



# BEA AquaLogic Service Bus™

## Tutorials



# Contents

## 1. Introduction to the BEA AquaLogic Service Bus Tutorials

What is BEA AquaLogic Service Bus? .....	1-1
AquaLogic Service Bus Concepts .....	1-3
Overview of the Tutorials .....	1-5
Focus of the Tutorials .....	1-5
How to Use the Tutorials .....	1-6

## 2. Getting Started with the BEA AquaLogic Service Bus Tutorials

About the Scenarios .....	2-2
Setting Up the Tutorials .....	2-2
Step 1: Install AquaLogic Service Bus .....	2-3
Step 2: Create an AquaLogic Service Bus Domain .....	2-3
Step 3: Change Domain Configuration Settings .....	2-9
Step 4: Start BEA AquaLogic Service Bus .....	2-13
Step 5: Build and Deploy Business Services .....	2-14
Step 6: Log in to the AquaLogic Service Bus Console .....	2-15
Where to Go from Here .....	2-16

## 3. Tutorial 1. Routing a Loan Application

Prerequisites .....	3-1
Tutorial Objectives .....	3-2
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 . . . . .	3-17
Step 6: Configure the LoanGateway1 Proxy Service . . . . .	3-23
Step 7: Test Your Loan Application Routing Configuration . . . . .	3-33

## 4. 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 . . . . .	4-4
Step 3: Create the Resources . . . . .	4-5
Step 4: Configure Routing for LoanGateway2 Proxy Service . . . . .	4-10
Step 5: Configure Routing Request Actions for LoanGateway2 Proxy Service . . . . .	4-15
Step 6: Configure Routing Response Actions for LoanGateway2 Proxy Service . . . . .	4-27
Step 7: Add a Default Routing Case . . . . .	4-29
Step 8: Test the Loan Application Routing Configuration . . . . .	4-30
Best Practices . . . . .	4-33

# 5. 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 Run Time 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 . . . . . 5-5
  - Step 3: Create the Resources . . . . . 5-5
  - Step 4: Configure the LoanGateway3 Proxy Service. . . . . 5-7
  - Step 5: Add a Default Routing Case. . . . . 5-25
  - Step 6: Test the Loan Application Routing Configuration. . . . . 5-28
  - Step 7: Using Reporting and Monitoring . . . . . 5-31

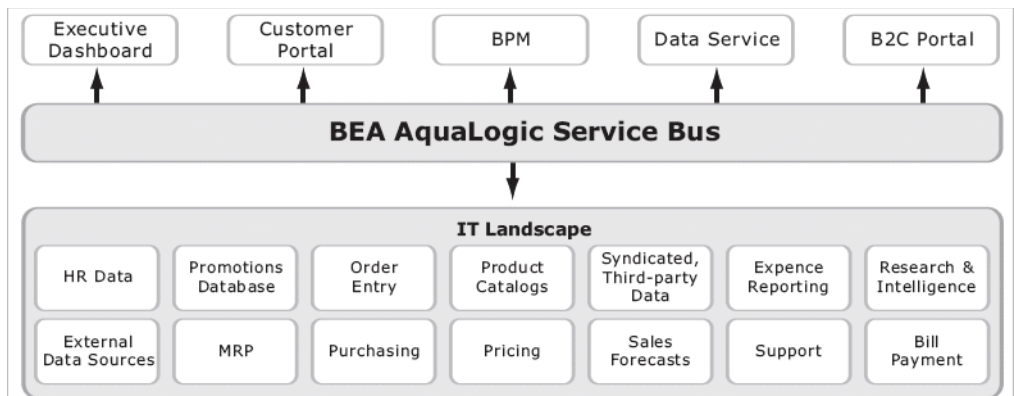


# Introduction to the BEA AquaLogic Service Bus Tutorials

## What is BEA AquaLogic Service Bus?

BEA AquaLogic Service Bus is a proven market-leading Enterprise Service Bus (ESB) built from the ground up for SOA lifecycle management. It provides foundation capabilities for service discovery and intermediation, rapid service provisioning and deployment, and governance. This service-infrastructure software adheres to the SOA principles of building coarse grained, loosely coupled, and standards-based services, creating a “neutral container” in which business functions may connect service consumers and back-end business services, regardless of underlying infrastructure. The following figure illustrates the role of BEA AquaLogic Service Bus as a service intermediary in an enterprise IT SOA landscape:

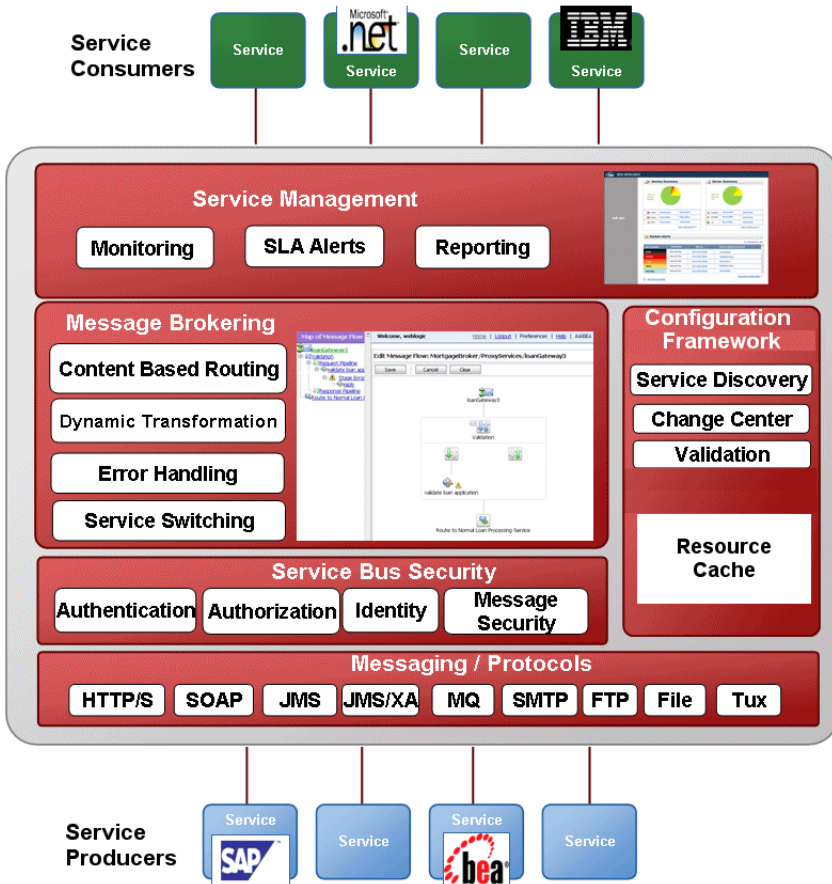
Figure 1-1 AquaLogic Service Bus Intermediary



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](#).

AquaLogic Service Bus manages the routing and transformation of messages in an enterprise system to promote seamless application integration. The following figure illustrates the functional components of AquaLogic Service Bus.

Figure 1-2 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. The concepts underlying AquaLogic Service Bus are briefly described in this section. For more information about AquaLogic Service Bus, see [BEA AquaLogic Service Bus Concepts and Architecture](#).

### Proxy Services and Business Services

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.

AquaLogic Service Bus allows you to implement proxy services independently and configure them dynamically, as driven by your business needs without costly infrastructure development efforts. The configuration functions are separated from the management functions in the AquaLogic Service Bus Console. 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.

For more information on proxy services, see [Proxy Services](#) 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](#).

*Business services* are AquaLogic Service Bus definitions of the enterprise services that exchange messages during business processes. A business service and its interface can be defined and configured using the AquaLogic Service Bus Console. A business service is configured by specifying its interface, type of transport it uses, its security requirements, and other characteristics. A business service definition is similar to that of a proxy service, but it does not have a pipeline. For more information on business services, see [Business Services](#) in [BEA AquaLogic Service Bus Concepts and Architecture](#). For information on how to configure a business service using the WebLogic Service Bus Console, see “Adding a Business Service” in [Business Services](#) in the [Using the AquaLogic Service Bus Console](#).

## Message Flows and Pipelines

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. A pipeline is a named sequence of stages containing actions, representing a non-branching one-way processing path. It is used to specify the message flow for service requests and responses. 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.

For more information on proxy services, see [Message Flow Modeling](#) in *BEA AquaLogic Service Bus Concepts and Architecture*.

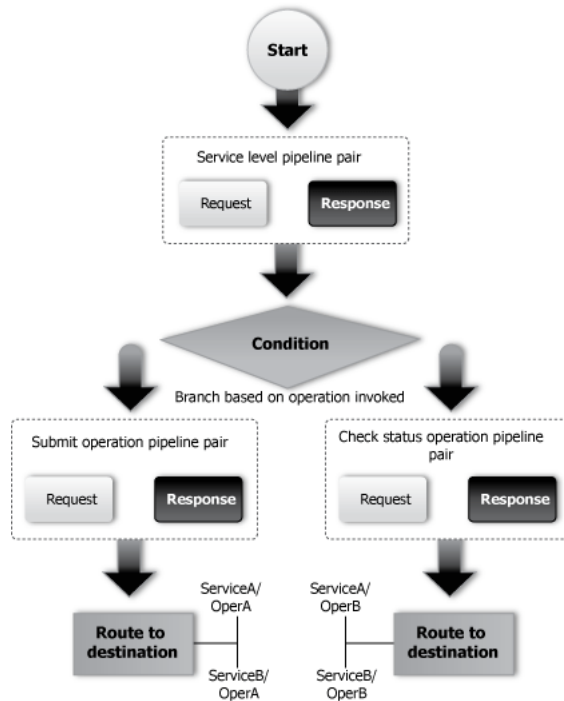
## Pipeline Pairs

*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.

The following figure illustrates sample operation pipelines in a proxy service.

Figure 1-3 Example of Operation Pipelines in a Proxy Service



For more information, see [Pipeline Pairs](#) in *BEA AquaLogic Service Bus Concepts and Architecture*.

## 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 AquaLogic Service Bus. You will learn how to configure and use 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 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 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 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\*](#)

## Introduction to the BEA AquaLogic Service Bus Tutorials

# Getting Started with the BEA AquaLogic Service Bus Tutorials

The AquaLogic Service Bus tutorials are based on a typical Web Services scenario that uses the AquaLogic Service Bus for configuring business services and adaptive message routing. This section describes tasks required to configure the AquaLogic Service Bus and WebLogic Server environment to deploy the business cases used in the tutorials. The AquaLogic Service Bus example suite is used as a basis for instruction in the following tutorials. (See [BEA AquaLogic Service Bus Examples Guide](#)).

You will perform the following pre-requisite tasks to prepare the tutorial domain:

- Install the AquaLogic Service Bus
- Configure the WebLogic Server environment and domain settings
- Deploy pre-built business services to implement the sample business scenario
- Start BEA AquaLogic Service Bus
- Log in to the AquaLogic Service Bus Console

Upon completion of this section, you will know how to deploy a typical AquaLogic Service Bus environment in which you can define resources and design web services. It is necessary to complete all steps in this section before you start the tutorials.

## About the Scenarios

The tutorials are based on a mortgage broker scenario describing a typical loan application process. 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.

In the first tutorial you will become familiar with the basic functionality of the AquaLogic Service Bus that facilitates web service mediation, including message routing and web service resource creation. In the subsequent tutorials you will develop and customize web services for routing, transforming and validating a loan application.

## Setting Up the Tutorials

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

- [Step 1: Install AquaLogic Service Bus](#)
- [Step 2: Create an AquaLogic Service Bus Domain](#)
- [Step 3: Change Domain Configuration Settings](#)
- [Step 4: Start BEA AquaLogic Service Bus](#)
- [Step 5: Build and Deploy Business Services](#)
- [Step 6: Log in to the AquaLogic Service Bus Console](#)

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 NormalLoan business 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:	
<i>BEA_HOME</i> /weblogic92/samples/servicebus/examples/src/examples/webservices/jws_basic/normal	The files contained in this folder are those required by the NormalLoan business service. The same file structure exists for the other three business services used in this tutorial ( <b>ManagerLoanReview</b> , <b>LoanSaleProcessor</b> , and <b>CreditRating</b> ).
SimpleBean.java	The jws-181 Web Service definition.
/client/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.

## Step 1: Install AquaLogic Service Bus

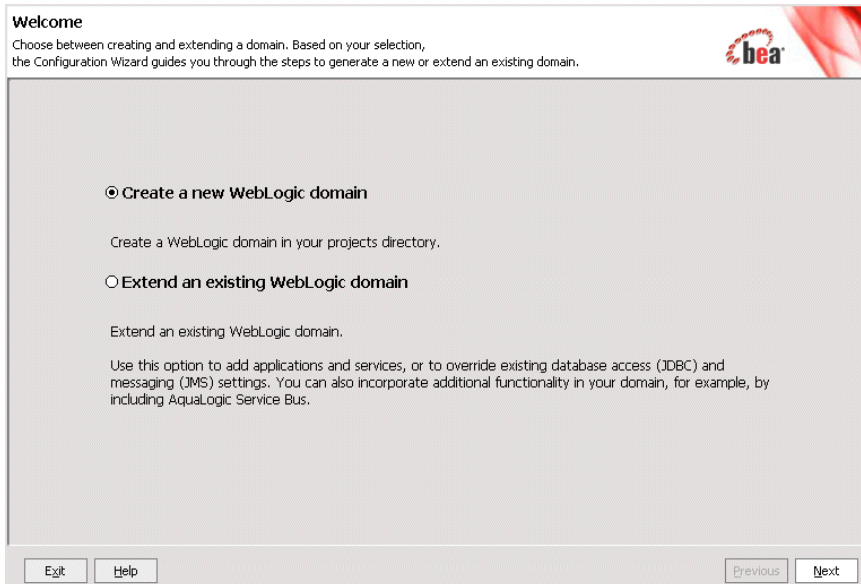
Install the AquaLogic Service Bus. For installation instructions, see [AquaLogic Service Bus Installation Guide](#).

## Step 2: Create an AquaLogic Service Bus Domain

To develop and run the AquaLogic Service Bus tutorials, you must create an AquaLogic Service Bus domain using the Configuration Wizard. You can invoke the Configuration Wizard in the graphical mode or console mode. For this tutorial, you will start the Configuration Wizard in graphical mode.

1. To start the Configuration Wizard in graphical mode, choose the Configuration Wizard option in the BEA Products group from the Windows Start menu :  
Start → All Programs → BEA Products → Tools → Configuration Wizard
2. Using the Configuration Wizard, select the product components you want to include in your domain. (See *Creating WebLogic Domains Using the Configuration Wizard*).
  - a. In the **Welcome** page, the **Create a new WebLogic domain** radio button is selected by default. Accept the default selection and click **Next** to proceed.

**Figure 2-1 Configuration Wizard - Create Domain Option**



- b. In the **Select Domain Source** page, select the **AquaLogic Service Bus** radio button. Click **Next** to proceed.

**Figure 2-2 Configuration Wizard - Select Domain Source**

Select Domain Source

Select the source from which the domain will be created. You can create the domain by clicking on the required components or by selecting from a list of existing domain templates.

**Generate a domain configured automatically to support the following BEA products:**

- WebLogic Server (Required)
- Workshop for WebLogic Platform
- AquaLogic Service Bus

**Base this domain on an existing template**

Template location:

- c. In the **Configure Username and Password** page, configure a new administrator username and password. You can use the default *weblogic* username but you must create a unique password.

**Figure 2-3 Configuration Wizard - Create Username and Password**

**Configure Administrator Username and Password**  
Create a user to be assigned to the Administrator role.  
This user is the default administrator used to start development mode servers.

[Disgard Changes](#)

\*User name:

\*User password:

\*Confirm user password:

Description:

- d. In the **Configure Server Start Mode and JDK** page, **Development Mode** is selected as the default startup mode for the WebLogic domain. Accept the default selection and click **Next** to proceed.

Figure 2-4 Configure Server Start Mode and JDK

**Configure Server Start Mode and JDK**

Choose the WebLogic domain startup mode and the J2SE Development Kit (JDK) to be used for the domain.

Before putting a domain into production, BEA recommends that you familiarize yourself with the content of "Securing a Production Environment" at <http://e-docs.bea.com/wls/docs90/lockdown/index.html>.

To use WebLogic JRockit in production, BEA recommends developing and testing your applications with WebLogic JRockit early in the project cycle. For information about WebLogic JRockit, see the WebLogic JRockit documentation.

**WebLogic Domain Startup Mode**

**Development Mode**  
Utilize boot.properties for username and password and poll for applications to deploy. Sun JDK recommended for better startup performance during iterative development.

**Production Mode**  
Require the entry of a username and password and do not poll for applications to deploy. WebLogic JRockit JDK recommended for better runtime performance and management.

**JDK Selection**

**BEA Supplied JDKs**

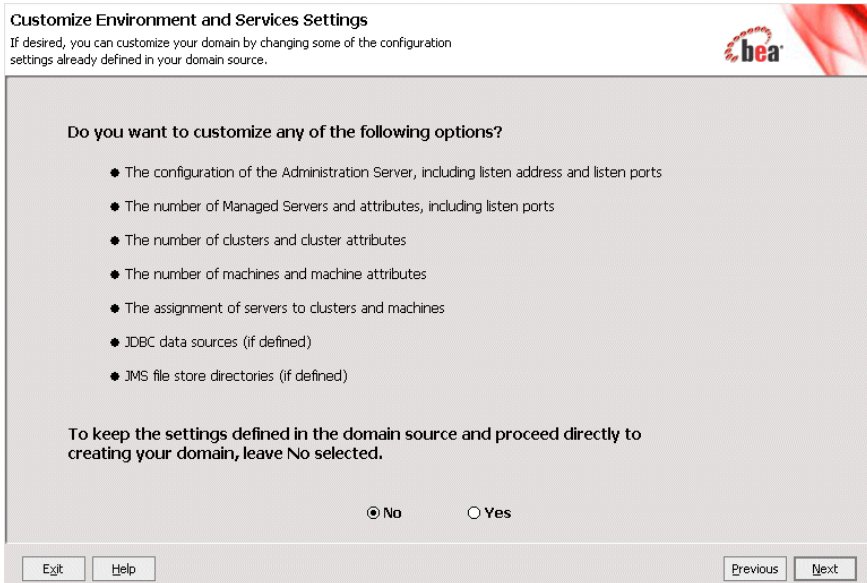
- JRockit SDK 1.5.0\_06 @ D:\bea\jrockit90\_150\_06
- Sun SDK 1.5.0\_06 @ D:\bea\jdk150\_06

**Other JDK**

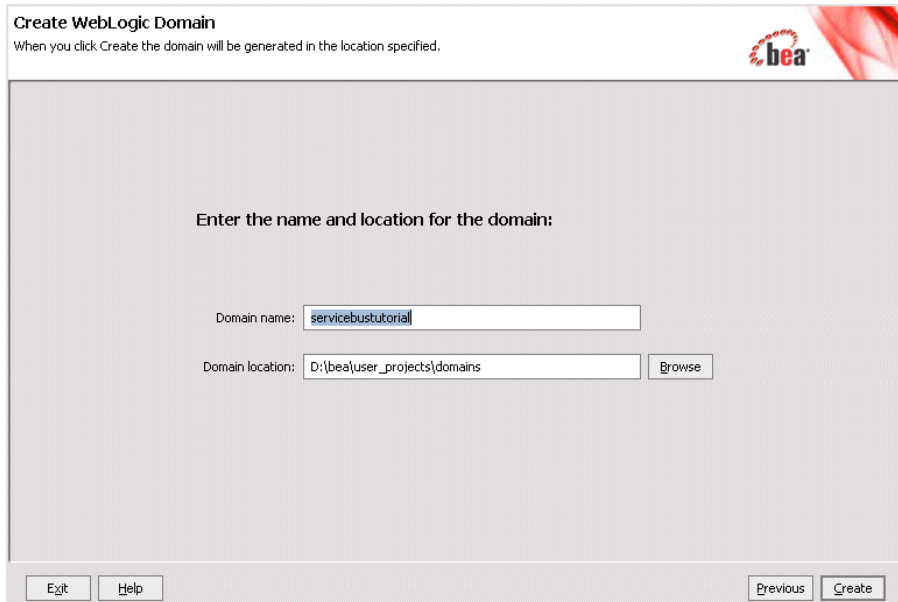
Location:

- e. In the **Customize Environment and Services Settings** page, the default customization option for the environment and services settings is **No**. Accept this option and click **Next** to proceed.

**Figure 2-5 Customize Environment and Service Settings**



f. Create a domain by specifying domain name as `ServiceBusTutorial`. Click **Create**.

**Figure 2-6 Configuration Wizard - Create Weblogic Domain**

## Step 3: Change Domain Configuration Settings

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

`BEA_HOME\weblogic92\samples\servicebus\examples`

This directory has two subdirectories: `\build` and `\src`. You must configure the files in the `\src` directory.

**Table 2-2 Configuration Settings**

Folder Name	Description
<code>\build</code>	This folder contains the files required to run the pre-configured AquaLogic Service Bus examples provided in the AquaLogic Service Bus Examples domain.
<code>\src</code>	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 <code>examples.properties</code> .

## Configure the `examples.properties` File

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

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

### To Modify the `examples.properties` File

1. Open the `examples.properties` file using your default text editor. The WebLogic server configuration in the `examples.properties` file will be as shown in the following file fragment:

**Figure 2-7** `example.properties` - Initial Settings

```
#####
# This file contains property settings for #
# use in Tutorial' ant build scripts. #
#####

### WebLogic configuration ###
-- Configure per your WLS installation.
wls.hostname=localhost
wls.port=7021
wls.username=weblogic
wls.password=weblogic
wls.server.name=AdminServer

### Quicksilver Examples locations ###
examples.home.dir=.
examples.src.dir=${examples.home.dir}/src
examples.build.dir=${examples.home.dir}/build
ex.webapp.dir=${examples.build.dir}/examplesWebApp
ex.webapp.webinf.dir=${ex.webapp.dir}/WEB-INF
ex.webapp.classes.dir=${ex.webapp.webinf.dir}/classes
```

2. Change the hostname to the name of the machine on which the `ServiceBusTutorial` domain is running. The default hostname `localhost` specified in the `examples.properties` can be used.
3. Change the port number to the port number of the `ServiceBusTutorial` domain. The port number `7021` specified in the `examples.properties` file, is the default port number of the examples domain.
4. Enter the new username and password that you specified when creating the domain. The default username and password in the `examples.properties` file is set as `weblogic`.
5. If you had changed the default name of the Admin Server, change the server name to the Admin Server name you specified when you created the `ServiceBusTutorial` domain. The default name assigned to the Admin Server by the Configuration Wizard is `AdminServer`.



**Note:** The username *newuser* and password *newpassword* shown in the following edited `examples.properties` file, have to be substituted with your new username and password.

6. Save the edited `examples.properties` file.

After you have completed editing settings, the `examples.properties` file must be displayed as shown in the following figure.

**Figure 2-8** `example.properties` - Edited Settings

```
#####
# This file contains property settings for #
# use in Tutorial' ant build scripts. #
#####

### WebLogic configuration ###
-- Configure per your WLS installation.
wls.hostname=localhost
wls.port=7001
wls.username=newuser
wls.password=newpassword
wls.server.name=xbusServer

### QuickSilver Examples locations ###
examples.home.dir=.
examples.src.dir=${examples.home.dir}/src
examples.build.dir=${examples.home.dir}/build
ex.webapp.dir=${examples.build.dir}/examplesWebApp
ex.webapp.webinf.dir=${ex.webapp.dir}/WEB-INF
ex.webapp.classes.dir=${ex.webapp.webinf.dir}/classes
```

## Edit the setEnv Script

The `setEnv` script contains the path to the `setDomainEnv.cmd` file used in a domain. The `setEnv` script file is named `setEnv.cmd` on Windows and named `setEnv.sh` on UNIX systems. You must edit the `setEnv` script file appropriate for your operating system, to reflect the path to the `setDomainEnv.cmd` file in the `ServiceBusTutorial` domain.

### To Modify the setEnv Script

1. Locate the script file appropriate for your operating system in the following directory:

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

2. Open the script file using your default text editor.

3. Edit the script file to reflect the following new path:

```
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.

4. Save the edited script file.

## To Run the `setEnv` Script

### On Windows:

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

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

3. On the command line, run the following command:

```
setEnv.cmd
```

### On UNIX:

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

```
% setEnv
```

where `BEA_HOME` is the directory in which you installed AquaLogic Service Bus.

## Edit the `setDomainEnv.cmd` file (Windows 2000 Users Only)

You must execute the following additional step if you use Windows 2000. On systems using Windows 2000, the length of the class path prevents you from running `ant` tasks executed during the tutorial. This additional step provides a workaround using the `subst` command to shorten the classpath passed by the `ant` tasks.

### To Shorten the Class Path

1. Open the `setDomainEnv.cmd` file located in the following directory:

```
BEA_HOME\user_projects\domains\ServiceBusTutorial\bin\
```

2. In the `setDomainEnv.cmd` file locate the entry that sets the `PRE_CLASSPATH`. The instruction will be listed as follows:

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

3. Identify any 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 example:

```
set
PRE_CLASSPATH=%PRE_CLASSPATH%;X:\sb-public.jar;X:\sb-internal.jar;Y:\wlicommon.jar;Y:\qs_pl3n_system.jar;X:\xbus-core.jar;Z:\wlxbean.jar;Z:\xquery.jar;Z:\apache_xbean.jar;Z:\binxml.jar
```

5. Save changes to the `setDomainEnv.cmd` file and close the file.
6. Run the `setDomainEnv.cmd` script by entering the following commands:

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

## Step 4: Start BEA AquaLogic Service Bus

You can start the AquaLogic Service Bus using one of the following methods:

- On Windows systems, from the Windows Start menu select:  
Start → All Programs → BEA Products → User Projects → *domain\_name* → Start Server for AquaLogic Service Bus Domain  
  
You must specify the name of the AquaLogic Service Bus domain you created for this tutorial as the *domain\_name*.
- 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.

When the AquaLogic Service Bus is started, a server command console window will display status information about the WebLogic Server.

**Note:** A valid username and password is required to start the server, if a production mode domain is used. For more information about creating and configuring domains, see [Creating WebLogic Domains Using the Configuration Wizard](#).

## Step 5: Build and Deploy Business Services

After you start the WebLogic server, you must run the build script to perform the following tasks:

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

### To Build and Deploy the Business Services

The following four web services have to be built and deployed, to implement the business services associated with the mortgage broker scenario used in the tutorials:

- **NormalLoan** - normal loan application business service
- **CreditRating** - credit rating business service
- **LoanSaleProcessor** - large loan application business service
- **ManagerLoanReview** - manager loan application response business service

1. Open a command window and ensure that you have run the `setEnv` script. For instructions to run the `setEnv` script, see [“To Run the setEnv Script” on page 2-12](#).
2. Navigate to the `webservices\jws_basic` directory. This directory contains four business services folders, one for each of the business services to be implemented in this tutorial.

```
cd
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservices
\jws_basic
```

3. Navigate to the appropriate directory for the business service you want to build. For example, to build the `NormalLoan` business service, select the `normal` folder:

```
cd normal
```

**Note:** To build the other business services (**CreditRating**, **LoanSaleProcessor**, and **ManagerLoanReview**), select the respective business service folders.

4. To deploy the Web Service, enter `ant` and press **Enter**. A message is displayed on the console, indicating success or failure of the executed operation.
5. To test the deployed Web Service, enter `ant run` and press **Enter**. A message from the business service is displayed at the command line. The message indicates whether the Web Service was run successfully.

- Repeat [step 3](#) through [step 5](#) of these build and deploy instructions to build each of the remaining Web Services (**CreditRating**, **LoanSaleProcessor**, and **ManagerLoanReview**) used in this tutorial.

## Step 6: Log in to the AquaLogic Service Bus Console

The AquaLogic Service Bus Console is a web services management dashboard which allows you to monitor web services and servers, and perform service management tasks. The console enables you to perform operational tasks such as configuring proxy and business services, setting up security, managing resources, and capturing data for tracking or regulatory auditing. It provides views to monitor current state and health of the AquaLogic Service Bus environment, by displaying detailed statistics about servers, services, and alerts. The AquaLogic Service Bus Console also enables you respond rapidly and effectively to changes in your service-oriented environment.

### To log in to the AquaLogic Service Bus Console

- Open a browser window and enter the following URL:

```
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:7001/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 tutorial domain.

**Note:** You can also open the AquaLogic Service Bus Console from the Windows Start menu by selecting the following options:

Start → All Programs → BEA Products → User Projects → *domain\_name* → Start Server for AquaLogic Service Bus Admin Console

You must specify the name of the AquaLogic Service Bus domain you created for this tutorial as the *domain\_name*.

## 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 loan application routing scenario. Each of the tutorials include instructions to design and configure the AquaLogic Service Bus resources, and procedures you can use to test the completed configurations.

# Tutorial 1. Routing a Loan Application

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

This section includes the following topics:

- [Prerequisites](#)
- [Tutorial Objectives](#)
- [Definition of the Scenario](#)
- [Steps in This Tutorial](#)

## Prerequisites

The following tutorial has to be completed before beginning this tutorial:

- [“Getting Started with the BEA AquaLogic Service Bus Tutorials” on page 2-1.](#)

## Tutorial Objectives

The objective of this tutorial is to create and test a routing scenario 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

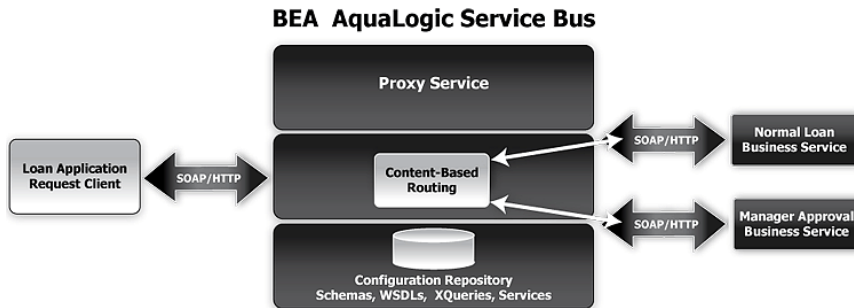
A primary mortgage company uses AquaLogic Service Bus to route loan applications to appropriate business services based on the interest rate requested. An application containing a request for a rate less than 5% requires management approval and is routed to an appropriate business service for processing. 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.

The following figure summarizes the logical architecture to support this scenario. It illustrates how 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

## 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 **ManagerLoanReview** business service; otherwise it is routed to the **NormalLoan** business service. The target business service returns a response similar to the following message:

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

**Note:** If the loan request is processed by the **NormalLoan** business service, **MANAGER** is replaced with **NORMAL** in the generated response.

## Required Resources

The following table lists the resources required to develop and run this tutorial.

Table 3-1 Routing Loan Application Tutorial Resources

Resource Name	Description
normalLoan	This is the WSDL resource
NormalLoan	This is an external business service used by AquaLogic Service Bus
ManagerLoanReview	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, 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. By performing the following steps you can design and configure a proxy service and the associated resources in AquaLogic Service Bus, to implement this use 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](#)

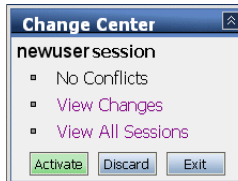
### Step 1: Prepare the Environment

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

Using the AquaLogic Service Bus Console, you can monitor resources and configurations in the AquaLogic Service Bus environment. You can perform system monitoring without initiating a session. However, you must first create an active session in the AquaLogic Service Bus Console in order to update or delete resources and modify their configuration properties.

The Change Center in the console, allows you to create and manage sessions. All updates to the system configuration during the current session are saved as temporary files. These changes come into effect only when the current session is activated. The following figure illustrates the Change Center pane available on the AquaLogic Service Bus Console.

**Figure 3-2 Manage Sessions Using Change Center**

In addition to creating and activating sessions, the Change Center also allows you to perform the following additional functions summarized in the table below.

**Table 3-2 Change Center Options**

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 recommended that you click **Create** before modifying any system configuration settings on the console. It is a good practice to save your configuration changes incrementally. To save and activate configuration changes in the current session, you must click **Activate** after you have completed making a set of changes.

## Create the MortgageBroker Project and File Folders

In the AquaLogic Service Bus system environment, system configuration entities are grouped into user defined projects. On the AquaLogic Service Bus Console, the Project Explorer page

displays a default project and the various user defined projects under the Projects folder. This page allows you to perform all project configuration tasks. Each project is represented by a project folder on the console. You can add new project folders and navigate the project trees. You can collapse and expand folders as required by clicking on them.

There are three pre-defined resources types in the AquaLogic Service Bus system environment. Each resource type is represented as a sub folder within the Project folder. You can create and configure resources of different types within the resource type folders.

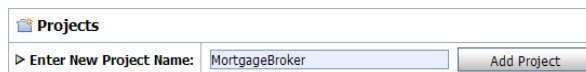
This section describes the steps to start a session and create a **MortgageBroker** project under the Projects folder. For this scenario, you will create three resource folders in the **MortgageBroker** project, to hold each of the following pre-defined resource types:

- ProxyService
- BusinessService
- WSDL

### 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**.  
**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 shown in the following figure.

**Figure 3-3 Create a Project**



The screenshot shows a web interface for the 'Projects' section. At the top, there is a header 'Projects' with a folder icon. Below it, there is a form with a label 'Enter New Project Name:' followed by a text input field containing 'MortgageBroker'. To the right of the input field is a button labeled 'Add Project'.

4. Click **Add Project**.

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

You must 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 the following figure.

**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. The Change Center environment setting options are listed in [Table 3-2 “Change Center Options”](#) .

## 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 WSDL Resource

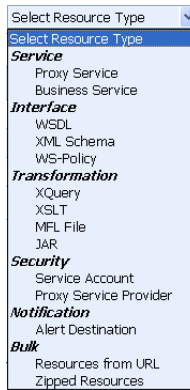
A WSDL defines the public contract (interface specification) between a client and a service, whether the service is a proxy service or business service. It is the formal description of a Web Service. A WSDL is used to describe what a Web Service’s interface is, where it resides, and how to invoke it. You create the WSDL resource first since 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**



The Create a New WSDL Resource page is displayed.

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\webservices\jws_basic\normal\NormalLoanApprovalService.wsdl
```
  - c. You can optionally enter information in the **Resource Description** to document the resource.
  - d. Click **Save** to create the WSDL resource.

The resource is now displayed in the resources pane. The displayed Create a New WSDL Resource page must be similar to the page shown in the following figure.

**Figure 3-6 Create a WSDL Resource**

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

## Create the managerApproval WSDL Resource

You must create a WSDL resource associated with the ManagerApproval Service, using the procedure “[To Import a WSDL](#)” on page 3-7. To import the ManagerApproval Service WSDL resource, you must use the configuration parameters listed in the following table.

**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 shown in the following figure.

**Figure 3-7 Create a Proxy Service**

Create a Proxy Service - General Configuration (MortgageBroker/ProxyService/)

Service Name\*

Description

Service Type\*

Create a New Service

WSDL Web Service

Messaging Service  (port or binding)

Any SOAP Service SOAP 1.1

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 page is displayed. The proxy service is based on the WSDL resource that you originally created, hence you must reference the resource here.

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



Figure 3-8 Select WSDL Definition

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

**Bindings**  
 NormalLoanApprovalServiceSoapBinding  
**Ports**  
 helloPort

- In the **Select WSDL Definitions** pane, from the **Ports** category, click on the WSDL port for the **normalLoan** WSDL (helloPort).
- Click **Submit** to complete your selection. The port name is displayed in the WSDL port text field as shown in the following figure.

Figure 3-9 LoanGateway1 Proxy Service

Service Name\*

Description

Service Type\* **Create a New Service**

WSDL Web Service    
 (port)  
 Messaging Service  
 Any SOAP Service

**Create From Existing Service**

Business Service    
 Proxy Service

Proxy Service Provider

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

The Create a Proxy Service - Transport Configuration page is displayed as shown in the following figure.

**Figure 3-10 Transport Configuration of Proxy Service**

Figure 3-10 shows the 'Edit a Proxy Service - Transport Configuration' page. The page title is 'Edit a Proxy Service - Transport Configuration (MortgageBroker/ProxyService/LoanGateway1)'. The form contains the following elements:

- Protocol\***: A dropdown menu with 'http' selected.
- Endpoint URI\***: A text input field with 'Format: /someName' and the value '/loan/gateway1' entered.
- Get All Headers**: Radio buttons for 'Yes' and 'No', with 'No' selected.
- Header**: A text input field and an 'Add' button.
- Table**: A table with columns 'HEADER' and 'ACTION'. The table is currently empty, and the text 'There are no headers configured.' is displayed below it.

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. The assumed default protocol is **http** since it is used 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 shown in the following figure.

**Figure 3-11 HTTP Transport Configuration of Proxy Service**

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

Authentication  None  
 Basic  
 Custom Authentication (See Advanced Settings)

Dispatch Policy default

Request Encoding

Response Encoding

Advanced Settings

<< Prev. | Next >> | Last >> | Cancel

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

The Create a Proxy Service - Message Level Security Configuration page is displayed as shown in the following figure.

**Figure 3-12 Message Level Security Configuration of Proxy Service**

Create a Proxy Service - Message Level Security Configuration (MortgageBroker/ProxyService/LoanGateway1)

Use Custom Authentication  Yes  No (See Advanced Settings)

Advanced Settings

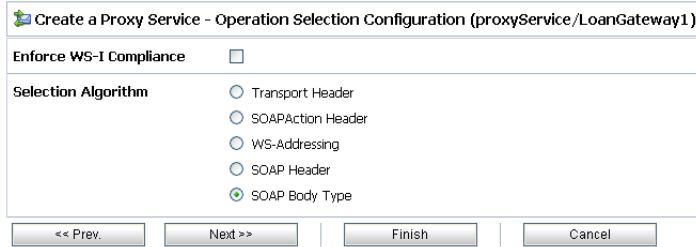
<< Prev. | Next >> | Last >> | Cancel

You do not need to configure custom authentication settings on this page since a security policy has not been defined for the Web Service.

13. Click **Next** to accept the default settings for the message level security configuration.

The Create a Proxy Service - Operation Selection Configuration page is displayed as shown in the following figure. On this page you will select the metadata accompanying the message, which identifies the Web Service operation to be invoked.

**Figure 3-13 Operation Selection Configuration of Proxy Service**



14. Accept the default **SOAP Body Type** selection algorithm.

**Note:** 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 there are multiple public methods accessible to a client, 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. The SOAP header can also specify the operation. However, by convention, the SOAP body defines this.

15. Click **Next**.

The Create a Proxy Service - Summary page is displayed as shown in the following figure. This page shows a summary of configuration settings for the proxy service.

Figure 3-14 Proxy Service - Configuration Settings Summary

Edit a Proxy Service - Summary (MortgageBroker/ProxyService/LoanGateway1)	
<b>General Configuration</b>	
Service Name	LoanGateway1
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Proxy Service Provider	
<b>Transport Configuration</b>	
Protocol	http
Endpoint URI	/loan/gateway1
Get All Headers	No
Headers	
<b>HTTP Transport Configuration</b>	
Authentication	None
<b>Message Level Security Configuration</b>	
Custom Authentication	None
<b>Operation Selection Configuration</b>	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type

Before registering the proxy service, you can review the configuration settings and change them if necessary, by clicking **Edit**.

- After reviewing 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.

Figure 3-15 Proxy Services in Resource Pane

Resources			
▶ Create Resource: <input type="text" value="Select Resource Type"/>			
			Items 1-3 of 3
<input type="checkbox"/>	Name ▲	Resource Type	Actions
<input type="checkbox"/>	 LoanGateway1	Proxy Service	   

- To turn on monitoring for the **LoanGateway1** proxy service, click the name of the proxy service— **LoanGateway1** in the **Resource** pane.

The View a Proxy Service page is displayed as shown in following figure. The **Operational Settings** tab on this page has options to enable monitoring for the proxy service.

**Figure 3-16 Monitoring Configuration of Proxy Service**

View a Proxy Service (MortgageBroker/ProxyService/LoanGateway1)		
Last Modified By	newuser	Description - no description -
Last Modified On	12/20/06 11:47 AM	
References	1	
Referenced By	0	

Configuration Details | Operational Settings | SLA Alert Rules

LoanGateway1 Operational Settings	
Service State	<input checked="" type="checkbox"/> Enabled
Service Monitoring	<input type="checkbox"/> Enabled
Aggregation Interval	0 hrs, 10 mins
Service SLA Alerting	<input checked="" type="checkbox"/> Enable Alerting at Normal level or above
Service Pipeline Alerting	<input checked="" type="checkbox"/> Enable Alerting at Normal level or above
Service Message Reporting	<input checked="" type="checkbox"/> Enabled
Service Logging	<input checked="" type="checkbox"/> Enable Logging at Debug level or above

Back | Update | Reset

To enable monitoring:

- a. Click the **Operational Settings** tab.
- b. Select the **Enabled** check box for the **Service Monitoring** setting.
- c. 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 AquaLogic Service Bus Console dashboard. The default value for the **Aggregation Interval** setting is set to 10 minutes.

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 Operations Guide*

- d. Click **Update**.

- e. Click the **Back** button at the bottom of the Monitoring Configuration page, to return to the MortgageBroker/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 ManagerLoanReview Business Service](#)

### Create the NormalLoan Business Service

1. In **Project Explorer**, select the **BusinessService** 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 start a session and make changes to the system configuration.

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

The Create a Business Service - General Configuration page is displayed as shown in the following figure.

Figure 3-17 Create Business Service

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

Service Name\*

Description

Service Type\*

**Create a New Service**

WSDL Web Service

Transport Typed Service  (port or binding)

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 shown in the following figure.



**Figure 3-18 Transport Configuration of a Business Service**

Create a Business Service Transport Configuration (MortgageBroker/BusinessService/NormalLoan)

Protocol*	http		
Load Balancing Algorithm	none		
Endpoint URI*	Format: http://host:port/someService		
	http://	Add	
	EXISTING URIS		OPTIONS
	http://localhost:7001/njws_basic_ejb/NormalSimpleBean		↑ ↓ 🗑
Retry Count	0		
Retry Interval	30		

- Accept the default Protocol, **http**.
- 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.

- The End Point URI of the business service is the endpoint 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`

The host and port values specified for `<host:port>` represent the machine and port on which your AquaLogic Service Bus server is running. They must be the same as the values specified in the following properties file:

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

For more information, see [“Configure the examples.properties File” on page 2-10](#).

**Note:** Delete any invalid endpoint URI in the existing URI list.

- 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 shown in the following figure.

**Figure 3-19 HTTP Transport Configuration of a Business Service**

Create a Business Service HTTP Transport Configuration (MortgageBroker/BusinessService/NormalLoan)

Timeout

HTTP Request Method  POST  
 GET

Authentication  None  
 Basic

Service Account

Follow HTTP redirects

Dispatch Policy

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 shown in the following figure.

**Figure 3-20 SOAP Binding Configuration of Business Service**

Create a Business Service - SOAP Binding Configuration (MortgageBroker/BusinessService/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 shown in the following figure. You can now check if all the settings of the business service have been implemented correctly.

**Figure 3-21 Business Service - Configuration Settings Summary**

General Configuration	
Service Name	NormalLoan
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Transport Configuration	
Protocol	http
Load Balancing Algorithm	none
Endpoint URI	http://localhost:7001/rjws_basic_ejb/NormalSimpleBean
Retry Count	0
Retry Interval	30
HTTP Transport Configuration	
Timeout	0
HTTP Request Method	POST
Authentication	None
Follow HTTP redirects	ENABLED
SOAP Binding Configuration	
Enforce WS-I Compliance	No

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

To turn on monitoring for the **NormalLoan** business service, click the name of the business service—**NormalLoan** in **Resources** pane. The View a Business Service page for NormalLoan business service is displayed as shown in the following figure. The Operational Settings tab on this page has options to enable monitoring for the business service.

**Figure 3-22 View Business Service Page**

View a Business Service (MortgageBroker/BusinessService/NormalLoan)		
Last Modified By	newuser	<b>Description</b> - no description -
Last Modified On	12/14/06 11:36 AM	
References	1	
Referenced By	3	

<a href="#">Configuration Details</a>	<a href="#">Operational Settings</a>	<a href="#">SLA Alert Rules</a>
---------------------------------------	--------------------------------------	---------------------------------

NormalLoan Operational Settings	
Service State	<input checked="" type="checkbox"/> Enabled
Service Monitoring	<input type="checkbox"/> Enabled
Aggregation Interval	0 hrs. 5 mins
Service SLA Alerting	<input checked="" type="checkbox"/> Enable Alerting at Normal level or above

To enable monitoring:

- a. Click the Operational Settings tab.
- b. Select the **Enable Monitoring** check box.
- c. 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 Operations Guide*

- d. Click **Update**.
- e. 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 ManagerLoanReview 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 3-17, but use the configuration parameters in the following table.

**Table 3-4 ManagerLoanReview Business Service**

Configuration Parameter	Value
Business Service Name	ManagerLoanReview
ManagerApprovalService WSDL	Select the managerApproval WSDL you created based on: <i>BEA_HOME\weblogic92\servicebus\examples\src\examples\w ebservices\jws_basic\manager\ManagerApprovalService.ws dl</i>  <b>Note:</b> Select the <b>helloPort</b> port.
EndpointURI	<code>http://&lt;host:port&gt;/mjws_basic_ejb/ManagerSimpleBean</code>

## Summary

When you complete “[Step 1: Prepare the Environment](#)” on page 3-4 to “[Step 5: Create the Business Services](#)” on page 3-17, you have created two business services (NormalLoan and ManagerLoanReview), 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](#)

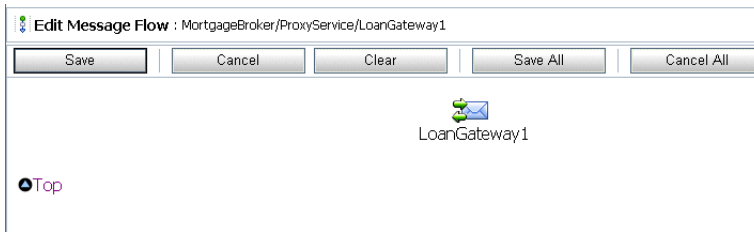
## Configure the Routing Behavior of the Message Flow

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.

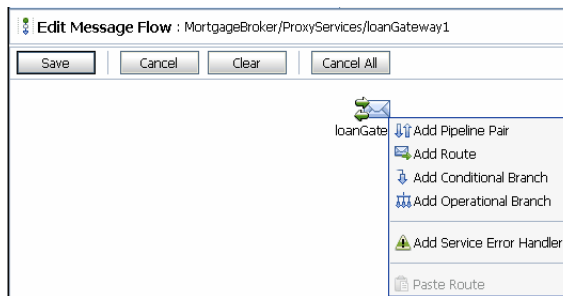
2. In the **Actions** column associated with the **LoanGateway1** proxy service, double click the **Edit Message Flow**  icon. The Edit Message Flow page for the proxy service **LoanGateway1** is displayed as shown in the following figure.

**Figure 3-23 Edit Message Flow for LoanGateway1 Proxy Service**

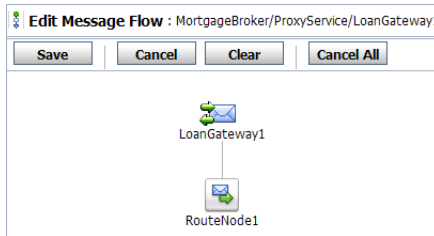


3. Click **LoanGateway1**, and select **Add Route** from the pull-down menu options.

**Figure 3-24 Configure Message Flow for LoanGateway1 Proxy Service**



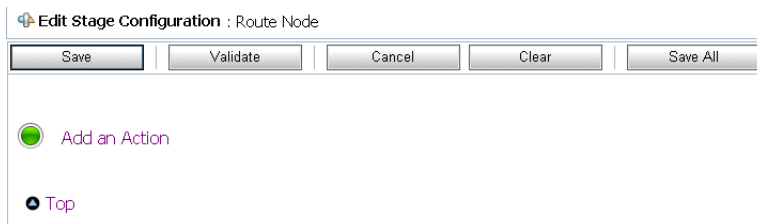
RouteNode1 is added in the configuration page as shown in the following figure.

**Figure 3-25 Add a Route Node**

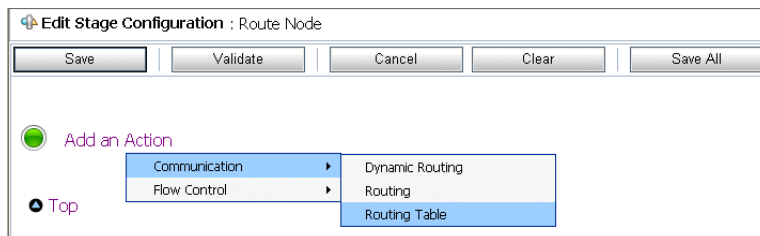
4. Click **RouteNode1** and select **Edit**→**Route** from the drop-down menu options.

The Edit Stage Configuration: Route Node page is displayed as illustrated in [Figure 3-27](#). This page contains a single **Add an Action** link.

**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.

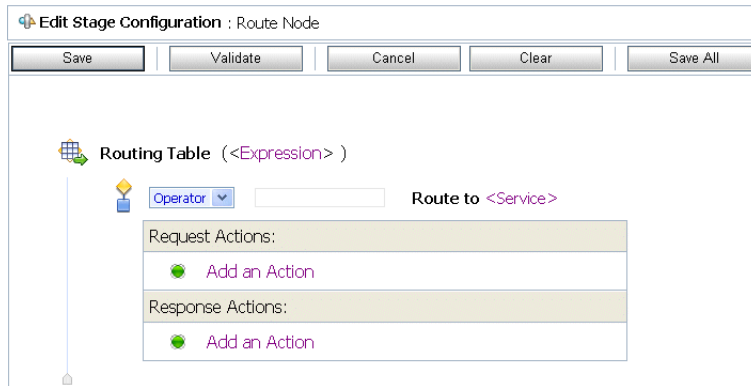
**Figure 3-26 Add an Action Link**

5. Click the **Add an Action** link, and select the **Communication**→**Routing Table** from the pull-down menu options.

**Figure 3-27 Add a Routing Table Element**

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

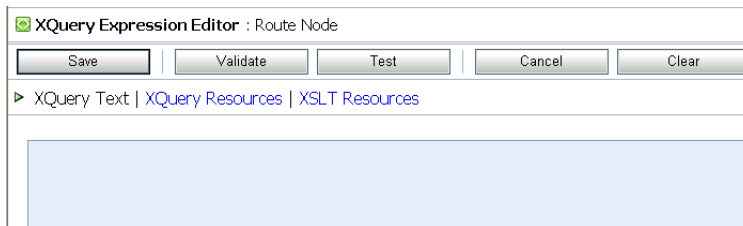
**Figure 3-28 Routing Table Configuration Page**



You must configure the routing table to route messages to business services based on the value of the interest rate element in the incoming message. You can configure content based routing by creating an XQuery expression, using the XQuery Expression Editor.

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

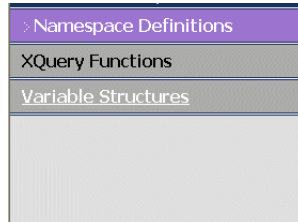
**Figure 3-29 XQuery Expression Editor**



7. Accept the default selection of XQuery Text in the page.
8. In the **Namespace Definitions** pane, click **Variable Structures**.

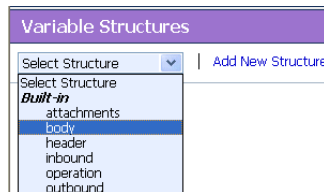


Figure 3-30 Namespace Definitions Pane



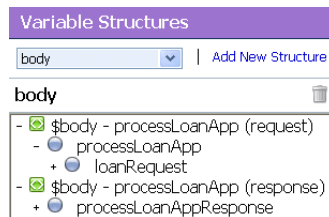
9. The **Variable Structures** pane is displayed. Select **body** from the **Select Structure** drop-down list in the **Variable Structures** pane.

Figure 3-31 Variable Structures Pane



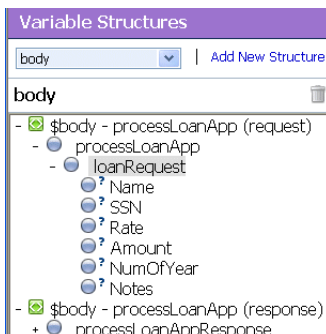
10. A structural representation of the body element is displayed in the **Variable Structures** pane.
11. Click + to expand the **processLoanApp** element, within the **\$body - processLoanApp (request)** element.

Figure 3-32 processLoanApp Element



12. Click + to expand the **loanRequest** element. A graphical representation of the structure of the loanRequest document is displayed. At run time, the proxy service makes its routing decision based on the value in the **Rate** element of the message.

Figure 3-33 loanRequest Element

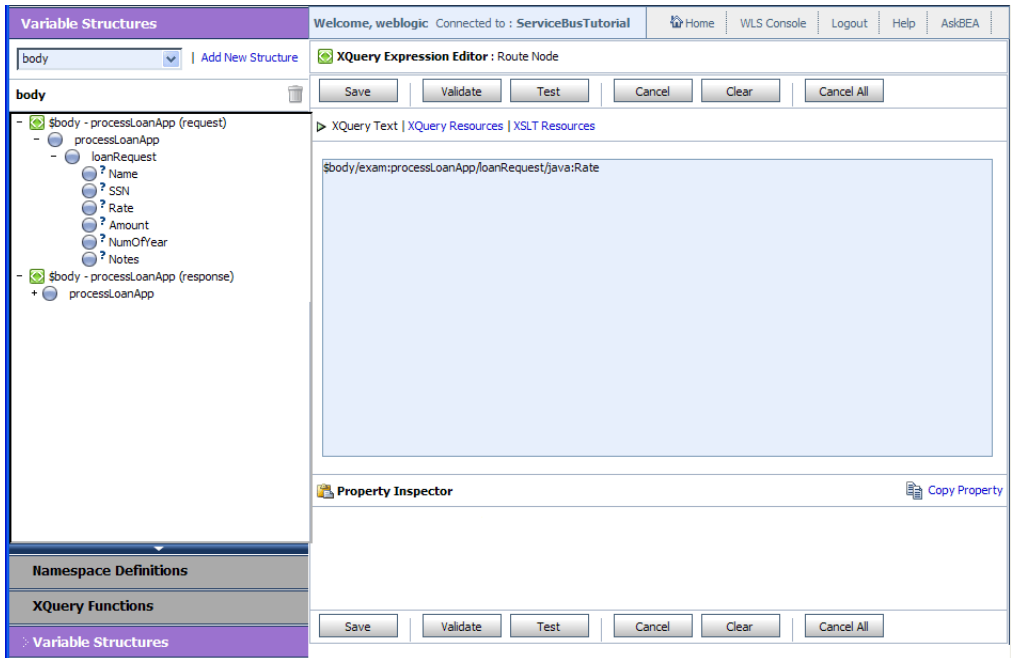


13. Select the circle icon associated with the **Rate** element. Drag and drop it in the XQuery Expression text box to the right. The following XQuery expression is written in the text box:

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

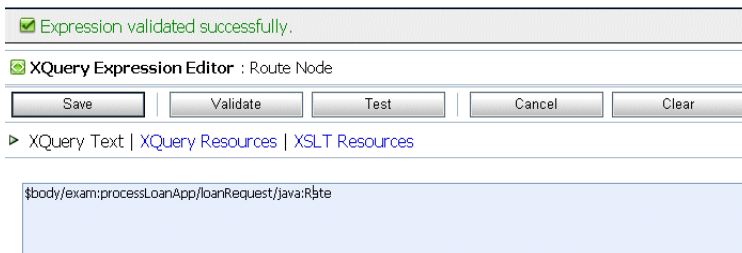
**Note:** The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **Rate** element in the **Variable Structures** pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XQuery expression 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 3-34 XQuery Expression Editor



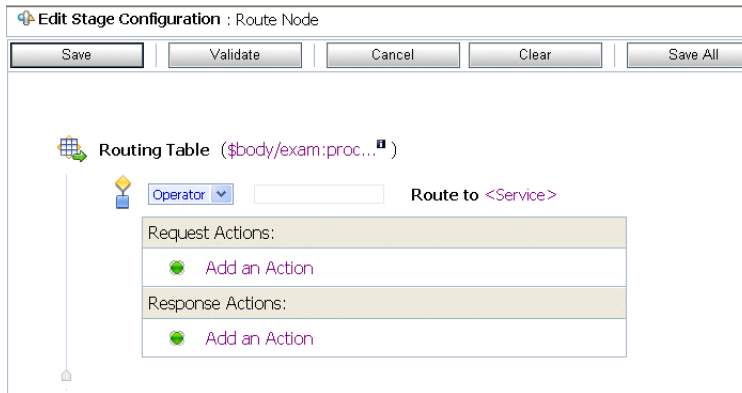
- Click **Validate** to validate the XQuery. It is a good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the **Validate** button.

Figure 3-35 XQuery Validation



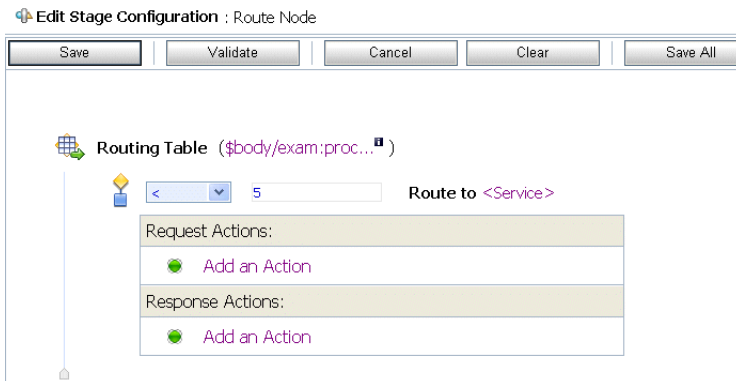
- Click **Save**. The routing table is displayed on the Edit Stage Configuration page. `<Expression>` is replaced by the expression that returns the value of the **Rate** element in the message.

**Figure 3-36 Routing Table View - Expression**



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

**Figure 3-37 Routing Table View - Routing Condition**



17. 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 page is displayed.
  - a. Select the **ManagerLoanReview** business service.
  - b. Click **Submit**.

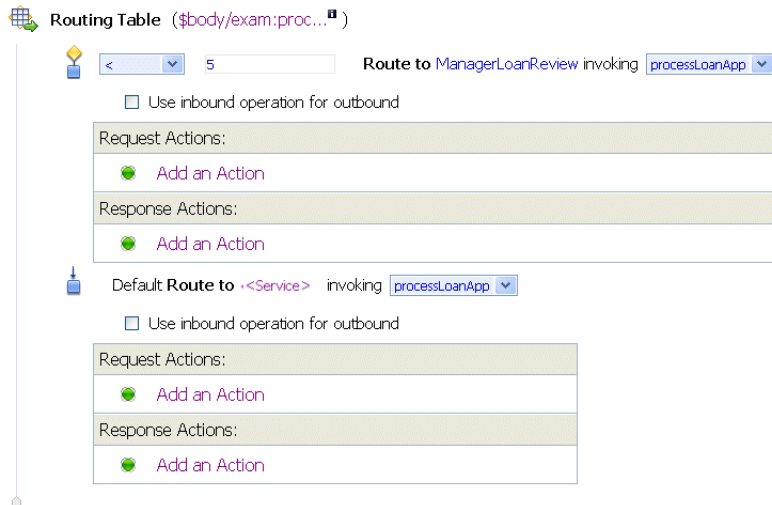
18. In the Routing Table Operation drop-down list, select the **processLoanApp** operation. This is the operation on the **ManagerLoanReview** business service that is invoked at run time if the rate requested in the loan request is less than 5.

You have now defined the case to route the loan application to the **ManagerLoanReview** 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.

**Figure 3-38 Routing Table Default Condition**

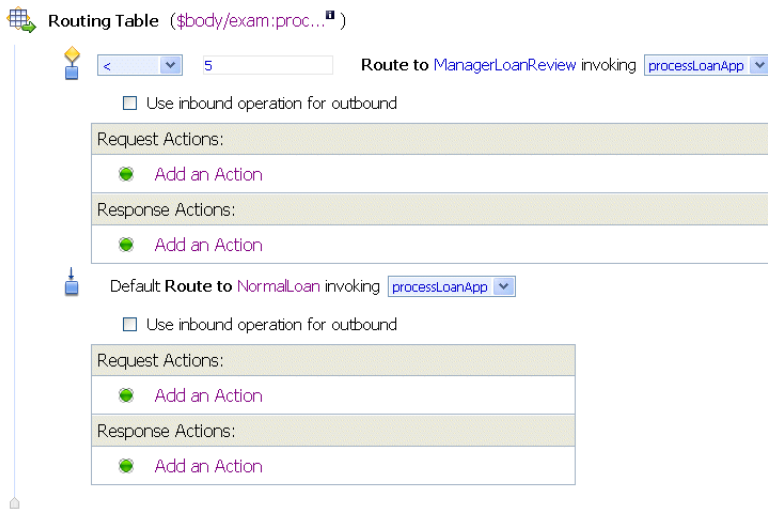


2. In the Default (else) condition, click the **Service** link. The Select Service page is displayed.
  - a. Select the **NormalLoan** business service.
  - b. Click **Submit**.
3. In the **Default** operation drop-down list, select the **processLoanApp** operation for the service.

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 shown in the following figure.

**Figure 3-39 Routing Behavior of the Message Flow**

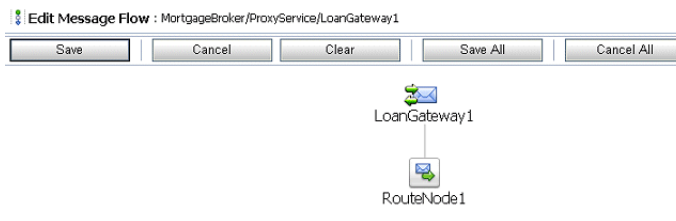


4. Click **Validate** to validate the **RouteNode1** routing configuration.

**Note:** It is a good practice to validate any updates to the routing behavior 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.

**Figure 3-40 LoanGateway1 Proxy Service Message Flow Map**



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.

## Summary

“[Step 6: Configure the LoanGateway1 Proxy Service](#)” on page 3-23 completes the configuration of the run time message routing behavior in the **LoanGateway1** Routing Table. A message is routed to the **ManagerLoanReview** 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. 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 **ManagerLoanReview** business service.
- A value of 5.0 or greater causes the message to be routed to the **NormalLoan** business service.

This section describes how you can test the proxy service for both cases. 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 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 ManagerLoanReview Business Service](#)
- [Testing the NormalLoan Business Service](#)

**Caution:** To run this test correctly, ensure that WebLogic Server is started in the AquaLogic Service Bus 2.6 (ServiceBusTutorial) domain.

### Testing the ManagerLoanReview 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:

## Tutorial 1. Routing a Loan Application

```
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\webservices\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. The interest rate is specified as an input parameter 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.

All the input parameters specified on the `arg line` are described in the following table.

**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



**Table 3-5 Input Parameters**

Parameter	Description
4.9	The requested interest rate
notes	A string that can contain miscellaneous notes

- 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 **ManagerLoanReview** business service processed the request because the interest rate requested was 4.9 (less than 5).

### Testing the NormalLoan Business Service

- Follow the procedure in the [Testing the ManagerLoanReview Business Service](#) section. Edit the `build.xml` file in the following directory:

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

- 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</b></i></b>
LOAN APPLICATION PROCESSING SERVICE
```

```
java] Rate: 5.1
```

```
BUILD SUCCESSFUL
```

The response message indicates that the **Normal Loan** business service processed the request because the interest rate requested was 5.1.

## Tutorial 1. Routing a Loan Application

# Tutorial 2. Transforming a Loan Application

Data transformation is the mapping of data from one format to another, to make information compatible in heterogeneous system environments. AquaLogic Service Bus can be configured to route and transforms messages when necessary, based on specific proxy service configurations.

This section includes the following topics:

- [Prerequisites](#)
- [Tutorial Objectives](#)
- [Definition of the Scenario](#)

## Prerequisites

The following tutorial has to be completed before beginning this tutorial:

- [“Tutorial 1. Routing a Loan Application” on page 3-1.](#)

## Tutorial Objectives

This 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 perform the following tasks:

- Configure content-based routing.

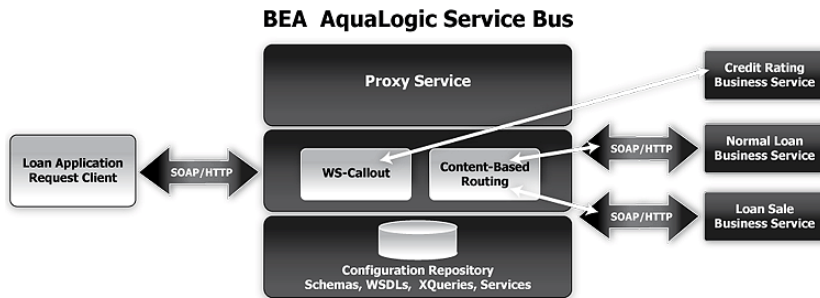
- Use an XPath expression to change the content in the message body.

## 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. The application is then 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.

The following figure 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



## 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 **CreditRating** business service 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 **MortgageBroker** project folder and the directory structure you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in the following table.

**Table 4-1 Routing a Loan Application Tutorial Resources**

Resource Name	Description
creditRating, loanSale	These are the WSDL resources
NormalLoan, LoanSaleProcessor, CreditRating	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 perform the following tasks:

- Import additional WSDL resources
- Register new business services and proxy services
- Configure the routing behavior for the proxy service, by executing the following tasks:
  - 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](#)
- [Step 3: Create the Resources](#)
- [Step 4: Configure Routing for LoanGateway2 Proxy Service](#)
- [Step 5: Configure Routing Request Actions for LoanGateway2 Proxy Service](#)
- [Step 6: Configure Routing Response Actions for 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

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
  - BusinessService
  - 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 **creditRating**) before creating the other resources required for this scenario. WSDL resources are the building blocks for creating the business and proxy services.

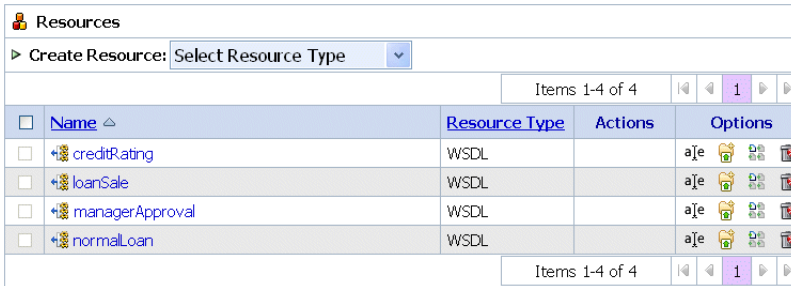
To import the appropriate WSDLs and create the WSDL resources, follow the steps described in “[Create the normalLoan WSDL Resource](#)” section in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#). However for this instance, name your resources and base them on the WSDLs described in the following table.

**Table 4-2 WSDL Configuration Settings**

Resource Name	Location of the WSDL
loanSale	<i>BEA_HOME</i> \weblogic92\servicebus\examples\src\examples\webse rvices\jws_basic\large\LargeLoanPurchasingService.wsdl
creditRating	<i>BEA_HOME</i> \weblogic92\servicebus\examples\src\examples\webse rvices\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 shown in the following figure.

Figure 4-2 WSDL Resources



## 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 3-9](#). For this instance, use the proxy service name and the endpoint URI described in the following table.

Table 4-3 LoanGateway2 Proxy Service Configuration Settings

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

After you complete this step, a summary of the configuration settings for the proxy service is displayed as shown in the following figure. You must review the configuration settings prior to registering the proxy service.




















**Figure 4-3 LoadGateway2 Proxy Service - Configuration Settings Summary**

General Configuration	
Service Name	LoanGateway2
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Proxy Service Provider	
Transport Configuration	
Protocol	http
Endpoint URI	/loan/gateway2
Get All Headers	No
Headers	
HTTP Transport Configuration	
Authentication	None
Message Level Security Configuration	
Custom Authentication	None
Operation Selection Configuration	
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 shown in the following figure.

**Figure 4-4 Proxy Service Resources**

Resources				
Create Resource:		Select Resource Type		
				Items 1-3 of 3
<input type="checkbox"/>	Name	Resource Type	Actions	Options
<input type="checkbox"/>	LoanGateway1	Proxy Service	   aje   	  
<input type="checkbox"/>	LoanGateway2	Proxy Service	   aje  	  

## 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:

- **CreditRating**—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 **NormalLoan** service in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#). To create the **LoanSaleProcessor** and **CreditRating** business services for this scenario, follow the steps described in [“Create the NormalLoan Business Service” on page 3-17](#). You must configure your services using the names, service types, and endpoint URIs described in the following two tables.

**Table 4-4 Configuration Settings for the LoanSaleProcessor Business Service**

Business Service Name	LoanSaleProcessor
Service Type	Select the <b>WSDL Web service</b> as the Service Type option. Select the <b>LoanSale</b> WSDL in the Select a WSDL page. Select the <b>helloPort</b> port in the Select a WSDL Definition page.
Endpoint URI	Ensure that the pre-populated value for the Endpoint URI in the <b>Existing URIs</b> field is <code>http://&lt;host:port&gt;/ljws_basic_ejb/LargeSimpleBean</code> where <code>&lt;host:port&gt;</code> 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 <code>BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties</code> For more information, see <a href="#">“Configure the examples.properties File” on page 2-10</a> .

**Table 4-5 Configuration Settings for the CreditRating Business Service**

<b>Business Service Name</b>	CreditRating
Service Type	Select the <b>WSDL port</b> option. Select the <b>creditRating</b> WSDL in the Select a WSDL page. Select the <b>helloPort</b> port in the Select a WSDL Definition page.
Endpoint URI	Ensure that the pre-populated value for the endpoint URI in the <b>Existing URIs</b> field is <code>http://&lt;host:port&gt;/crejws_basic_ejb/CreditSimpleBean</code> where <code>&lt;host:port&gt;</code> 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 <code>BEA_HOME\weblogic92\samples\servicebus\examples\src\examples.properties</code> For more information, see <a href="#">“Configure the examples.properties File” on page 2-10</a> .

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 shown in the following figure.

**Figure 4-5 Business Service Resources**

<input type="checkbox"/> Name	Path	Actions
<input type="checkbox"/> CreditRating	MortgageBroker/BusinessService	
<input type="checkbox"/> LoanSaleProcessor	MortgageBroker/BusinessService	
<input type="checkbox"/> ManagerLoanReview	MortgageBroker/BusinessService	
<input type="checkbox"/> NormalLoan	MortgageBroker/BusinessService	

## Summary

Upon completing [Step 1: Prepare the Environment](#) to [Step 3: Create the Resources](#) you have created the resources required for this scenario. You have also configured the proxy service with a base configuration. From [Step 4: Configure Routing for LoanGateway2 Proxy Service](#) to [Step 6: Configure Routing Response Actions for LoanGateway2 Proxy Service](#), you will configure the proxy service by adding the routing, transformation, and Service Callout behavior for the loan application messages.

## Step 4: Configure Routing for 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.


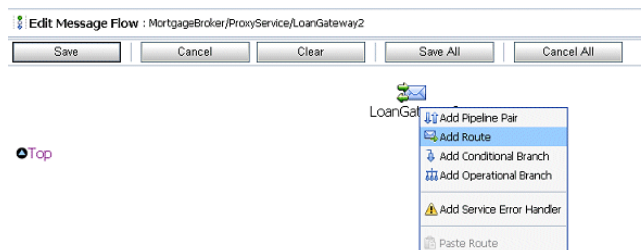
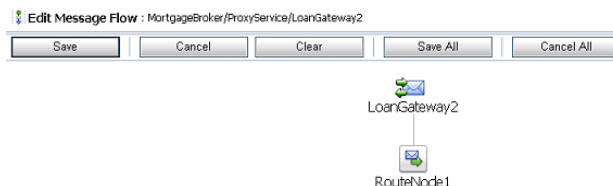
2. In the **Actions** column associated with the **LoanGateway2** proxy service, click the **Edit Message Flow**  icon.

Figure 4-6 Edit Message Flow -LoanGateway2



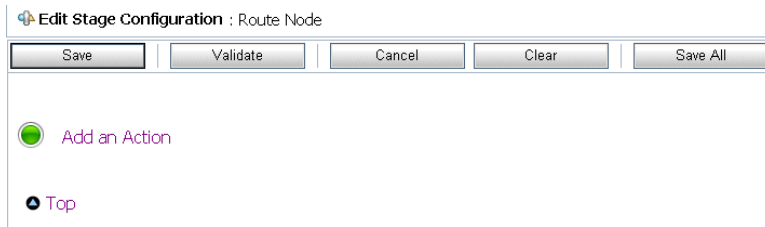
3. Click **LoanGateway2**, then select **Add Route** from the drop-down menu. In the configuration dialog, **RouteNode1** is added.

Figure 4-7 Edit Message Flow - LoanGateway2 RouteNode1



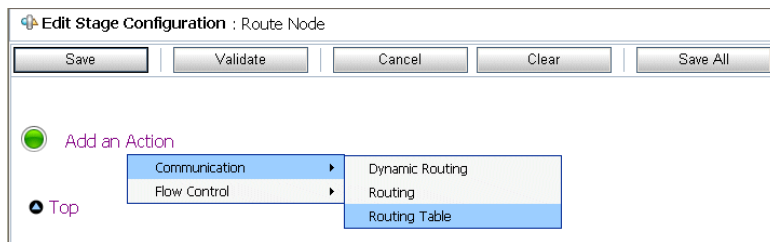
4. Click **RouteNode1** and select **Edit Route** from the drop-down menu. The Edit Stage Configuration page is displayed and contains a single link, **Add an Action**.

**Figure 4-8 Add an Action Link**



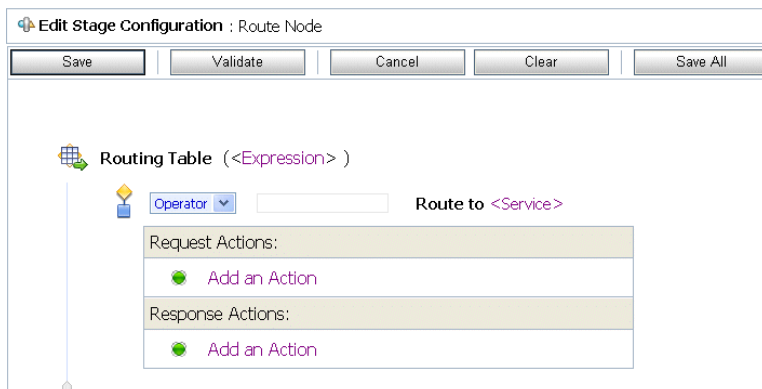
5. Click **Add an Action**, then select **Communication**→**Routing Table** from the pop-up menu.

**Figure 4-9 Add a Routing Table Element**



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

**Figure 4-10 Routing Table Configuration Page**

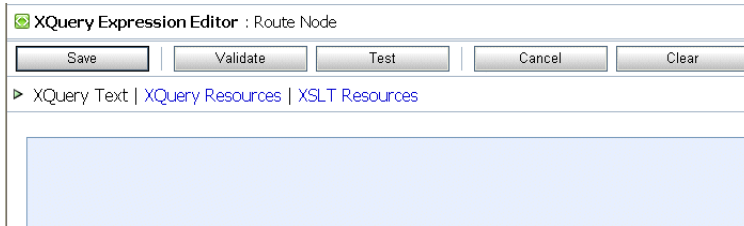


To configure the routing table to route messages to business services based on the **amount** element of the incoming message, you must create an XQuery expression using the XQuery Expression Editor.

## Configure the Routing Expression

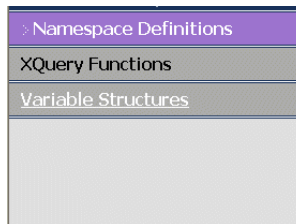
1. In the Routing Table, click the <Expression> link. The XQuery Expression Editor page is displayed.

Figure 4-11 XQuery Expression Editor



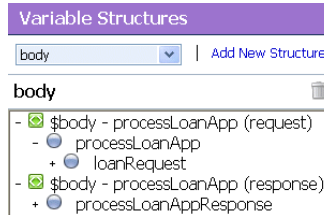
2. Accept the default selection of XQuery Text in the page.
3. In the **Namespace Definitions** pane, click **Variable Structures**.

Figure 4-12 Namespace Definitions Pane



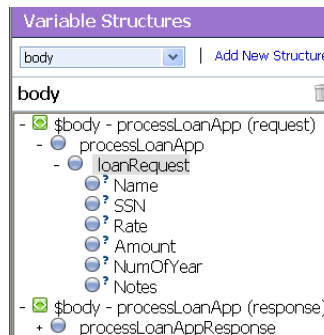
4. Click **Variable Structures** to display the Variable Structures pane. Select **body** from the **Select Structure** drop-down list in the **Variable Structures** pane.


Figure 4-13 Variable Structures Pane



5. A structural representation of the **body** element is displayed in the Variable Structures pane. Click + to expand the **processLoanApp** element, within the **\$body - processLoanApp (request)** element.
6. Click + to expand the **loan request** element. A graphical representation of the structure of the **loan request** document is displayed. At run time, the proxy service makes its routing decision based on the value in the **amount** element of the message.

Figure 4-14 loanRequest Element



7. Click the Node icon  associated with the **Amount** node element, drag and drop it in the XQuery Expression text box. The following XQuery expression is written in the text box.

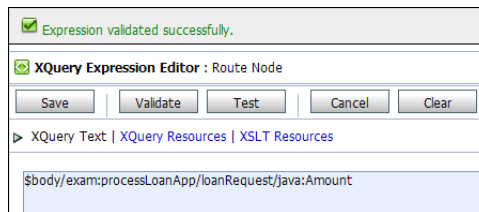
```
$body/exam:processLoanApp/loanRequest/java:Amount
```

**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.

8. 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 shown in the following figure.

**Figure 4-15 Validate Routing Expression**



9. 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.

10. 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.

11. 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 page is displayed.

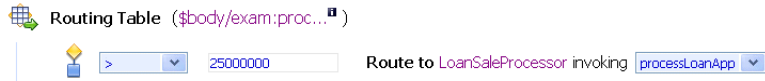
- a. Select the **LoanSaleProcessor** business service
- b. Click **Submit**.

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

This is the operation on the **LoanSaleProcessor** 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 shown in the following figure.



**Figure 4-16 Routing the Loan Application to LoanSaleProcessor**

When a loan application with a loan amount 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. The application is then forwarded to the secondary mortgage company's business service to be processed. [Step 5: Configure Routing Request Actions for 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 Routing Request Actions for 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 Message Transformation](#)

### Configure the Service Callout Input Parameter

A Service Callout is used to send the loan application to the **CreditRating** business service, which returns the credit rating of an applicant. Before adding the Service Callout action, you must configure the Service Callout input parameter by performing the following tasks:

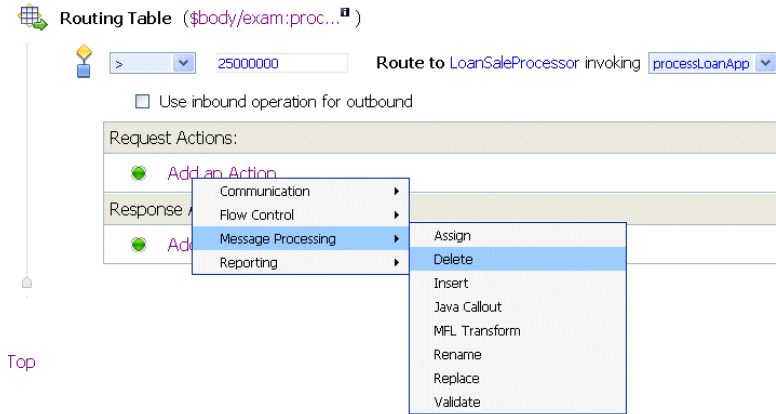
- Delete the “xsi:type” attribute from the message
- Assign an input parameter for the Service Callout action
- Rename the namespace for the Service Callout input parameter

### Add a Delete Action

You must drop the “xsi:type” attribute from the message by adding and configuring a Delete action as follows:

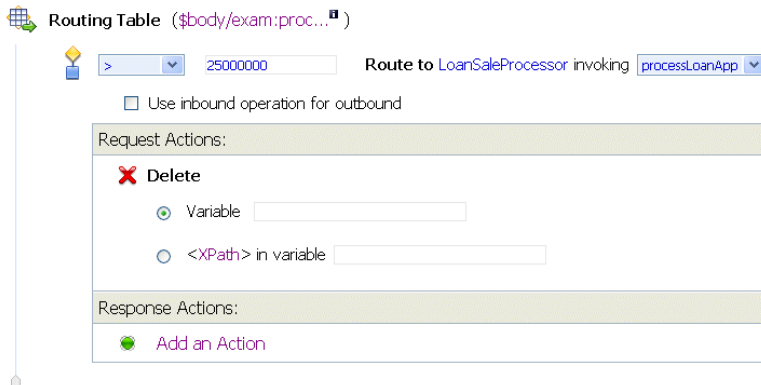
1. In the Routing Table **Request Actions** pane, select **Add an Action**→**Message Processing**→**Delete**.

Figure 4-17 Add a Delete Action



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

Figure 4-18 Initial Delete Action

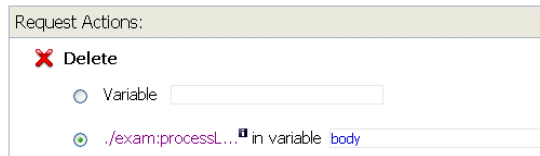


2. Select the **<XPath> in variable** option, then click **<XPath>**. The XPath Expression Editor page is displayed.
  - a. In the XPath Expression text box, enter the following expression:  
`./exam:processLoanApp/loanRequest/@xsi:type`
  - b. Click **Validate**, then **Save**. You will return to the Edit Stage Configuration page.
3. Enter **body** in the variable text field of the **<XPath> in variable** option in the **Delete** action.

The configuration for the **Delete** action is therefore:


Delete `./exam:processLoanApp/loanRequest/@xsi:type` in variable `body` as shown in the following figure.

**Figure 4-19 LoanGateway2 Routing Request Delete Action**



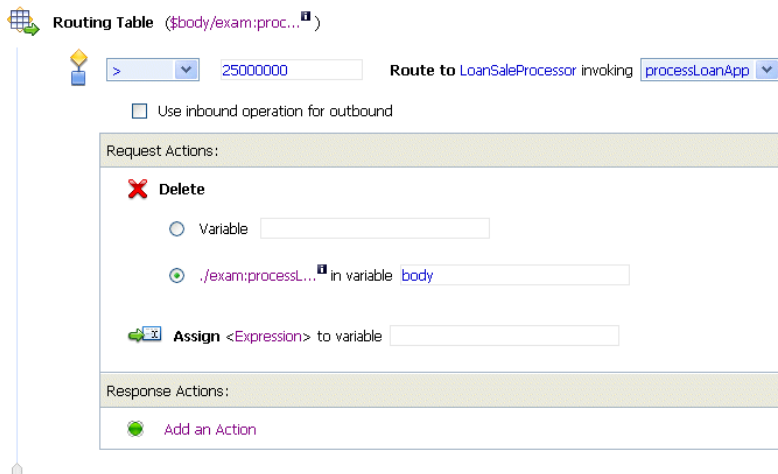
### Assign a Service Callout Input Parameter

Next, you must assign an input parameter for the Service Callout action as follows:

1. Click on the **Delete** icon  in the **Request Actions** pane of the Routing Table.
2. Select **Add an Action**. Then select **Message Processing**→**Assign**.

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

**Figure 4-20 Initial Assign Action**

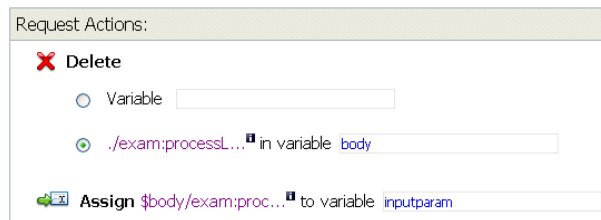


3. Click the `<Expression>` link. The XQuery Expression Editor page is displayed.
4. Accept the default selection of **XQuery Text** in the page.

5. Navigate to the **Variables Structures** pane. Select **body** from the **Variables Structures** drop-down list.
6. Expand the element list. Drag and drop the **loanRequest** element into the XQuery expression text box. The following XQuery expression is written in the text box:  

```
$body/exam:processLoanApp/loanRequest
```
7. 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.
8. Enter `inputparam` in the variable text box as shown in the following figure.

**Figure 4-21 LoanGateway2 Routing Request Assign Action**



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

## Rename Input Variable Namespace

Next, you must rename the namespace of the input parameter you have assigned for the Service Callout using the steps below


1. Click on the **Assign**  icon in the **Request Actions** pane of the Routing Table.
2. Select **Add an Action**→**Message Processing**→**Rename**. The **Rename** action configuration fields are added to the Request Actions pane.

Figure 4-22 Initial Input Variable Rename Action

The screenshot shows the 'Request Actions' configuration window. It contains several actions:

- Delete:** A red 'X' icon and the text 'Delete'. Below it are two radio buttons: 'Variable' (unselected) and './exam:processl...' in variable 'body' (selected).
- Assign:** A green arrow icon and the text 'Assign \$body/exam:proc...' to variable 'inputparam'.
- Rename:** A blue 'X' icon and the text 'Rename <XPath> in variable' followed by a text box containing 'inputparam' and the word 'to'. Below this are three radio buttons: 'localname' (unselected), 'namespace' (unselected), and 'localname' (selected) followed by 'and namespace' and an unchecked checkbox.

3. In the **Rename** statement, click the `<XPath>` link. The XPath Expression Editor is displayed.
4. 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`.

5. Click **Save** to save the XPath expression and return to the Edit Stage Configuration page.
6. In the variable text box, enter `inputparam`.

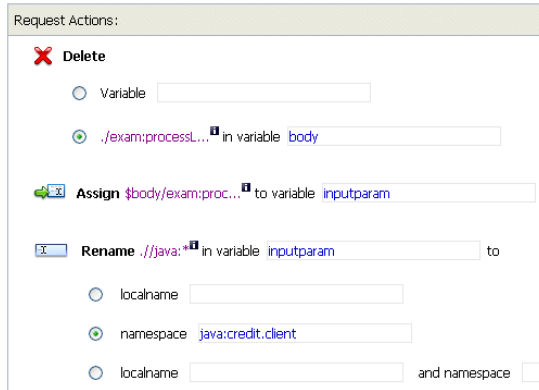
You have completed creating a condition in which the XPath expression finds all the namespaces with the `java` prefix in the `inputparam` context variable. The next step specifies the namespace to replace the namespaces identified by the XPath expression.

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

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


The **Rename** action is displayed as shown in the following figure.

Figure 4-23 LoanGateway2 Routing Request Rename Action



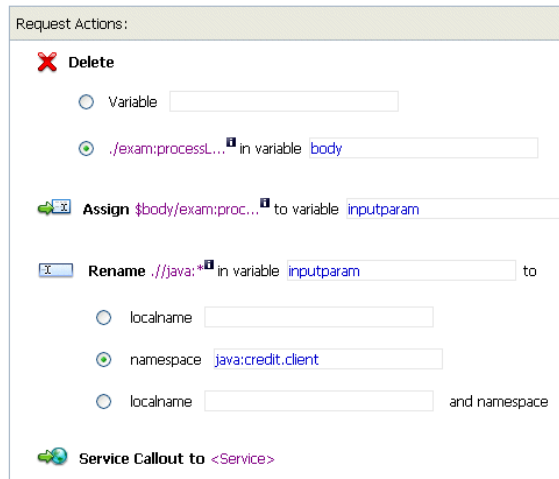
## Add a Service Callout Action

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

1. In the **Request Actions** pane of the Routing Table, click the **Rename**  icon.
2. Select **Add an Action**→**Communication**→**Service Callout**.

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

Figure 4-24 Initial Service Callout Action



3. Click the Service Callout **Service** link. The Select Service page is displayed.
  - a. In the Select Service page, select **CreditRating**.
  - b. Click **Submit**.

The **Service Callout** action must be displayed as shown in the following figure.

**Figure 4-25 Service Callout Action with Configured Service**

Service Callout to CreditRating invoking Operation

Request Actions:

+ Add an Action

Response Actions:

+ Add an Action

4. 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 as shown in the following figure.

**Figure 4-26 Service Callout Action - Configured Operation**

Service Callout to CreditRating invoking processLoanApp

Configure Soap Body  Configure Payload Parameters

Request Parameters:

loanRequest

Response Parameters:

return

SOAP Request Header:

SOAP Response Header:

Request Actions:

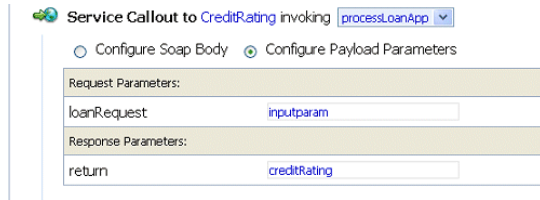
+ Add an Action

Response Actions:

+ Add an Action

- a. In the **Service Callout Request Parameters** section, set the **loanRequest** 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 shown in the following figure.

**Figure 4-27 Service Callout Action - Configured Parameters**



This step completes the Service Callout configuration to invoke the **CreditRating** Web Service. The credit rating returned by this Web Service is assigned to the `$creditRating` context variable.

## Configure Message Transformation

In this section, you will configure the message transformation to match the public contract (interface requirements) of the **LoanSaleProcessor** business 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 Message Namespace”](#) on page 4-22.
- Insert a 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 4-23.

### Rename the Message Namespace

1. Click the **Service Callout** icon and 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.
3. 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`.

4. Click **Save** to save the XPath expression and return to the Edit Stage Configuration page.
5. 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 to replace the namespaces identified by the XPath expression.

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

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

The **Rename** action is displayed as shown in the following figure.

**Figure 4-28 Message Namespace Rename Action**

**Rename** `//java:*` in variable `body` to

localname

namespace `java:large.client`

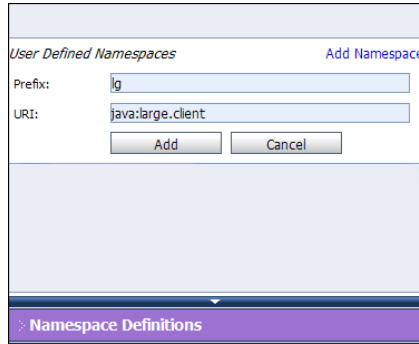
localname  and namespace

### Insert a New Element in the Message

1. Click the **Rename**  icon and 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, navigate to the **User Defined Namespaces** area.
  - 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 shown in the following figure.

**Figure 4-29 User Defined Namespace**



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 **CreditRating** business service. This is the business service to which we configured a Service Callout action in section “[Add a Service Callout Action](#)” on page 4-20).

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

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

6. In the **Insert** action, 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

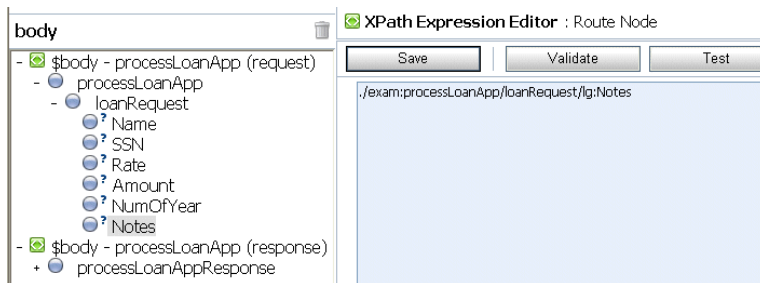
```
./exam:processLoanApp/loanRequest/java:Notes
```

**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.

- In this statement, replace the namespace **java** with the namespace **lg**

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

**Figure 4-30 Configure Outbound Message**

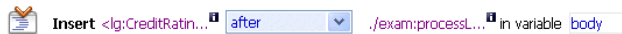


- Click **Validate**, then **Save** to return to the Edit Stage Configuration page.

- 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 must be displayed as shown in the following figure.

**Figure 4-31 Insert New Element into Message**



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

The configuration of the outbound message must be displayed as shown in the following two figures.

Figure 4-32 Configuration of Outbound Message

Request Actions:

- Delete**
  - Variable
  - `./exam:processL...` in variable `body`
- Assign** `$body/exam:proc...` to variable `inputparam`
- Rename** `./java:*` in variable `inputparam` to
  - localname
  - namespace `java:credit_client`
  - localname  and namespace
- Service Callout to CreditRating** invoking `processLoanApp`
  - Configure Soap Body
  - Configure Payload Parameters

Request Parameters:	
loanRequest	<code>inputparam</code>
Response Parameters:	
return	<code>creditRating</code>

SOAP Request Header:

SOAP Response Header:

Request Actions:

  - Add an Action

Response Actions:

  - Add an Action
- Rename** `./java:*` in variable `body` to
  - localname
  - namespace `java:large_client`
  - localname  and namespace
- Insert** `<lg:CreditRatin...` `after` `./exam:processL...` in variable `body`

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

## Step 6: Configure Routing Response Actions for 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 4-27.
- Revert the namespace to the namespace in the original message: `java:normal.client` as described in [“Add a Rename Action”](#) on page 4-28.

### 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 Routing Table Response Actions.

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:
 

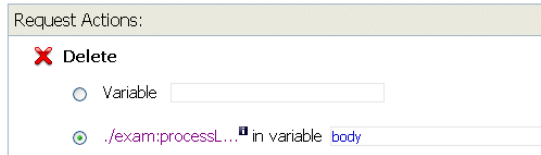
```
./exam:processLoanAppResponse/return/lg:CreditRating
```

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 of the `<XPath>` in **variable** option in the Delete action.

The configuration for the Delete action is therefore:

Delete `./exam:processLoanAppResponse/return/lg:CreditRating` in variable `body` as shown in the following figure.

**Figure 4-33 Response Delete Action**



You have configured the AquaLogic Service Bus to remove the `<CreditRating>` element from the response message, when 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 Message Namespace”](#) on page 4-22).

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

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

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 shown in the following figure.

**Figure 4-34 Add Rename Action**

After these steps are complete, the response action configuration of the proxy service must be displayed as shown in the following figure.

**Figure 4-35 Configuration of Response Actions of Proxy Service**

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.

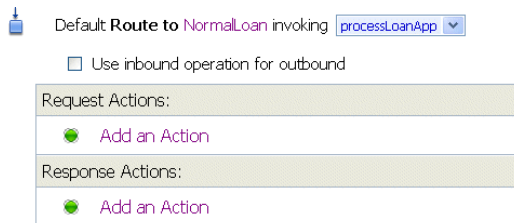
1. Click the Routing Table **Case**  icon, then select **Insert Default Case**.

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

- b. Click **Submit**.
3. In the default case configuration operation drop-down list, select the operation **processLoanApp** as shown in the following figure.

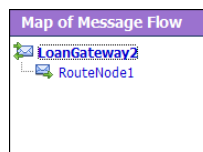
**Figure 4-36 Add Default Routing Case**



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.

The message flow map of the proxy service must be displayed as shown in the following figure.

**Figure 4-37 Map of Message Flow of the LoanGateway2 Proxy Service**



## 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. It 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\webservices\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" />
```

## Tutorial 2. Transforming a Loan Application

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 3-33](#).

5. 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 4-31](#) to edit the `build.xml` file in the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

To test the **NormalLoan** business service, enter a value of 25000000 for the loan amount in the `build.xml` file (see [“Step 7: Test Your Loan Application Routing Configuration” on page 3-33](#)), then save the changes.

2. 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 **NormalLoan** business 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 2. Transforming a Loan Application

# Tutorial 3. Validating a Loan Application

AquaLogic Service Bus proxy services can be configured to validate messages passed between clients and business services. Validation actions can be configured at any stage in the message handling pipeline. However these actions are usually configured in the early stages to eliminate erroneous messages. XQuery conditional expressions can be used to perform explicit validation during message processing. You can configure message validation logic in the following ways:

- By validating 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*.
- By validating the message and reporting errors. For more information, see “Adding an Action” in [Proxy Services: Error Handlers](#) in *AquaLogic Service Bus Console Online Help*.

This section includes the following topics:

- [Prerequisites](#)
- [Tutorial Objectives](#)
- [Definition of the Scenario](#)
- [Steps in This Tutorial](#)

## Prerequisites

The following tutorials have to be completed before beginning this tutorial:

- [“Tutorial 1. Routing a Loan Application” on page 3-1.](#)
- [“Tutorial 2. Transforming a Loan Application” on page 4-1.](#)

## Tutorial Objectives

The objective of this tutorial is to provide steps to configure AquaLogic Service Bus resources to implement message validation and error handling, using the AquaLogic Service Bus Console.

This tutorial 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:

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

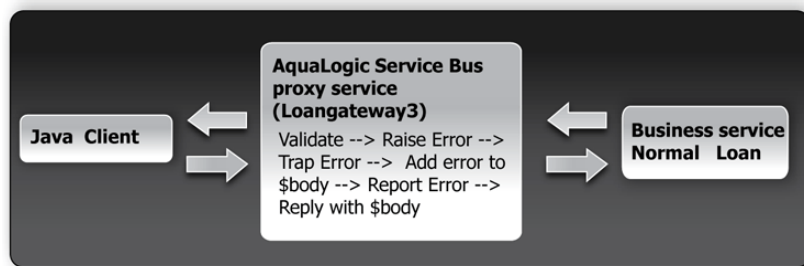
## Definition of the Scenario

A primary mortgage company routes every loan application through an AquaLogic Service Bus proxy service that 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.

The following figure summarizes the logical architecture to support this scenario. It illustrates how AquaLogic Service Bus is positioned in the enterprise system environment to mediate the messaging between the client and the business services.

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



## Overview of the Run Time 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. When the value of the loan duration field is not an integer, the validate action raises an error. The context variable, **\$fault**, is populated with the error details. 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 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 routing is as follows:

- Returned by the response pipeline to the client along with the validation error message  
or
- Forwarded to the **NormalLoan** business service, which returns the following reply:  
The loan application was accepted.

## Required Resources

Use the **MortgageBroker** project folder 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 the following table.

**Table 5-1 Routing a Loan Application Tutorial Resources**

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

## Steps in This Tutorial

In this tutorial, 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, by performing the following tasks:

- 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](#)
- [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

For this tutorial, use the **MortgageBroker** project folder 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
  - BusinessService
  - WSDL

## Step 3: Create the Resources

In this step, you will create the proxy service, **LoanGateway3**. You will use the existing **normalLoan** WSDL resource to validate the content of the message. If the loan application is valid and meets the expected criteria it is routed to the existing **NormalLoan** business service. The **normalLoan** WSDL resource and the **NormalLoan** business service were created in [“Create the normalLoan 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. For creating this proxy service instance, follow the steps described in “[To Create the Proxy Service](#)” on page 3-9, and use the proxy service name and the endpoint URI specified in [Table 5-2](#).

**Table 5-2 LoanGateway3 Proxy Service Configuration Settings**

<b>Proxy Service Name</b>	LoanGateway3
<b>Service Type</b>	Select normalLoan WSDL
<b>Endpoint URI</b>	/loan/gateway3

**Figure 5-2 LoanGateway3 Proxy Service**

The screenshot shows a configuration window titled "Create a Proxy Service - General Configuration (MortgageBroker/ProxyService/)". The "Service Name\*" field contains "LoanGateway3". The "Description" field is empty. Under "Service Type\*", the "Create a New Service" section is active, with "WSDL Web Service" selected. The dropdown menu shows "MortgageBroker/WSDL/normalLoan" and the "port" field contains "helloPort". Other options like "Messaging Service", "Any SOAP Service", and "Any XML Service" are unselected. The "Create From Existing Service" section has "Business Service" and "Proxy Service" options, both with empty input fields and "Browse..." buttons. The "Proxy Service Provider" field is empty with a "Browse..." button. At the bottom are "Next >>", "Last >>", and "Cancel" buttons.

While configuring the proxy service, accept the default settings for the HTTP transport configuration, message level security configuration, and default **SOAP Body Type** operation selection algorithm. When the Create a Proxy Service - Summary page is displayed, you must review the summary of configuration settings prior to registering the proxy service.

When you complete this step, the `MortgageBroker/ProxyService` folder must contain the following proxy services you created in this tutorial and the previous tutorials:

- LoanGateway1

- LoanGateway2
- LoanGateway3

## Step 4: Configure the LoanGateway3 Proxy Service

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

- **Request and response pipelines** for the operations on the proxy service
- **Error handler pipelines** defined for stages, pipelines, and proxy services

Each pipeline can include one or more stages, which in turn include actions.

To configure the behavior of the **LoanGateway3** proxy service, you must complete the following tasks:

- [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.

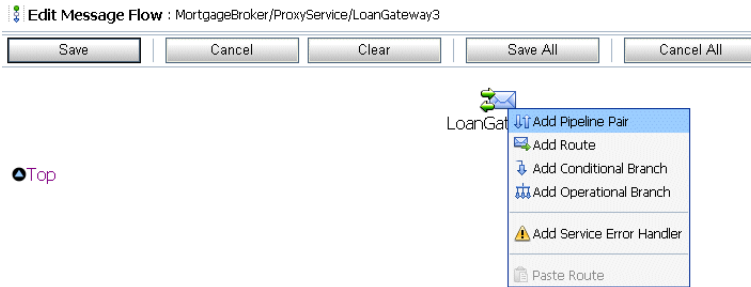
2. In the **Actions** column associated with the **LoanGateway3** proxy service, click the **Edit**

**Message Flow**  icon.

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

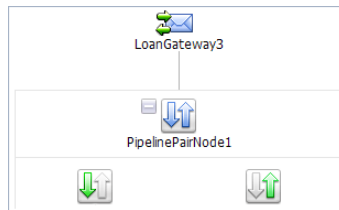
3. Click the **LoanGateway3** node, then select **Add Pipeline Pair**.

**Figure 5-3 Add Pipeline Pair Node**




The **PipelinePairNode1** is created and placed in the message flow. Request and response pipelines are displayed for this node as shown in the following figure.

**Figure 5-4 Add Pipeline Pair**



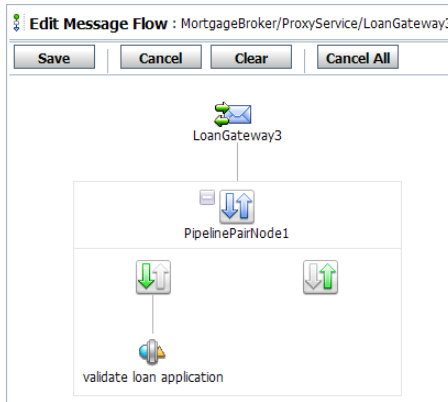
## Configure the Request Actions for the LoanGateway3 Proxy Service

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

1. Click the **Request**  icon, then select **Add Stage**.
2. Click **Stage1**, then select **Edit Name and Description**.
3. Enter **validate loan application** and click **Save**.

The page is displayed as shown in the following figure.

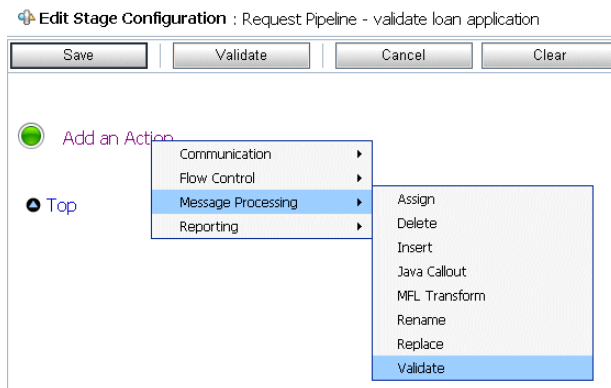
Figure 5-5 Add Stage to Request Pipeline



### Configure an Action to Perform Message Validation

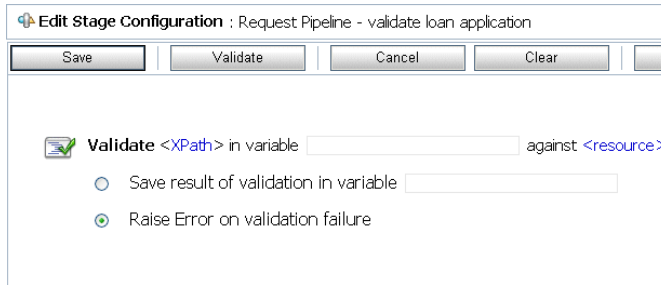
1. To edit the stage, click **validate loan application**, then select **Edit Stage**. The Edit Stage Configuration page is displayed.
2. Select **Add an Action**, then select **Message Processing** → **Validate** from the drop-down list. The Validate action is added to the stage.

Figure 5-6 Add Validate Action



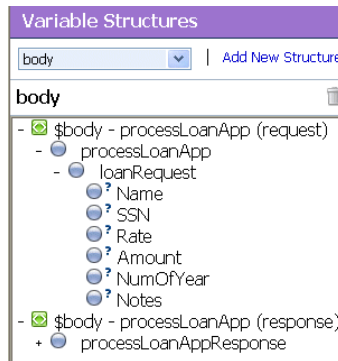
3. In the Validate statement, click the **XPath** link to edit the XPath expression. The XPath Expression Editor page is displayed.

Figure 5-7 XPath Expression Editor



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.

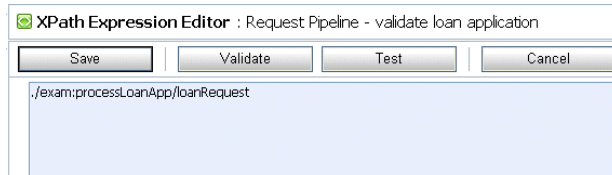
Figure 5-8 Variable Structures Pane



- c. Select the **Node** 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
```

Figure 5-9 XQuery Expression Editor



**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**.

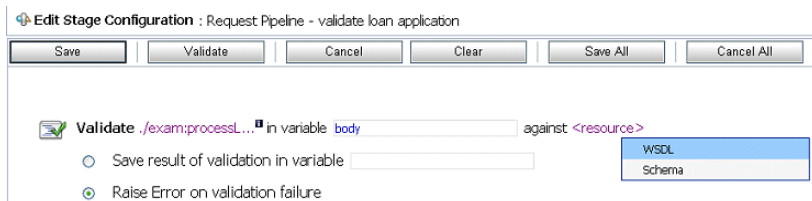
The Edit Stage Configuration page displays `<XPath>` which is replaced by the XQuery expression.

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

9. Click **Resource** and select **WSDL** from the list. The Select a WSDL page is displayed.

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

Figure 5-10 WSDL Resource Selection

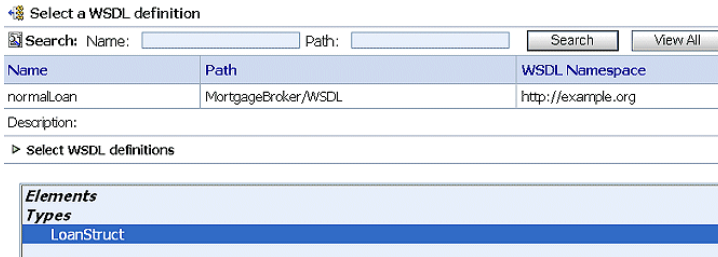


10. In the Select a WSDL page, select the **normalLoan** WSDL resource.

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

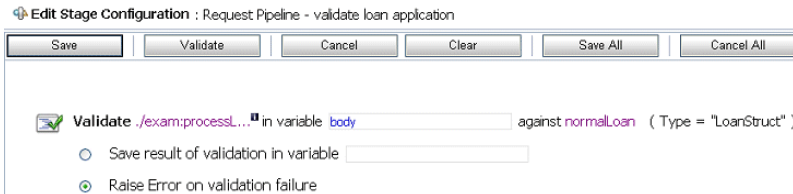
- a. In the Select WSDL Definitions pane, from the Types category, select the WSDL type for this WSDL as **loanStruct**.

Figure 5-11 WSDL Definitions Pane



- b. Click **Submit** to complete the selection.
11. Select the **Raise Error on validation failure** option. The Validate action must be displayed as shown in the following figure.
  12. Click **Save** to save the configuration.

Figure 5-12 Configured Message Validate Action



**Note:** Message validation must be done as early as possible in a message flow. This prevents unnecessary processing of invalid message content. By raising an exception, further message processing is stopped and an error 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, message processing is more modular since the configured validate actions logically break down the message flow. If multiple validations are performed on a message, the validate action is logically expanded.

## Add a Stage Error Handler

The following tasks are performed to configure the Stage Error Handler:

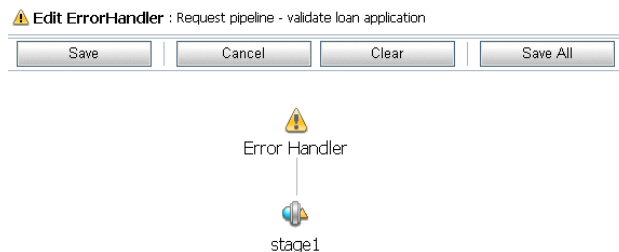


1. Configure error code checking using an **If...Then** action and **Else** actions ([Add an If...Then action to the Stage Error Handler](#)).
2. Add a Replace action within the **If** portion of the **If...Then...** action of the stage. This action replaces the **Notes** element value with the **fault** 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](#)).

### Add Stage Error Handler Element

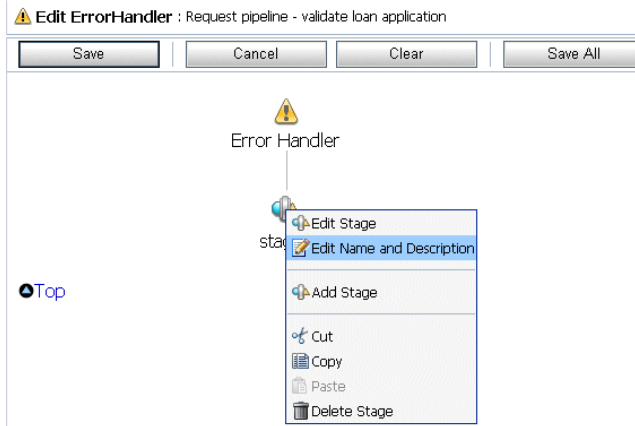
1. Click **validate loan application**, then select **Add Stage Error Handler**. The Edit Error Handler page is displayed.
2. Click **Error Handler**, then select **Add Stage**. The **stage1** node is displayed.

**Figure 5-13 Add Stage Error Handler**



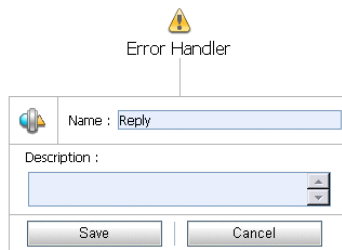
3. Click **stage1** and select **Edit Name and Description** as shown in the following figure.

Figure 5-14 Edit Stage Error Handler



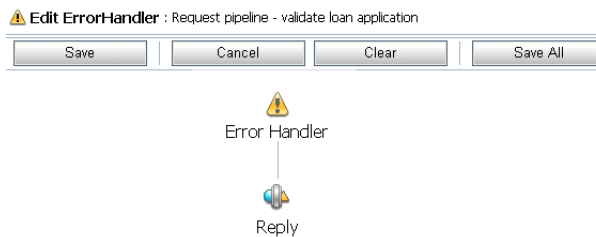
4. Change the stage name to **Reply**, then click **Save**.

Figure 5-15 Edit Name and Description of Stage



The Edit Stage Configuration page is displayed as shown in the following figure.

Figure 5-16 Edit Error Handler Page

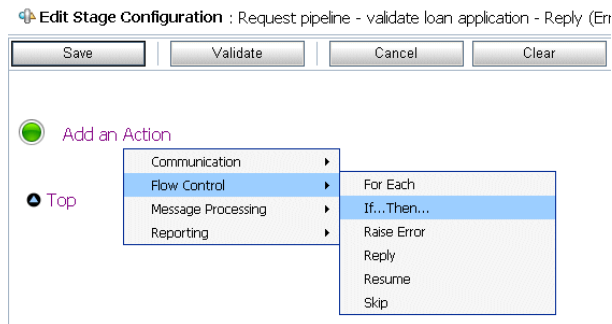


- To edit the stage, click **Reply**, then select **Edit Stage**.

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

- Select **Add an Action**, then select **Flow Control** → **If...Then** to add an **If...Then** action to the Reply stage.

**Figure 5-17 Add Flow Control Action**

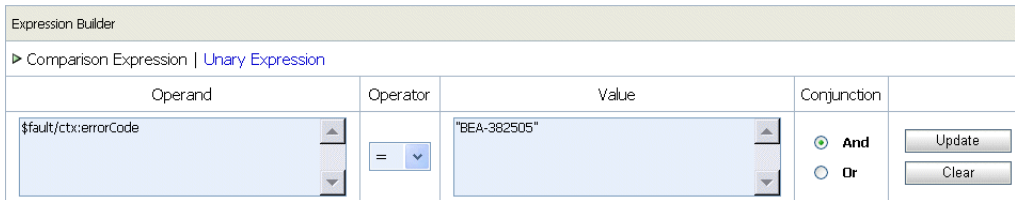


The **If...Then** action is added to the Reply stage.

- Click the **If...Then** action **<Condition>** link to specify the expression that needs to be validated. The Xquery Condition Editor page is displayed.
  - Select the **Builder** pane.
  - In the **Comparison Expression** pane within the **Expression Builder** pane, enter `$fault/ctx:errorCode` in the Operand textbox.
  - Select the = Operator from the drop-down list.
  - Enter "BEA-382505" (including double quotes) in the **Value** pane.
  - Select the radio button for the **And** conjunction.

The Expression Builder Pane is displayed as follows:

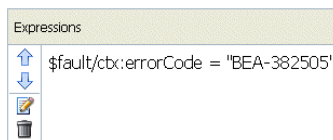
**Figure 5-18 Expression Builder Pane**



f. Click **Add** to add expression to the **Expressions** textbox.

The following expression is displayed in the Expressions textbox:

**Figure 5-19 Expressions Textbox**



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

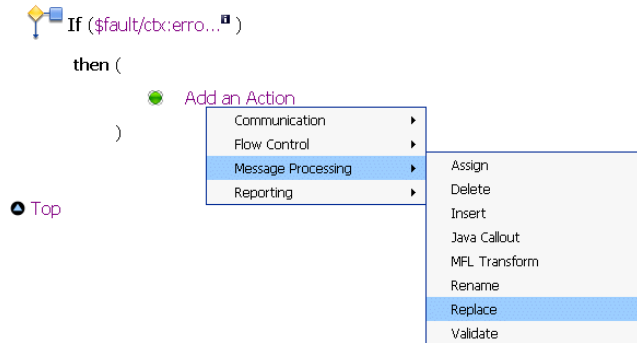
### Add a Replace Action to the Stage Error Handler

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

**Note:** When adding multiple Actions 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.

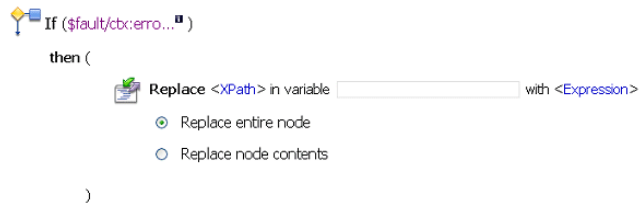
2. Select **Message Processing**→**Replace**.

Figure 5-20 Add a Replace Action



The Replace statement must be displayed as shown in the following figure.

Figure 5-21 Replace Statement



3. In the **Replace** statement, click the `<XPath>` link to edit the XPath expression. The XPath Expression Editor page is displayed.
4. In the **Variable Structures** pane, select **body** from the **Select Structure** drop-down list.


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.

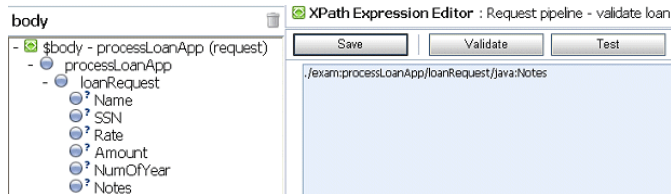
- b. Click + to expand the **loanRequest** element.

## Tutorial 3. Validating a Loan Application

- c. Select the **Node**  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
```

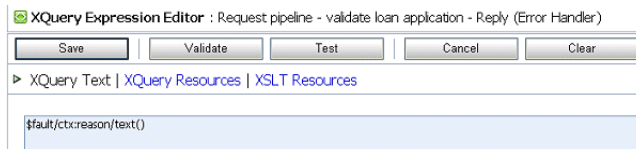
**Figure 5-22 Replace Statement XPath Expression**



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.
8. In the **XQuery Text** pane, enter:  

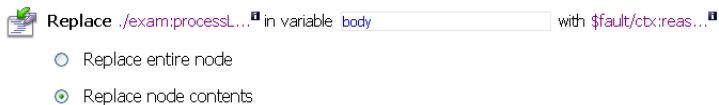
```
$fault/ctx:reason/text()
```

**Figure 5-23 Replace Statement XQuery Expression**




9. Click **Validate**, then **Save**.
10. Select the **Replace node contents** option. You need not replace the entire node.  
The **Replace** action is displayed as shown in the following figure.

**Figure 5-24 Add Error Handler Using Replace Action**

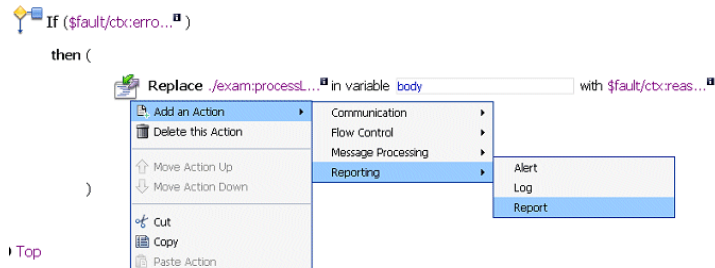


## Add a Report Action to Report the Error

Next, you must 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.

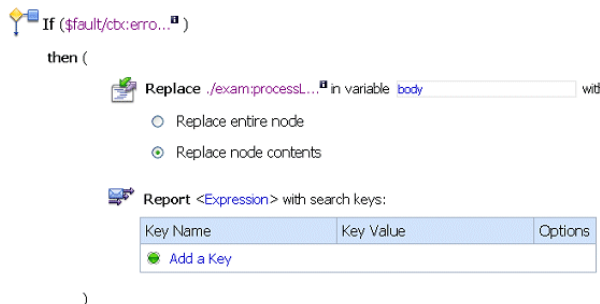
1. Click the **Replace**  icon and select **Add an Action**→**Reporting**→**Report**. The **Report** action is added to the page.

**Figure 5-25 Adding a Report Action**




The Stage Error Handler is displayed as shown in the following figure.

**Figure 5-26 Stage Error Handler with Report Action**



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.

## Tutorial 3. Validating a Loan Application

- 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.

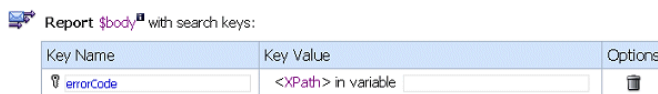
- b. Click **Validate**, then **Save**.

The Edit Stage Configuration page is displayed.


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

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

**Figure 5-27 Report Action Expression Key**



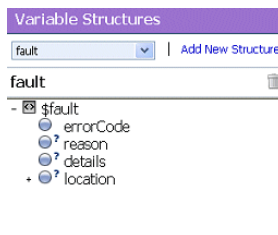
Report `$body` with search keys:


Key Name	Key Value	Options
<input type="text" value="errorCode"/>	<XPath> in variable <input type="text"/>	

- b. In the key value field, click `<XPath>` link. The XPath Expression Editor is displayed.

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

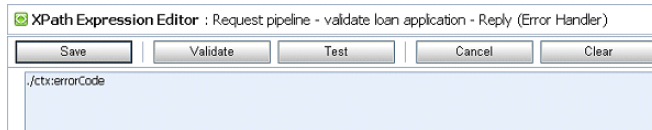
**Figure 5-28 Fault Variable Structure**



- a. Select the **Node**  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.



**Figure 5-29 Report Action ErrorCode Expression**

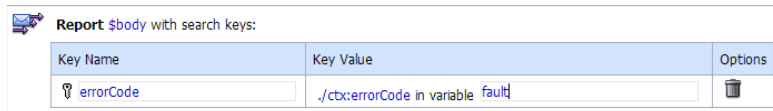
**Note:** The drag and drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the **errorCode** element in the Variable Structures pane. The expression is displayed in the **Property Inspector** palette. Place the cursor in the XPath expression 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**.

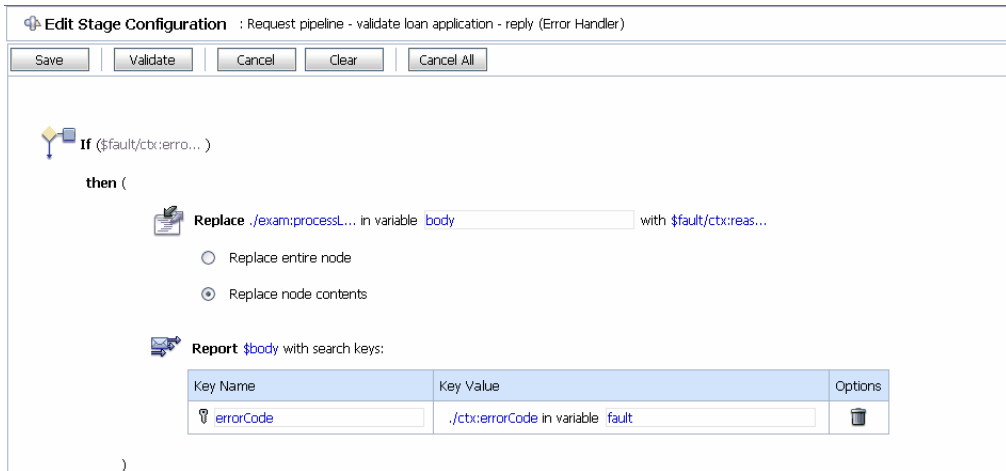
8. In the **Variable** text field, enter **fault**.

The **Report** action must be displayed as shown in the following figure.

**Figure 5-30 Configured Report Action**

The Edit Stage Configuration page is displayed as shown in the following figure.

**Figure 5-31 Request Actions for LoanGateway3 Proxy Service**



### Add an Else Action to the Stage Error Handler

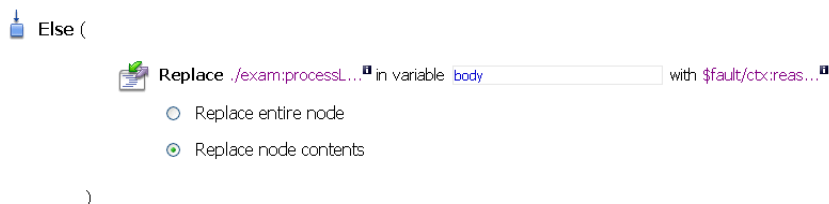
1. Click the **If...Then** icon.
2. Select **Add else Condition** to add an Else action to the Stage Error Handler.

### Add a Replace Action Within the Else Action


1. Select **Add an Action** to add a Replace statement, then select **Message Processing**→**Replace**.
2. Execute step 3 to step 10 in the [Add a Replace Action to the Stage Error Handler](#) section.

The Edit Stage Configuration page is displayed as shown in the following figure.

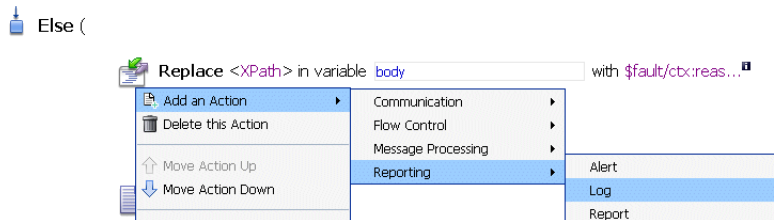
**Figure 5-32 Edit Stage Configuration Page - Else Replace Action**



## Add a Log Action Within the Else Action

1. Click the **Replace**  icon within the **Else** Action.
2. Select **Add an Action** to add a Replace statement, then select **Reporting**→**Log**.

**Figure 5-33 Reporting Log Action**



The Reply Error Handler Else action is displayed as shown in the following figure.

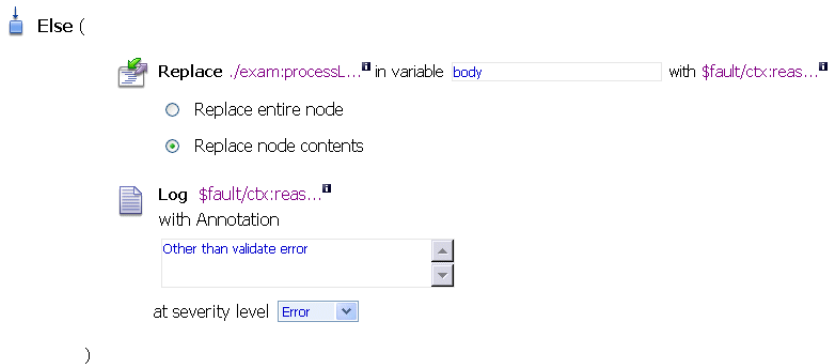
**Figure 5-34 Reporting Log Action**



3. Click the **<Expression>** link to display the XQuery Expression Editor page. Enter `$fault/ctx:reason/text()` in the **XQuery Text** page.
4. Click **Save** to save the expression and return to the Edit Stage Configuration page.
5. Enter `Other than validation error`, in the **With Annotation** text box.
6. Select **Error** from the **At Severity Level** drop-down list.

The completed Else condition must be displayed as shown in the following figure.

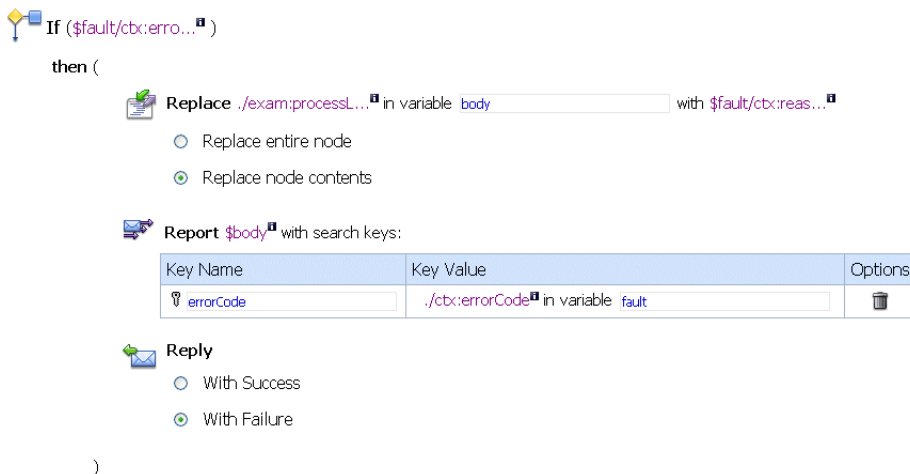
**Figure 5-35 LoanGateway3 Proxy Service Request Else Actions**



### Add a Reply Action to Reply with a Failure Code

1. Click the **Report** icon under the **If...Then** action.
2. Select **Add an Action**, then select **Flow Control**→**Reply**. The Reply action is added to the page.
3. Select the **With Failure** option. The Report Reply Action is displayed as shown in the following figure.

**Figure 5-36 Report Reply Action with Failure Code**



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

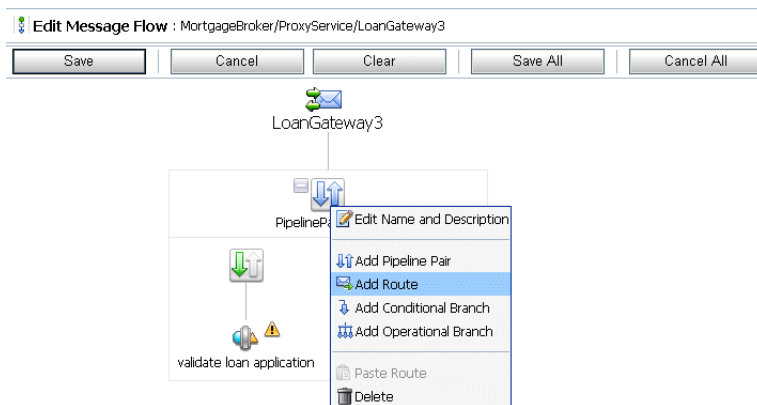
## Step 5: Add a Default Routing Case

You must now add a Route Node to the Pipeline pair and configure it to enable default routing to the **NormalLoan** business service.

### Configure the Route Node

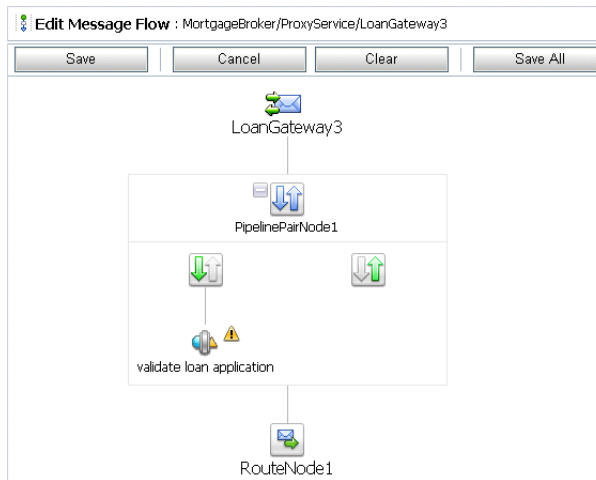
1. Click **PipelinePairNode1**, then select **Add Route**.

Figure 5-37 Pipeline Route Node



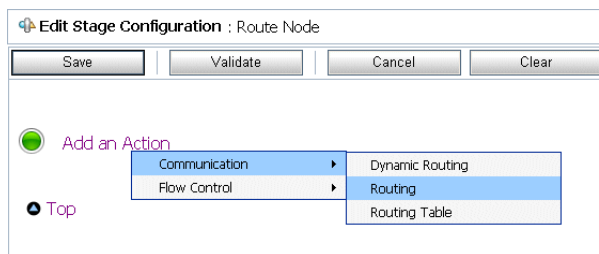
The Edit Message Flow page is displayed as shown in the following figure.

Figure 5-38 Edit Message Flow Page - Pipeline Route Node



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

Figure 5-39 RouteNode1 Communication Routing Action



The **Route to** action is displayed in the Edit Stage Configuration page as shown in the following figure.

Figure 5-40 Route to Statement

Figure 5-40 shows the 'Edit Stage Configuration : Route Node' dialog box. At the top, there are buttons for 'Save', 'Validate', 'Cancel', and 'Clear'. Below these, there is a section titled 'Route to <Service>' with a small icon. Underneath, there are two sections: 'Request Actions:' and 'Response Actions:'. Each section contains a single button labeled 'Add an Action'.

4. In the Route statement, click the <Service> link to define the service to which you want to route messages. The Select Service page is displayed as shown in the following figure.

Figure 5-41 Route to Statement Service

Figure 5-41 shows the 'Select Service' dialog box. It has a search area with 'Name' and 'Path' input fields, a 'Search' button, and a 'View All' button. Below the search area is a table with 7 items. The table has columns for 'Name', 'Path', and 'Resource Type'. The 'NormalLoan' service is selected, indicated by a radio button and a pink background.

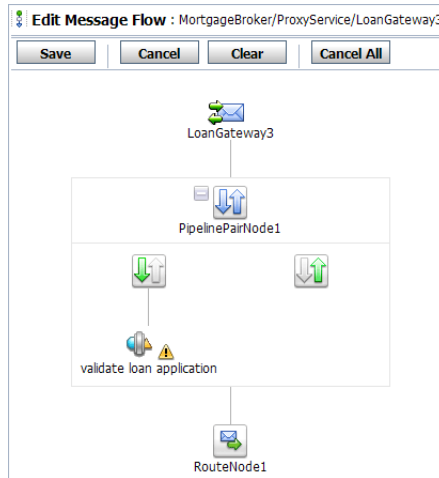
Name	Path	Resource Type
CreditRating	MortgageBroker/BusinessService	Business Service
LoanGateway1	MortgageBroker/ProxyService	Proxy Service
LoanGateway2	MortgageBroker/ProxyService	Proxy Service
LoanGateway3	MortgageBroker/ProxyService	Proxy Service
LoanSaleProcessor	MortgageBroker/BusinessService	Business Service
ManagerLoanReview	MortgageBroker/BusinessService	Business Service
<b>NormalLoan</b>	MortgageBroker/BusinessService	Business Service

At the bottom of the dialog, there are 'Submit' and 'Cancel' buttons, and a pagination control showing 'Items 1-7 of 7'.

- 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`. Click **Save**

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 page is displayed as shown in the following figure.

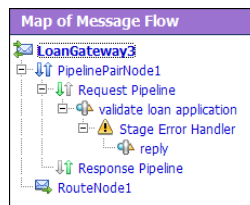
Figure 5-42 LoanGateway3 Proxy Service Message Flow



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 shown in the following figure.

Figure 5-43 Map of Message Flow for LoanGateway3 Proxy Service



## Step 6: 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 must test the configuration. To test the routing configuration, you must change the value of the **Loan duration (in years)** field in the `build.xml` file. Then you must test the routing logic by verifying whether the proxy service behavior changes for different loan values. If the loan value entered is a non integer, a fault must be triggered and an error message displayed in the **Notes** field of the message. If the loan value entered is an integer, the message must be routed to the **NormalLoan** business service. The message returned in the command



window indicates whether a validation exception occurred or the message was routed to the appropriate business service.

This section describes the steps to test the proxy service for the above cases and includes the following test case scenarios:

- [To Test Validation Exception](#)
- [To Test Routing to NormalLoan 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](#).

**Caution:** To run this test correctly, ensure that WebLogic Server is started in the BEA AquaLogic Service Bus 2.6 (ServiceBusTutorial) domain.

### To Test Validation Exception

1. Open a command window. If you have not run the `setEnv.cmd` to set up your environment, you must do so now.

2. Change to the following directory:

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

3. In this directory, run the following command at the command prompt:

```
setEnv.cmd
```

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

4. Change to the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

5. Open `build.xml`. Scroll to the end of the `build.xml` file to the string:

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

a. 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 specifies a loan duration value of 20.5 (a non-integer value) and changes the URI to that of the proxy service running in the tutorial domain.

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
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

- b. Save changes to the `build.xml` file.
6. To test the routing logic of the proxy service, at the command prompt, enter:

```
ant run
```

The following 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
BUILD SUCCESSFUL
```

This response message indicates that a validation exception occurred because the loan duration specified was not an integer.

### To Test Routing to NormalLoan Business Service

1. Navigate to the following directory:

```
BEA_HOME\weblogic92\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

2. Follow the procedure in the [To Test Validation Exception](#) section, to edit the `build.xml` file. In this case, to test the **NormalLoan** business service, you must enter a loan duration value of 20 in the `build.xml` file (see [Table 5-3](#)).

- a. Replace the `<arg line="http://$. . . string with:`

```
<arg line="http://${wls.hostname}:${wls.port}/loan/gateway3 steve 555
2500 20 4.9 notes" />
```
  - b. Save changes to the `build.xml` file.
3. To test the routing logic of the proxy service, on the command prompt, enter:

```
ant run
```

The following response message from the Web Service must be 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 loan request because the loan duration value was 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 Operations Guide](#)

- [Reporting](#) in *AquaLogic Service Bus Console Online Help*

## Monitoring

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 Operations Guide*
- [Monitoring](#) in *AquaLogic Service Bus Console Online Help*