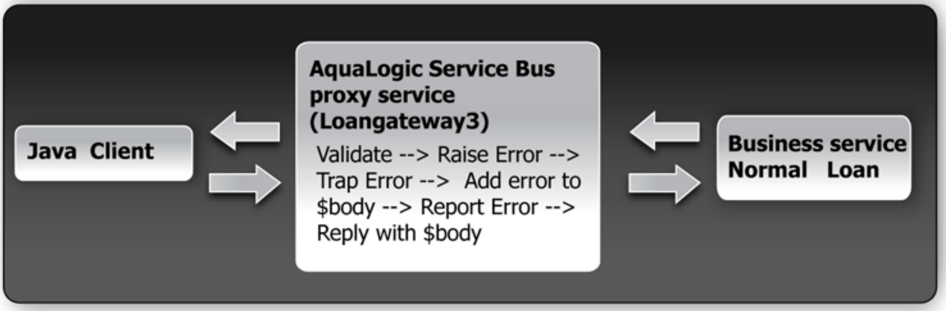
Definition of the Scenario

A primary mortgage company routes every loan application through a BEA AquaLogic Service Bus proxy service which validates the message. If the application is:

- Incomplete, it is written to an error directory and an error message is returned to the client.
- Complete, it is routed to a business service for review.
- Approved, the service returns a message indicating whether the loan is accepted or rejected.

Figure 5-1 summarizes the logical architecture to support this scenario. It illustrates where AquaLogic Service Bus fits in the enterprise to mediate the messaging between the client and the business services.

Figure 5-1 Validating a Loan Application using AquaLogic Service Bus


Java Client
AquaLogic Service Bus proxy service (Loangateway3)
Validate --> Raise Error --> Trap Error --> Add error to $body --> Report Error --> Reply with $body
Business service Normal Loan

Overview of the Process in AquaLogic Service Bus

In this scenario, you can configure AquaLogic Service Bus to validate an incoming message in the message flow against a schema definition in a WSDL resource. Every field is checked for valid content. All elements must be populated for the application to be valid. However, in this case, when the value of the loan duration field is not an integer, the validation action raises an error. The context variable, **$fault**, is populated with the error. The error is trapped by a stage error handler which replaces the **body** context variable `<Notes>` field value with the text of the **$fault** `<reason>` element. A **Report** action then saves the error code and, for context, the message body for later viewing and searching in the console. The **Reply** action returns the body context variable to the client.

In this scenario, the message is:

- Returned by the response pipeline to the client along with the validation error message

or

- Forwarded to a business service, **NormalLoanApprovalService**, that returns a reply: `The loan application was accepted.`

Required Resources

Use the Project folder, `MortgageBroker`, and the directory structure that you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in Table 5-1.

Table 5-1 Routing a Loan Application Tutorial Resources

Resource Name	Description
normalLoan	The WSDL resource
NormalLoan Web Service	The external business service used by AquaLogic Service Bus
LoanGateway3	The AquaLogic Service Bus proxy service

Steps in This Tutorial

In this tutorial, BEA AquaLogic Service Bus is used to route a loan application within a Mortgage company to a target Web Service. The loan application has to be completed correctly for the application to be processed. If there is an error in the application, AquaLogic Service Bus raises a validation exception, and returns an error message to the client.

You will create a proxy service, **LoanGateway3**, and configure it to route a message to the **normalLoan** business service. You will configure the behavior of the proxy service, which includes:

- Creating a proxy service and edit the request pipeline to include a validate action
- Using the context variable, **$fault**, to hold the error
- Trapping the error using a stage error handler to write the context variable element `$fault<reason>` to the value of the body context variable `<Notes>` field.

Complete the following steps to design and configure the proxy service and the associated resources in AquaLogic Service Bus to resolve this user case scenario:

- Step 1: Prepare Your Environment
- Step 2: Create a Session and Set up a Project
- Step 3: Create the Resources
- Step 4: Configure the LoanGateway3 Proxy Service

- Step 5: Add a Default Routing Case
- Step 6: Test the Loan Application Routing Configuration
- Step 7: Using Reporting and Monitoring

Step 1: Prepare Your Environment

Ensure that AquaLogic Service Bus is running in the domain you created for the tutorial and that you have completed the steps described in Chapter 3, "Tutorial 1. Routing a Loan Application".

Step 2: Create a Session and Set up a Project

For this tutorial, use the project folder, `MortgageBroker`, and the directory structure you created in Chapter 3, "Tutorial 1. Routing a Loan Application" to hold the project artifacts.

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation pane, select **Project Explorer.**

   The Project Explorer pane is opened in the navigation pane and a project page is displayed in the console.
3. In **Project Explorer** expand the **MortgageBroker** project tree to expose the sub folders containing the project artifacts:
   - ProxyService
   - BusinessServices
   - WSDL

Step 3: Create the Resources

In this step, you will create the proxy service, **LoanGateway3**. You will use the existing **normalLoan** service WSDL resource to validate the content of the message. If the loan application is valid, that is, if it meets the expected criteria, and is routed to the existing **normalLoan** business service. The WSDL resource and business service were created in "Create the normalLoan Service WSDL Resource" section in Chapter 3, "Tutorial 1. Routing a Loan Application".

Create a Proxy Service

In this step, you create a proxy service that is used to route the loan application to the appropriate business service. Follow the steps described in "To Create the Proxy Service" on page 3-10, but for this instance use the proxy service name and the Endpoint URI described in Table 5-2.

Table 5-2 Proxy Service Configuration Settings

Proxy Service Name	`LoanGateway3`
Service Type	Select `normalLoan` WSDL
Endpoint URI	`/loan/gateway3`

You should also review the configuration settings prior to registering the proxy service.

When you complete this step, the `MortgageBroker/ProxyService` folder contains the proxy services you created in this tutorial and in Chapter 3, "Tutorial 1. Routing a Loan Application":

- LoanGateway1
- LoanGateway2
- LoanGateway3

Step 4: Configure the LoanGateway3 Proxy Service

The proxy service is implemented in AquaLogic Service Bus as a message flow, which includes request and response pipelines. AquaLogic Service Bus message flows define the implementation of proxy services. The message flows can include zero or more pipeline pairs:

- Request and response pipelines for the proxy service (or for the operations on the service)
- Error handler pipelines that can be defined for stages, pipelines, and proxy services.

Pipelines can include one or more stages, which in turn include actions. To configure the behavior of the **LoanGateway3** proxy service, for the message flow, you must complete the following:

- Create a Pipeline Pair
- Configure the Request Actions for the LoanGateway3 Proxy Service

Create a Pipeline Pair

1. In the AquaLogic Service Bus Console navigation pane, select **Project Explorer**.

   The Project Explorer pane is opened in the navigation pane.

   Note: You must be in a session to edit resources. If you have not already done so, begin a session (click **Create** in the Change Center) so that you can configure the proxy service and edit the message flow.

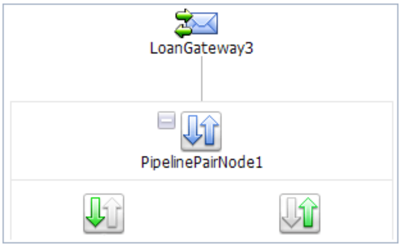
2. In the **Actions** column associated with the **LoanGateway3** proxy service, click the **Edit Message Flow** icon .

   The **Edit Message Flow** page for the proxy service **LoanGateway3** is displayed.

3. Click the **LoanGateway3** node, then select **Add Pipeline Pair**.

The **PipelinePairNode1** is created and placed in the message flow. Request and response pipelines are displayed for this node as in Figure 5-2.

Figure 5-2 Add Pipeline Pair



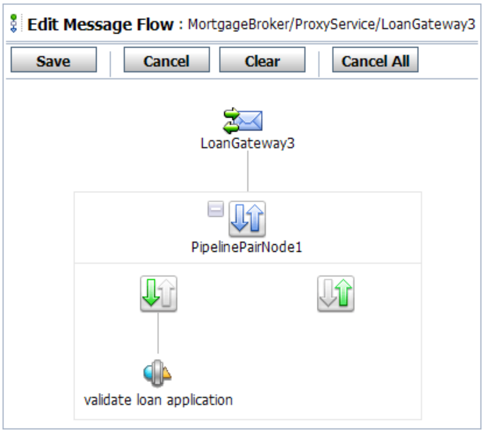
Configure the Request Actions for the LoanGateway3 Proxy Service

You must now configure the request actions for the proxy service.

1. Click the request pipeline , then select **Add Stage**.
2. Accept the default name, **Stage 1**.
3. Click **Stage1**, then select **Edit→Name and Description.**
4. Enter **validate loan application** and click **Save.**

The page is displayed as in Figure 5-3.

Figure 5-3 Add Stage to Request Pipeline


Edit Message Flow : MortgageBroker/ProxyService/LoanGateway3
Save
Cancel
Clear
Cancel All
LoanGateway3
PipelinePairNode1
validate loan application

Configure an Action to Perform Message Validation

1. To edit the stage, click **validate loan application**, then select **Edit→Stage.** The Edit Stage Configuration page is displayed.

2. Click the **Add an Action** link and select **Message Processing→Validate** from the drop-down list. The Validate action is added to the stage.

3. In the Validate statement, click the **XPath** link to edit the XPath expression. The XPath Expression Editor page is displayed.

4. Click **Variable Structures**. The Variable Structures pane is displayed.

5. In the Variables Structure pane, from the **Select Structure** drop-down list, select **body**.

   A structural representation of the **body** element is displayed in the Variables Structure pane.

   a. Click **+** beside the **processLoanApp** element to expand the element.

      The **processLoanApp** element is within the **$body - processLoanApp (request)** element.

   b. Click + to expand the **loanRequest** element.

      A graphical representation of the structure of the loan request document is displayed.

c. Select the circle icon ● associated with the **loanRequest** element, drag and drop it in the XQuery Expression text box. The XQuery expression is written in the text box.

```
./exam:processLoanApp/loanRequest
```

Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **loanRequest** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the **XQuery** expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XQuery Expression text box.

6. Click **Validate** to validate the XQuery.

   It is good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button. In this case the expression is valid.

7. Click **Save**.

   The Edit Stage Configuration page displays `<XPath>` which is replaced by the XQuery expression.

8. In the **Variable** text field, enter **body**.

9. Select **resource→WSDL** from the list. The Select a WSDL browser window is displayed.

   You must validate the message against the resource as the proxy service is based on the WSDL resource that you originally created.

10. In the Select a WSDL window, select the **normalLoan** WSDL.

    The **Select WSDL Definitions** pane is populated with the content categories of the WSDL.

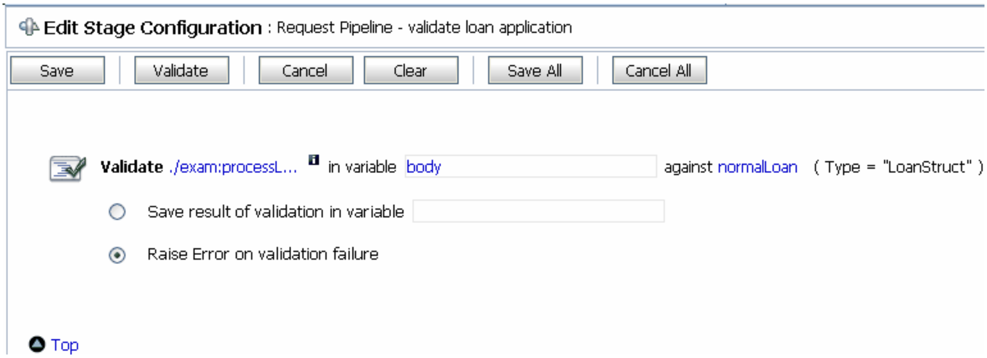
    a. In the Select WSDL Definitions pane, from the Types category, select the WSDL type for this WSDL as **loanStruct**.

    b. Click **Submit** to complete the selection.

11. Select the **Raise Error on validation failure** option. The validate action appears as in Figure 5-4.

12. Click **Save** to save the configuration.

Figure 5-4 Configure Action for Message Validation


Edit Stage Configuration : Request Pipeline - validate loan application
Save
Validate
Cancel
Clear
Save All
Cancel All
Top

Validate ./exam:processL... in variable body against normalLoan (Type = "LoanStruct")

○ Save result of validation in variable

● Raise Error on validation failure

Note: The validation of a message must occur as early as possible in a message flow. This prevents the unnecessary processing of a message in the case where the content of the message is invalid. When an exception is raised, further processing is stopped and a response is immediately returned to the client.

Messages can be validated at the route node or using a separate Validate action (as in this case). When you create a Validation stage, the processing of the message is more modular and logically breaks down the actions in the message flow. If multiple validations are preformed on a message, the validate action is logically expanded.

Add a Stage Error Handler

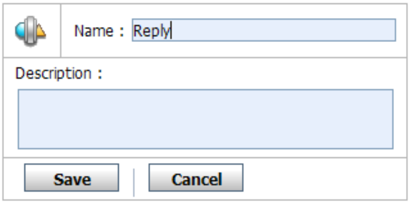
The Stage Error Handler has to be configured as follows:

1. Configure the Stage Error Handler to check the error code using an **If...Then** action and **Else** actions (Add an If...Then action to the Stage Error Handler).
2. Add a Replace action within the If portion of the **If...Then...** action of the stage. Configure this action to replace the **Notes** element value with the **fault** context variable **reason** text. The **fault** description is returned to the client (Add a Replace Action to the Stage Error Handler).
3. Add a Report action within the If portion of the **If...Then...** action of the stage (Add a Report Action to Report the Error).
4. Add a Replace action within the **Then...** section of the stage (same as step 2).
5. Add a Log action within the **Then...** action of the stage (Add a Log Action Within the Else Action).
6. Add a **Reply with Failure** for both conditions of the stage (Add a Reply Action to Reply with a Failure Code).

The steps to add the Stage Handler are as follows:

1. Click **validate loan application**, then select **Add→Stage Error Handler**. The Edit Error Handler page is displayed.
2. Click **Error handler**, then select **Add Stage**. The **stage1** node is displayed.
3. Click **stage1** and select **Edit→Name and Description** as in Figure 5-5.
4. Enter the name **Reply** for the stage, then click **Save**. The reply stage is displayed on the page.

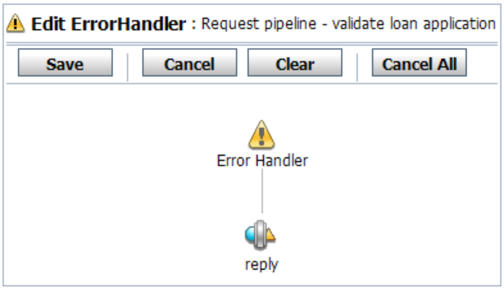
Figure 5-5 Edit Name and Description of Stage


Name : Reply

Description :

5. To edit the stage, click **reply**, then select **Edit→Stage.** The Edit Stage Configuration page is displayed as in Figure 5-6.

Figure 5-6 Edit Error Handler page


Edit ErrorHandler : Request pipeline - validate loan application

Add an If...Then action to the Stage Error Handler

1. Click the **Add an Action** link, then select **Flow Control→If...Then** to add an **If...Then** action to the Reply stage.
2. Click the **<Condition>** link to specify the expression that needs to be validated. The Xquery Condition Editor page is displayed.

   a. Enter `$fault/ctx:errorCode` in the Operand window

   b. Select the = Operator from the drop-down list.

   c. Enter `"BEA-382505"` in the Value window

   d. Select the **And** Conjunction from the drop-down list.

   e. Click **Add** to add expression to the Expressions window.

3. Click **Save** to save the expression and return to the Edit Stage Configuration page.

Add a Replace Action to the Stage Error Handler

1. Click the **Add an Action** link within the **Then...** portion of the **If...Then** action to add a Replace statement.

2. Select **Message Processing→Replace**.

3. In the **Replace** statement, click the **XPath** link to edit the XPath expression. The XPath Expression Editor page is displayed.

4. From the **Select Structure** drop-down list in the Variable Structures pane, select **body**.

   A structural representation of the **body** element is displayed in the Variable Structures pane.

   a. Click + beside the **processLoanApp** element to expand the element.

      The **processLoanApp** element is within the **$body - processLoanApp (request)** element.

   b. Click + to expand the **loanRequest** element.

   c. Select the circle icon ● associated with the **Notes** element of **$body**, drag and drop it on the XPath Expression text box. The following XPath expression is written in the text box.

      ```
      ./exam:processLoanApp/loanRequest/java:Notes
      ```

      Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **Notes** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XPath expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XPath Expression text box.

5. Click **Validate** to validate the expression, then click **Save**.

6. In the Variable text field, enter `body`.

7. Click the `<Expression>` link to display the XQuery Expression Editor page and accept the default selection of **XQuery Text** in the page.

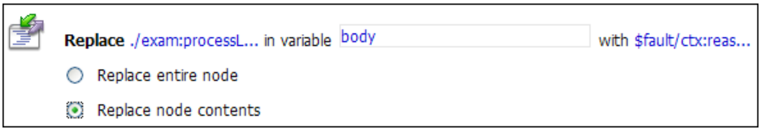
8. In the XQuery expression text box, enter:
   `$fault/ctx:reason/text()`

9. Click **Validate**, then **Save**.

10. Select the **Replace node contents** option. You need not replace the entire node.

The replace action should appear as in Figure 5-7.

Figure 5-7 Add Error Handler Using Replace Action



Replace ./exam:processL... in variable body with $fault/ctx:reas...

○ Replace entire node

● Replace node contents

Add a Report Action to Report the Error

Add a Report action and configure it to report the **body** context variable as detailed context, and set a key name of **errorCode** and the key value to the actual error code found in the **fault** context variable.

1. Click the **Replace** action and select **Add an Action**→**Reporting**→**Report**. The Report action is added to the page.

2. In the Report statement, click the **<Expression>** link. The XQuery Expression Editor page is displayed.

3. Accept the default selection of **XQuery Text** in the page.

4. In the **Variables Structures** pane, select **body - processLoanApp** from the **Select Structure** drop-down list.

   a. Select the icon associated with the **$body** element and drag and drop it on the XQuery Expression text box.

      The **$body** XQuery expression is written in the text box.

      Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **body** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XQuery expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XQuery Expression text box.

b. Click **Validate**, then **Save.**

You return to the Edit Stage Configuration page.

5. In the key-value pair table, select the link **Add a key** .

   a. Enter a name for the key. In this case, enter **errorCode**.

   b. In the key value field, click **XPath**.

6. In the **Variable Structures** pane, select **fault** from the **Select Structure** drop-down list.

   a. Select the icon ● associated with the **errorCode** and drag and drop it in the XPath Expression text box**.**

   The `./ctx:errorCode` expression is written in the text box.

   Note: The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **errorCode** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XPath expression text box and click **Copy Property**. The expression is copied to the text box. You can also copy the expression in the palette and paste it into the XPath Expression text box.

   This step enables you to index the reporting data by error code. The keys are a convenient way to identify a message. You can use the error code to search for reporting entries in the reporting module.

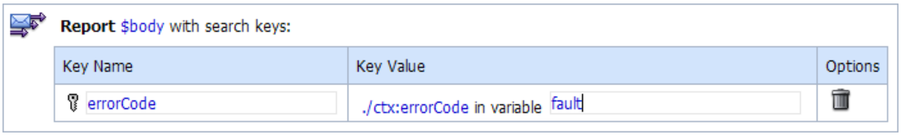
7. Click **Validate** to validate the XQuery, then click **Save**.

8. In the **Variable** text field, enter **fault**.

Note: To Add an action on a page, click the Action preceding the placement of the new action and select the Action to be added from there. Actions are added in sequence.

The report action should appear as in Figure 5-8.

Figure 5-8 Add Report Action

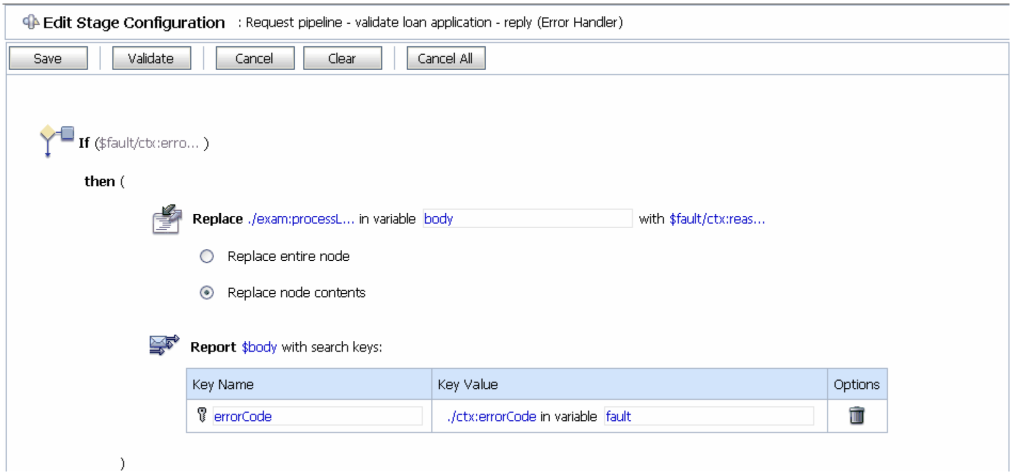


Report $body with search keys:

Key Name	Key Value	Options
errorCode	./ctx:errorCode in variable fault	🗑

The Edit Stage Configuration page appears as in Figure 5-9.

Figure 5-9 Request Actions for LoanGateway3 Proxy Service



Edit Stage Configuration : Request pipeline - validate loan application - reply (Error Handler)

If ($fault/ctx:erro...)

then (

Replace ./exam:processL... in variable body with $fault/ctx:reas...

○ Replace entire node

● Replace node contents

Report $body with search keys:

Key Name	Key Value	Options
errorCode	./ctx:errorCode in variable fault	🗑

)

Add an Else Action to the Stage Error Handler

Click the If...Then icon and select **Add else Condition** to add an Else action to the Stage Error Handler.

Add a Replace Action Within the Else Action

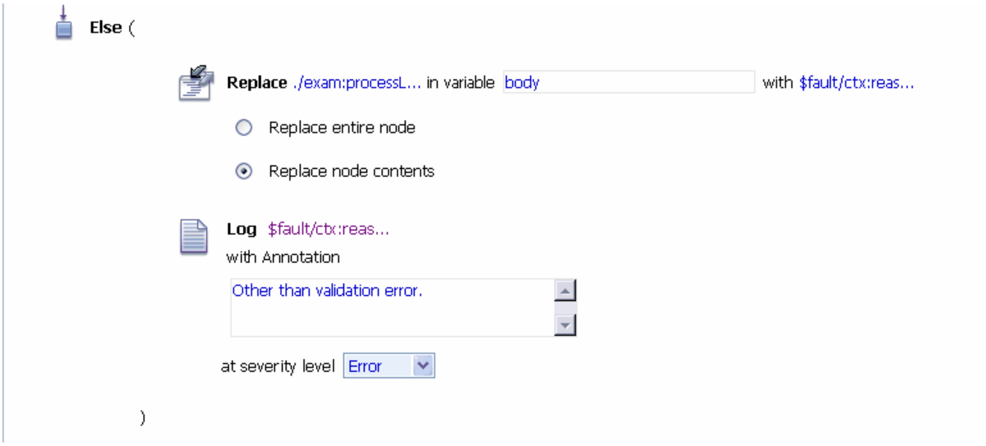
1. Click the **Add an Action** link to add a Replace statement. Select **Message Processing→Replace**.
2. Repeat all the steps in the Add a Replace Action to the Stage Error Handler section.

Add a Log Action Within the Else Action

1. Click the **Add an Action** link to add a Replace statement. Select **Reporting→Log**.
2. Click the `<Expression>` link to display the XQuery Expression Editor page. Enter `$fault/ctx:reason/text()` in the **XQuery Text** window.
3. Click **Save** to save the expression and return to the Edit Stage Configuration page.
4. Enter `Other than validation error`, in the With Annotation window.
5. Select **Error** from the **At Severity Level** drop-down list.

Now, the completed Else condition appears as in Figure 5-10.

Figure 5-10 Request Actions for LoanGateway3 Proxy Service - Else Condition



Else (

Replace ./exam:processL... in variable body with $fault/ctx:reas...

○ Replace entire node

● Replace node contents

Log $fault/ctx:reas...

with Annotation

Other than validation error.

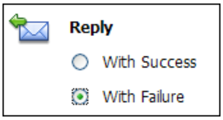
at severity level Error

)

Add a Reply Action to Reply with a Failure Code

1. Click the **Report** action and select **Add an Action→Flow Control→Reply**. The Reply action is added to the page.
2. Select the **With Failure** option as in Figure 5-11.

Figure 5-11 Reply Action to Reply with Failure Code



Reply

○ With Success

● With Failure

3. Click **Save** the stage configuration and return to the Edit Error Handler page.
4. Click **Save** the error handler configuration.

Step 5: Add a Default Routing Case

Configure the Route Node

Add a Route Node to the Pipeline pair, then configure the Route Node to route to the **normalLoan** business service.

1. Click **PipelinePairNode1**, then select **Add Route Node.**

2. Click **RouteNode1** and select **Edit→Route Node**. The Edit Stage Configuration page is displayed.
3. Click **Add an Action**, then select **Communication→Routing**.

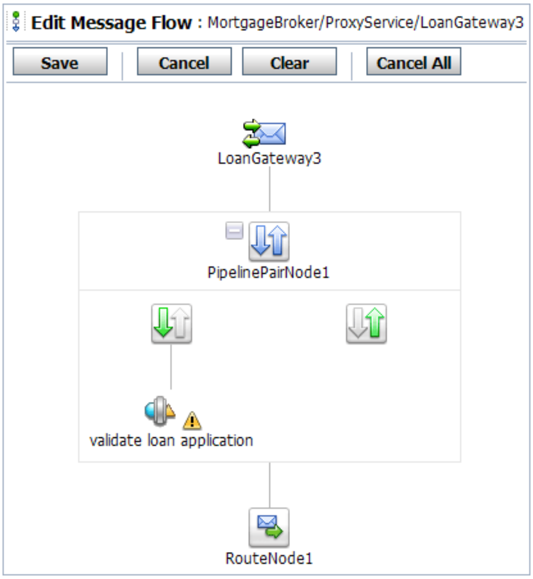
   The Route action is displayed in the page.
4. In the Route statement, click the **Service** link to define the service to which you want to route the messages. The Select Service page is displayed.
   a. Select the `NormalLoan` business service.
   b. Click **Submit** and return to the Edit Stage Configuration page.
5. In the **Operation** drop-down list, select `processLoanApp`.

   This is the operation on the `NormalLoan` business service that is invoked at run time.

   You have now defined the case for routing the loan application to the `NormalLoan` business service.

   The Edit Message Flow screen appears as in Figure 5-12.

Figure 5-12 Configure Message Flow of LoanGateway3 Proxy Service

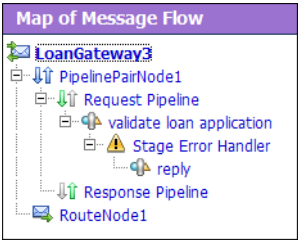

Edit Message Flow : MortgageBroker/ProxyService/LoanGateway3
Save
Cancel
Clear
Cancel All
LoanGateway3
PipelinePairNode1
validate loan application
RouteNode1

6. Click **Save** to save the configuration and return to the MortgageBroker/ProxyService page.

7. Click **Activate**, then **Submit**.

You can view the map of the message flow of the proxy service as in Figure 5-13.

Figure 5-13 Map of Message Flow of the LoanGateway3 Proxy Service



Map of Message Flow

```
LoanGateway3
  PipelinePairNode1
    Request Pipeline
      validate loan application
        Stage Error Handler
          reply
    Response Pipeline
  RouteNode1
```

Step 6: Test the Loan Application Routing Configuration

Now that you have configured BEA AquaLogic Service Bus to work with the client and the target business services, you can test the configuration. Testing is performed at the command prompt by changing the value of the **Loan duration** (in years) field in the `build.xml` file and then running `ant`. The behavior for the proxy service in this scenario depends on the correct value being entered for the loan duration. If the value is entered for the loan is a non integer, a fault is triggered and an error message results in the Notes field of the message. If the value is entered for the loan is an integer, the message is routed to the **NormalLoanApprovalService** business service.

This section describes how you can test the proxy service for both cases. The message returned in the command window after running a test indicates whether a validation exception results or whether the message was routed to the appropriate business service.

Note: Run the tests in this section multiple times (at least five times) to generate data to enable you to explore the reporting and monitoring features described in Step 7: Using Reporting and Monitoring.

This section includes the following test scenarios:

- To Test the Validation Exception
- To Test the NormalLoanApprovalService Business Service

To run this test correctly, ensure that WebLogic Server is started in the BEA AquaLogic Service Bus 2.5 (ServiceBusTutorial) domain.

To Test the Validation Exception

1. Open a command window.
2. If you have not run the `setEnv.cmd` to set up your environment, you must do so now.
   a. Change to the following directory:
   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\
   ```
   b. In this directory, at the command prompt, run the following command:
   ```
   setEnv.cmd
   ```
   Note: If you are testing the configuration on a UNIX system, run the `setEnv.sh` script at the command prompt.
3. Change to the following directory:
   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
   s\jws_basic\normal
   ```
4. To test the routing behavior of the proxy service, in this case, you must edit the `build.xml` file run target:
   a. Open `build.xml`.
   b. Scroll to the end of the `build.xml` file looking for the string:
   ```
   <arg line="http://$...
   ```
   c. Replace the string found at this location with:
   ```
   <arg line="http://${wls.hostname}:${wls.port}/loan/gateway3 steve 555
   2500 20.5 4.9 notes" />
   ```
   This changes the URI to that of the proxy service running in the tutorial domain, and specifies a value of 20.5 (a non-integer value) for the loan duration.

   The remainder of the input parameters specified on the `arg line` are described in Table 5-3.

Table 5-3 Input Parameters

Input Parameter	Description
steve	a single string representing the client name
555	the client ID

Table 5-3 Input Parameters

2500	the requested loan amount
20.5	the duration of the loan
4.9	the requested interest rate
notes	a string that can contain miscellaneous notes

5. Ensure that the business services are deployed as follows:

   a. In the command window, change to the following directory:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
   ices\jws_basic\normal
   ```

   b. Enter `ant`, then select **Enter**.

6. To run the test, at the command prompt, enter:

   ```
   ant run
   ```

   This tests the routing logic of the proxy service. A response message from the Web Service is displayed at the command line.

```
   [java] Loan Application Response: ALSB Validate action failed validation

    [java] Rate: 4.9 Duration: 20.5

BUILD SUCCESSFUL
```

The response message indicates that a validation exception occurred because the loan duration is not an integer.

To Test the NormalLoanApprovalService Business Service

1. Follow the procedure To Test the Validation Exception to edit the `build.xml` file in the following directory:

   ```
   BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservice
   s\jws_basic\normal
   ```

   Note: In this case (to test the **NormalLoanApprovalService** business service), enter a value of 20 for the loan duration in the `build.xml` file (see Table 5-3), then save the changes.

2. If you have not already done so, deploy the business services as follows:

a. In a command window, change to the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webserv
ices\jws_basic\normal
```

a. Enter `ant`, then press **Enter**.

3. To test the routing logic of the proxy service, on the command prompt, enter:

```
ant run
```

A response message from the Web Service is displayed at the command prompt:

```
[java] Loan Application Response: APPROVED BY THE <i><b>NORMAL</b></i>
LOAN APPLICATION PROCESSING SERVICE

[java] Rate: 4.9

BUILD SUCCESSFUL
```

The response message indicates that the **NormalLoan** business service processed the request because the loan duration requested is an integer.

Step 7: Using Reporting and Monitoring

Now that you have tested the proxy service you can explore the reporting and monitoring features of the AquaLogic Service Bus Console.

Note: You must have run the tests in Step 6: Test the Loan Application Routing Configuration multiple times (at least five times) to generate data to enable you to explore the reporting and monitoring features.

Reporting

AquaLogic Service Bus includes a JMS Reporting Provider for message reporting. The Reporting module in the AquaLogic Service Bus Console displays the information captured from this reporting provider. You can create your own reporting provider using the Reporting Service Provider Interface (SPI) if you do not wish to use the out-of-the-box reporting provider.

The dashboard of AquaLogic Service Bus Console contains reporting features that are presented in a drill-down format. When you explore the links presented, the information is made available to you.

For more information, see:

- Reporting in *AquaLogic Service Bus User Guide*
- Reporting in *AquaLogic Service Bus Console Online Help*

Monitoring

BEA AquaLogic Service Bus can monitor and collect run time information for both systems operations and business auditing purposes. AquaLogic Service Bus aggregates run time statistics that you can view on a customizable Dashboard. The Dashboard allows you to monitor the health of the system and alerts you to problems in your messaging services. With this information, you can quickly and easily isolate and diagnose problems as they occur.

For more information, see:

- Monitoring in *AquaLogic Service Bus User Guide*
- Monitoring in *AquaLogic Service Bus Console Online Help*