



BEA AquaLogic Service Bus™

Tutorials

Copyright

Copyright © 2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

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

Trademarks or Service Marks

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

Contents

Introduction to the BEA AquaLogic Service Bus Tutorials

What is BEA AquaLogic Service Bus?	1-1
AquaLogic Service Bus Basic Concepts	1-2
Overview of the Tutorials	1-4
Focus of the Tutorials	1-5
How to Use the Tutorials	1-5

Getting Started with the BEA AquaLogic Service Bus Tutorials

About the Scenarios	2-1
How to Use the Tutorials	2-1
Setting Up the Tutorials	2-2
Step 1: Install BEA AquaLogic Service Bus	2-3
Step 2: Change the Configuration Settings	2-4
Step 3: Start BEA AquaLogic Service Bus	2-7
Step 4: Build and Deploy the Business Services	2-7
Step 5: Log in to the AquaLogic Service Bus Console	2-8
Where to Go from Here	2-9

Tutorial 1. Routing a Loan Application

Prerequisites	3-1
Tutorial Goals	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 Your Environment.	3-4
Step 2: Create a Session and Set up a Project	3-4
Step 3: Create the WSDL Resources	3-7
Step 4: Create a Proxy Service.	3-9
Step 5: Create the Business Services	3-16
Step 6: Configure the LoanGateway1 Proxy Service	3-22
Step 7: Test Your Loan Application Routing Configuration	3-27

Tutorial 2. Transforming a Loan Application

Prerequisites	4-1
Tutorial Goals	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 Your Environment.	4-4
Step 2: Create a Session and Set up a Project	4-4
Step 3: Create the Resources	4-5
Step 4: Configure the Routing for the LoanGateway2 Proxy Service.	4-10
Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service	4-13
Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service	4-19
Step 7: Add a Default Routing Case	4-22
Step 8: Test Your Loan Application Routing Configuration	4-23

Best Practices.	4-26
-------------------------	------

Tutorial 3. Validating a Loan Application

Prerequisites.	5-2
Tutorial Goals	5-2
Features Highlighted	5-2
Definition of the Scenario	5-2
Overview of the Process in AquaLogic Service Bus	5-3
Required Resources	5-3
Steps in This Tutorial.	5-4
Step 1: Prepare Your Environment.	5-5
Step 2: Create a Session and Set up a Project	5-5
Step 3: Create the Resources	5-5
Step 4: Configure the LoanGateway3 Proxy Service.	5-6
Step 5: Add a Default Routing Case.	5-15
Step 6: Test Your Loan Application Routing Configuration	5-17
Step 7: Using Reporting and Monitoring	5-20

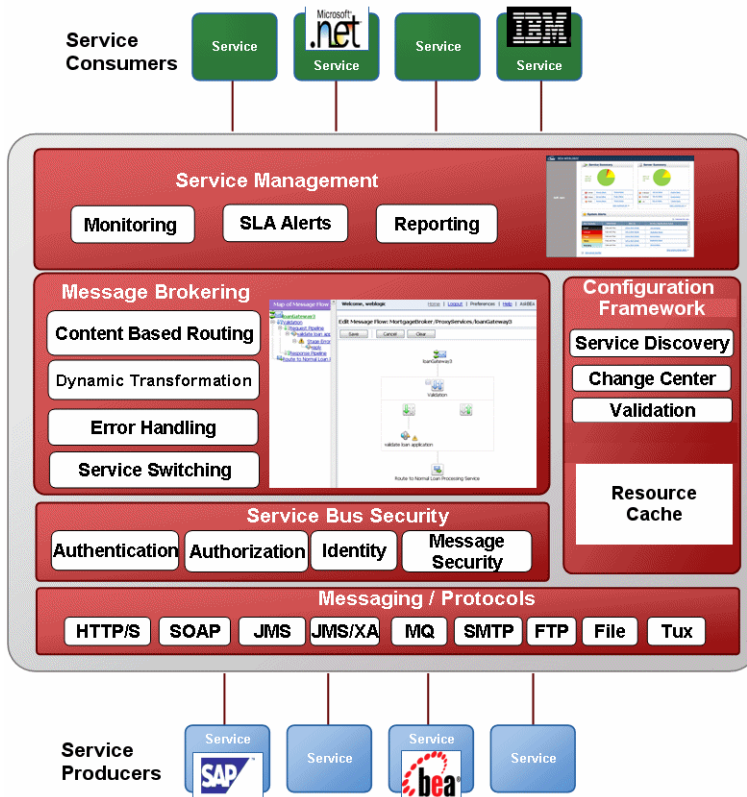
Introduction to the BEA AquaLogic Service Bus Tutorials

What is BEA AquaLogic Service Bus?

BEA AquaLogic Service Bus™ is a configuration-based, policy-driven Enterprise Service Bus (ESB). It is targeted for service-oriented integration, managing Web Services, and providing traditional message brokering across heterogeneous IT environments. AquaLogic Service Bus manages the routing and transformation of messages in an enterprise system to promote seamless application integration. The AquaLogic Service Bus Console enables you to configure a proxy service specific to your needs and to manage your 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. To learn more, see [*BEA AquaLogic Service Bus Concepts and Architecture*](#).

The following figure illustrates the intermediary function of AquaLogic Service Bus, and shows the servers on which it relies.

Figure 1-1 AquaLogic Service Bus Architecture



AquaLogic Service Bus Basic Concepts

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

In this tutorial, you will see the benefits of using AquaLogic Service Bus acting as an intermediary between clients and business services. This tutorial focuses on the configuration of the proxy service. AquaLogic Service Bus is also used to manage your Web services. The configuration functions are separated from the management functions in the AquaLogic Service Bus Console. This separation allows implementations to evolve independently and dynamically, as driven by the needs of the business without requiring costly infrastructure development efforts.

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

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

Business services are remote services or external endpoints that are typically not implemented by the AquaLogic Service Bus server. They are definitions of the enterprise services with which you want to exchange messages. For more information on business services, see [Business Services](#) in *Using the AquaLogic Service Bus Console*.

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

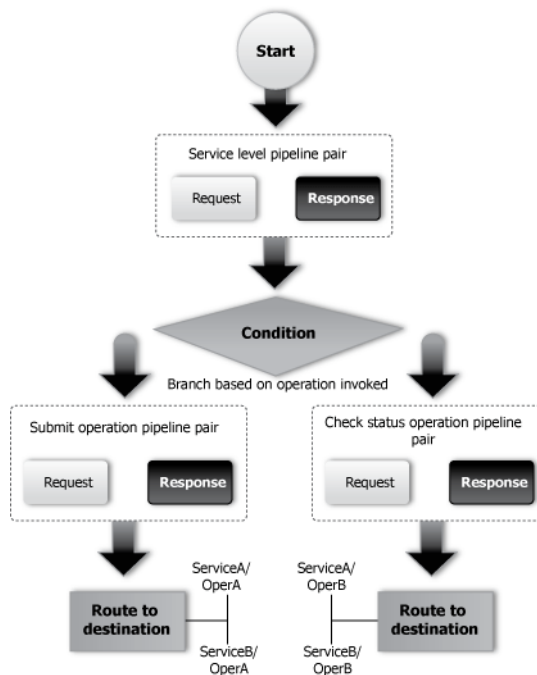
Each pipeline is a sequence of stages containing actions. A **stage** is a user-configured processing step. Messages fed into the pipelines are accompanied by a set of message context variables that contain the message contents and can be accessed or modified by actions in the pipeline stages.

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

Each pipeline consists of a sequence of stages containing actions. However, a single service level request pipeline might optionally branch out into operational pipelines (at most one per operation and optionally a default operational pipeline). 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 shows an example of operation pipelines in a proxy service:

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



For more information, see “Pipeline Pairs” in [BEA AquaLogic Service Bus Concepts and Architecture](#). For information about how to use the WebLogic Service Bus Console to configure proxy services and other resources, see [Using the AquaLogic Service Bus Console](#).

Overview of the Tutorials

These tutorials show how AquaLogic Service Bus is used to route a loan application to appropriate business services based on the configurations set in the proxy service. The AquaLogic Service Bus Console is comprised of JSP-based portlets that support the configuration and design of the proxy service message flows and the associated resources.

Focus of the Tutorials

These tutorial highlights 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. Working through the tutorial you will:

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

How to Use the Tutorials

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

- Provide you with a good understanding of the basic concepts underlying BEA AquaLogic Service Bus.
- Show you how to navigate the AquaLogic Service Bus Console and demonstrate the capabilities of the Console.
- Provide the basic steps that will help you to learn to configure more complicated scenarios.

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

- [Tutorial 1. Routing a Loan Application](#) is based on scenarios that demonstrate how AquaLogic Service Bus facilitates the routing of messages within an enterprise.
- [Tutorial 2. Transforming a Loan Application](#) describes how to configure your existing AquaLogic Service Bus configuration to address the new requirement of adding a transformation stage.
- [Tutorial 3. Validating a Loan Application](#) describes how a BEA AquaLogic Service Bus proxy service can be configured to validate a message passing from a client to a business service.

Document Conventions

The following conventions are used throughout this document:

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

Related Documents

- [*Introducing the BEA AquaLogic Service Bus Examples*](#)
- [*BEA Products Installation Guide*](#)
- [*BEA AquaLogic Service Bus Concepts and Architecture*](#)
- [*AquaLogic Service Bus Console Online Help*](#)

Getting Started with the BEA AquaLogic Service Bus Tutorials

In this section, you learn how to set up your environment, configure your resources, and start BEA AquaLogic Service Bus so that you can create the solutions that implement the business cases presented in the tutorial. You must complete the steps in this section before you start the other tutorials.

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

About the Scenarios

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

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.

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

The following tutorials are available:

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

Setting Up the Tutorials

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

`BEA_HOME\weblogic91\samples\servicebus\examples`

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

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

Filename...	Description
<code>/src/setEnv.cmd</code>	This file sets the environment properties.
<code>/src/examples.properties</code>	This file contains business service build and deployment properties.

Filename...	Description
For each business service:	
<i>BEA_HOME</i> /weblogic91/samples/servicebus/examples/src/examples/webservices/jws_basic/normal	The files contained in this folder are those required by the normal business service. The same files structure exists for the other three business services used in this tutorial (manager, large, and credit).
SimpleBean.java	The jws-181 Web service definition.
LoanStruct.java	The message definition.
build.xml	The build script is used to build each of the services using ant.
*.wsdl	A Web Service Definition Language (WSDL) file defines each of the services.
/client/Main.java	The Java client with which each of the business services interacts.

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

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

Step 1: Install BEA AquaLogic Service Bus

1. Install AquaLogic Service Bus. For instructions on how to do this, see the [BEA Products Installation Guide](#).
2. Create a new AquaLogic Service Bus domain in which to develop the tutorials. The Configuration Wizard guides you through the process of creating a new domain using the configuration templates.

For this tutorial, in the Configuration Wizard, make sure that you:

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

- Select **Development Mode** and the default PointBase database.
- Create a domain named `ServiceBusTutorial`.

For instructions on how to use the configuration wizard, see [Creating WebLogic Domains Using the Configuration Wizard](#).

Step 2: Change the Configuration Settings

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

`BEA_HOME\weblogic91\samples\servicebus\examples`

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

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 running 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> .

Edit the `examples.properties` File

To deploy the business services to your running tutorial domain, you must modify the properties responsible for specifying the WebLogic Server behavior. You do so in the *WebLogic configuration* section in the following file:

`BEA_HOME\weblogic91\samples\servicebus\examples\src\examples.properties`

To Modify the `examples.properties` File

1. Change the **hostname** to the name of the machine on which your `ServiceBusTutorial` domain is running.
1. Change the **port number** to the port number of the tutorial domain.
2. Enter the **username** you specified when creating the domain.
3. Enter the **password** you specified when setting up the domain.

4. Change the server name to the name you specified when you created the domain (if you specified a name other than the default name (AdminServer) assigned by the Configuration Wizard).

Listing 2-1 Fragment of examples.properties File

```
### WebLogic configuration ###
-- Configure per your WLS installation.
wls.hostname=localhost
wls.port=7001
wls.username=weblogic
wls.password=weblogic
wls.server.name=AdminServer
```

Edit and Run the setEnv.cmd File

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

`BEA_HOME\weblogic91\samples\servicebus\examples\src`

The new path is:

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

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

To Run the setEnv Script

Windows:

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

`BEA_HOME\weblogic91\samples\servicebus\examples\src\examples`

3. In the examples directory, on the command line, run the following command:

```
..\setEnv.cmd
```

UNIX:

```
% cd BEA_HOME/weblogic91/samples/servicebus/examples/src/examples
```

```
% ../setEnv
```

where *BEA_HOME* represents the directory in which you installed AquaLogic Service Bus.

Edit the setDomainEnv.cmd File—Windows 2000 Users Only

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

1. Edit the setDomainEnv.cmd file located in the Service Bus domain installation location, *BEA_HOME*\user_projects\domains\ServiceBusTutorial\bin\
2. In the setDomainEnv.cmd file search for the instruction that sets the PRE_CLASSPATH. The instruction looks like:

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

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

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

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

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

5. Save the changes and close setDomainEnv.cmd.

6. Run the `setDomainEnv.cmd`.

- a. `cd` to `BEA_HOME\user_projects\domains\ServiceBusTutorial\bin\`
- b. Run `setDomainEnv.cmd`

Step 3: Start BEA AquaLogic Service Bus

Start AquaLogic Service Bus using one of the following methods:

- From the Windows Start menu, select

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

Where *domain_name* represents the name you assigned your AquaLogic Service Bus domain when you created it.

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

A server command console window, which shows status information about Weblogic Server is displayed.

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

Step 4: Build and Deploy the Business Services

You must start the server before building the Web services. See “[Step 3: Start BEA AquaLogic Service Bus](#)” on page 2-7. After the server is started, you must run a build script to:

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

To Build and Deploy the Business Services

1. In the Command window, ensure that you have run the setEnv script, as described in [“To Run the setEnv Script” on page 2-5](#).

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

```
cd
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic
```

Four business service folders exist in this directory. The following steps describe how to build and run each of the business services.

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

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

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

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

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

A message from the business service is output to the command line. The message indicates whether the Web service was run successfully.

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

Step 5: Log in to the AquaLogic Service Bus Console

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

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

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

```
http://localhost:7001/sbconsole
```

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

Where to Go from Here

After you complete the steps required to set up the tutorials, you are ready to proceed to [“Tutorial 1. Routing a Loan Application” on page 3-1](#), which describes how to configure BEA AquaLogic Service Bus with the resources required for the routing scenario.

Each of the tutorials include instructions for designing and configuring the AquaLogic Service Bus resources, followed by test sections that you can use to test the configurations you complete.

Tutorial 1. Routing a Loan Application

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

This section includes the following topics:

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

Prerequisites

Complete [“Setting Up the Tutorials”](#) on page 2-2.

Tutorial Goals

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

- Importing Web Service Definition Language (WSDL) files
- Registering business services
- Creating a proxy service
- Configuring content-based routing

Features Highlighted

Developing this tutorial exposes you to the following features of BEA AquaLogic Service Bus:

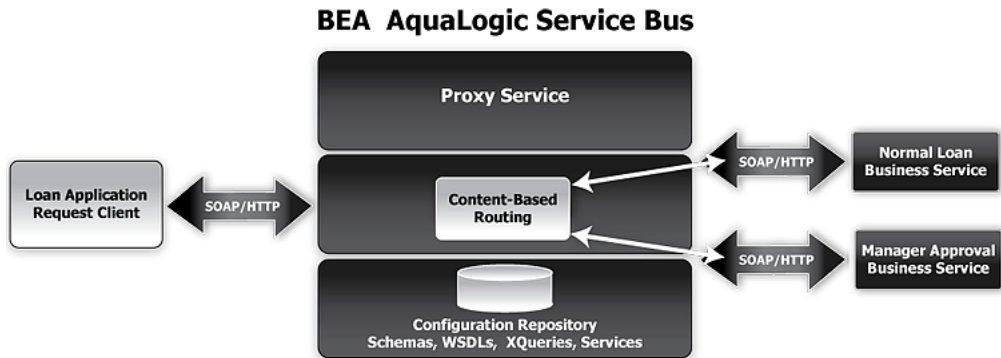
- The AquaLogic Service Bus Console
- Implementation details of proxy services: message flows and actions
- Editing XQuery expressions using the XQuery Expression Editor
- Working with context variables to manipulate the content of the message

Definition of the Scenario

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

The following figure summarizes the logical architecture to support this scenario. It illustrates where BEA AquaLogic Service Bus fits in your enterprise to mediate the messaging between the clients and the business services for the routing scenario.

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

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

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

Required Resources

The following table lists the resources required to develop and run the *Routing a Loan Application* tutorial.

Table 3-1 Routing Loan Application Tutorial Resources

Resource Name	Description
normalLoan	This is the WSDL resource

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

Steps in This Tutorial

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

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

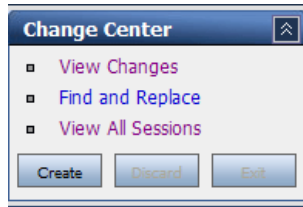
- [Step 1: Prepare Your Environment](#)
- [Step 2: Create a Session and Set up a Project](#)
- [Step 3: Create the WSDL Resources](#)
- [Step 4: Create a Proxy Service](#)
- [Step 5: Create the Business Services](#)
- [Step 6: Configure the LoanGateway1 Proxy Service](#)
- [Step 7: Test Your Loan Application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

Step 1: Prepare 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 [“Setting Up the Tutorials” on page 2-2](#).

Step 2: Create a Session and Set up a Project

You must first start a session in the AquaLogic Service Bus Console. You can view resources and configurations in the AquaLogic Service Bus Console without being in a session, but to edit, change, or delete items or properties in a project, you must be in an active session. The **Change Center** manages sessions in the console as shown in the following figure. Sessions are saved as temporary files until the session is activated.

Figure 3-2 Manage Sessions Using Change Center

The following table provides a summary of the Change Center functionality.

Table 3-2 Description of the Change Center

Click...	To...
Create	Create a new session. You must create a session to make changes to the configuration. You can exit this session using the Exit function, then reenter 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.
Exit	Discontinue the session and save the session state. You can reenter the discontinued session by clicking Edit.
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 good practice to save your configuration changes incrementally.

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

Create the MortgageBroker Project and File Folders

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

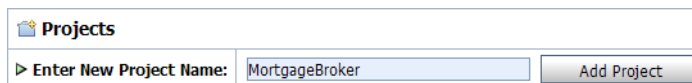
- ProxyService
- BusinessServices
- WSDL

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

To Start a Session and Create a Project

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation panel, select **Project Explorer**.
The Project Explorer pane is opened in the navigation panel and a project page is displayed in 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



Projects	
▶ Enter New Project Name:	<input type="text" value="MortgageBroker"/> <input type="button" value="Add Project"/>

4. Click **Add Project**.

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

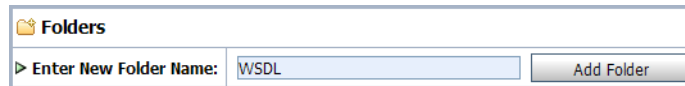
You will create the WSDL folder first, and add a WSDL resource to it. The WSDLs are the basis on which you create the business services and the proxy service. You subsequently create other folders and resources for this scenario.

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 (see [Table 3-2](#)).

Step 3: Create the WSDL Resources

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

Create the normalLoan Service WSDL Resource

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

To Import a WSDL

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation panel, 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** panel, from the **Select Resource Type** drop-down list, select **WSDL**.

Figure 3-5 Select a Resource Type

	Select Resource Type
	Service
	Proxy Service
	Business Service
Folders	
▶ Enter New Folder Name	Interface
	WSDL
	XML Schema
	WS-Policy
Name ▼	Transformation
	XQuery
	XSLT
	MFL File
	Security
	Service Account
	Proxy Service Provider
	Bulk
Resources	Resources from URL
	Zipped Resources
▶ Create Resource:	Select Resource Type ▼

A page that allows you to create a new WSDL resource is displayed as shown in [Figure 3-6](#).

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

- a. Enter the resource name **normalLoan**.
- b. Browse to the location of the WSDL associated with the normal loan process:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal\NormalLoanApprovalService.wsdl
```

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

Figure 3-6 Create a WSDL Resource

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

The resource is now displayed in the resources panel.

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

Create the ManagerApproval Service WSDL Resource

Import the WSDL associated with the ManagerApproval Service. Use the procedure [“To Import a WSDL” on page 3-7](#) to do this using the following parameters.

Table 3-3 ManagerApproval Service WSDL

Configuration Parameter	Value
Resource Name	managerApproval
ManagerApprovalService WSDL	BEA_HOME\weblogic91\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 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 Edit a Proxy Service - General Configuration page is displayed as shown in [Figure 3-8](#).

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

The Select a WSDL browser window is displayed. The proxy service is based on the WSDL resource you originally created, hence you must reference the resource here.

6. In the Select a WSDL window, select the `normalLoan` WSDL.

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

- a. In the Definitions panel, from the Ports category, select the WSDL port for this WSDL (`helloPort`).
- b. Click **Submit** to complete your selection.

Figure 3-7 Select WSDL Definition

Select a WSDL definition

Search: Name: Path:

Name Space:

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

Description: normalLoan


► **Select WSDL definitions**

Ports

helloPort

The WSDL port text field is populated with the port name as shown in the following figure.

Figure 3-8 Create Proxy Service

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

*Service Name

LoanGateway1

Description

*Service Type

Create a New Service

☒ WSDL port

MortgageBroker/WSDL/normalLoan#helloPort

Browse...

☐ WSDL binding

Browse...

☐ Messaging Service

☐ Any SOAP Service

☐ Any XML Service

Create From Existing Service

☐ Business Service

Browse...

☐ Proxy Service

Browse...

Proxy Service Provider

Browse...

Next >>


Finish

Cancel

7. Click **Next** to continue the configuration.

The Edit a Proxy Service - Transport Configuration page is displayed.

Figure 3-9 Transport Configuration of Proxy Service

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

*Protocol

http

*Endpoint URI

Format: /someName

/loan/gateway1

Get All Headers

☐ Yes

☒ No

Header

Add

HEADER

There are no headers configured.

<< Back

Next >>

Finish

Cancel

3-12

BEA AquaLogic Service Bus Tutorials

8. Accept the default protocol, HTTP.

Note: The protocol you select on this page determines the format for the Endpoint URI that you will specify in the next step. HTTP is the default protocol as it is the assumed protocol of 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.
11. Click **Next**.

The Edit a Proxy Service - HTTP Transport Configuration page is displayed.

Figure 3-10 HTTP Transport Configuration of Proxy Service

Because a security policy has not been defined for the Web service, you do not need to change configuration settings on this page.

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

The Edit a Proxy Service - Operation Selection Configuration page is displayed.

Figure 3-11 Operation Selection Configuration of Proxy Service

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


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

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

14. Click **Next**.

A summary of the configuration settings for the proxy service is displayed.

Figure 3-12 Proxy Service - Configuration Settings Summary


 Edit a Proxy Service - Summary (Path - MortgageBroker/ProxyService)	
General Configuration Edit >>	
Service Name	LoanGateway1
Description	
Service Type	Web Service - SOAP (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Proxy Service Provider	
Transport Configuration Edit >>	
Protocol	http
Endpoint URI	/loan/gateway1
Get All Headers	No
Headers	
HTTP Transport Configuration Edit >>	
Basic Authentication Required	DISABLED
Operation Selection Configuration Edit >>	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type
<div><< Back Save Cancel</div>	

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

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

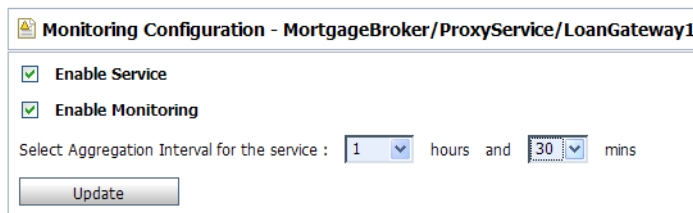
Note: You must always select **Save** prior to activating a session configuration in the Change Center.

The **LoanGateway1** proxy service is displayed in the Resources panel in the **MortgageBroker** project, **ProxyService** folder.

16. If you want to turn on monitoring for the LoanGateway1 proxy service, click the Manage Monitoring icon  associated with the **LoanGateway1** proxy service in the Resources panel.

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

Figure 3-13 Monitoring Configuration of Proxy Service



The image shows a dialog box titled "Monitoring Configuration - MortgageBroker/ProxyService/LoanGateway1". It contains two checked checkboxes: "Enable Service" and "Enable Monitoring". Below these is a label "Select Aggregation Interval for the service :" followed by a dropdown menu set to "1", the word "hours", and the word "and", followed by another dropdown menu set to "30" and the word "mins". At the bottom is an "Update" button.

To enable monitoring:

- Select the **Enable Monitoring** check box.
- 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, specifically the Dashboard.

For more information about using the AquaLogic Service Bus Dashboard and monitoring services, see:

- [Monitoring](#) in the *AquaLogic Service Bus Console Online Help* and
- [Monitoring](#) in the *BEA AquaLogic Service Bus User Guide*.

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

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

Step 5: Create the Business Services

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

In this step, you will define the normal loan processing and manager approval business services:

- [To Create the Normal Loan Business Service](#)
- [To Create the managerLoanReviewService Business Service](#)

To Create the Normal Loan 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 you created for the MortgageBroker project are displayed. Click **Create** to make changes to the session.

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

The Edit a Business Service - General Configuration page is displayed.

Figure 3-14 Create Business Service

Edit a Business Service - General Configuration (Path - MortgageBroker/BusinessService)

*Service Name:

Description:

*Service Type

Create a New Service

☒ WSDL port:

☐ WSDL 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 port**, 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.

5. Click **Next**.

The Edit a Business Service - Transport Configuration page is displayed.

Figure 3-15 Transport Configuration of Business Service

 Edit a Business Service - Transport Configuration (Path - MortgageBroker/BusinessService)

*Protocol

http

Load Balancing Algorithm


none

*Endpoint URI

Format: http://host:port/someService

http://

Add

EXISTING URIS	ACTION
http://localhost:7001/njws_basic_ejb/NormalSimpleBean	

Retry Count

0

Retry Interval

30

<< Back

Next >>

Finish

Cancel

6. Accept the default protocol, HTTP.

7. For the Load balancing Algorithm, select `none` because load balancing is relevant only when a service specifies multiple end points.

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

8. Ensure that the pre-populated value for the Endpoint URI in the **Existing URIs** field is `http://host:port/njws_basic_ejb/NormalSimpleBean`

where `host:port` represents the machine and port on which your AquaLogic Service Bus server is running. The host and port values must be the same as the values specified in `BEA_HOME\weblogic91\samples\servicebus\examples\src\examples.properties`

For more information, see [“Edit the examples.properties File” on page 2-4.](#)

The Endpoint URI of the business service is the endpoint URI on the server on which the service is deployed.

Note: Delete any other invalid endpoint URI in the Existing URIs list.

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

The Edit a Business Service—HTTP Transport Configuration page is displayed.

3-18

BEA AquaLogic Service Bus Tutorials

Figure 3-16 HTTP Transport Configuration of Business Service

Edit a Business Service - HTTP Transport Configuration (Path - MortgageBroker/BusinessService)	
Timeout	<input type="text" value="0"/>
HTTP Request Method	<input checked="" type="radio"/> POST <input type="radio"/> GET
Basic Authentication Required	<input type="checkbox"/>
Service Account	<input type="text"/> <input type="button" value="Browse..."/>
Follow HTTP redirects	<input checked="" type="checkbox"/>
Dispatch Policy	<input type="text" value="default"/> ▼
Request Encoding	<input type="text"/>
Response Encoding	<input type="text"/>
<input type="button" value=" << Back"/> <input type="button" value=" Next >> "/> <input type="button" value=" Finish "/> <input type="button" value=" Cancel "/>	

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

The Edit a Business Service—SOAP Binding Configuration page is displayed.


Figure 3-17 SOAP Binding Configuration of Business Service

Edit a Business Service - SOAP Binding Configuration (Path - MortgageBroker/BusinessService)	
Enforce WS-I Compliance	<input type="checkbox"/>
<input type="button" value=" << Back"/> <input type="button" value=" Next >> "/> <input type="button" value=" Finish "/> <input type="button" value=" Cancel "/>	

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

The Edit a Business Service—Summary page is displayed. You can now review the business service to check that you implemented it correctly.

Figure 3-18 Business Service - Configuration Settings Summary


 **Edit a Business Service - Summary (Path - MortgageBroker/BusinessService)**

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

<< Back

Save

Cancel

12. Click **Save** to accept the configuration settings.
13. If you want to turn on monitoring for the NormalLoan business service, click the Manage Monitoring icon  associated with the **NormalLoan** business service in the Resources panel.

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

- a. Select the **Enable Monitoring** check box.
- b. Enter an aggregation interval—the period over which aggregated statistics related to the service are computed for display in the AquaLogic Service Bus Console, specifically the Dashboard.

For more information about using the AquaLogic Service Bus Dashboard and monitoring services, see:

- [Monitoring](#) in the [AquaLogic Service Bus Console Online Help](#) and
- [Monitoring](#) in the [BEA AquaLogic Service Bus User Guide](#).

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

This step completes the configuration of the NormalLoan business service in the AquaLogic Service Bus Console.

To Create the managerLoanReviewService Business Service

This section describes how to create the business service to which a loan application is routed in the case that the interest rate requested in that loan application is less than 5%.

Follow the same steps as described in the preceding section: “[To Create the Normal Loan Business Service](#)” on page 3-16, but use the configuration parameters in the following table.

Table 3-4 ManagerLoanReviewService Business Service

Configuration Parameter	Value
Business Service Name	managerLoanReviewService
ManagerApprovalService WSDL	<p>Select the managerApproval WSDL you created based on: <i>BEA_HOME\weblogic91\servicebus\examples\src\examples\w ebervices\jws_basic\manager\ManagerApprovalService.ws dl</i></p> <p>Note: Select the helloPort port.</p>
EndpointURI	<code>http://host:port/mjws_basic_ejb/ManagerSimpleBean</code>

Summary

By completing “[Step 1: Prepare Your Environment](#)” on page 3-4 to “[Step 5: Create the Business Services](#)” on page 3-16, you created two business services (NormalLoan and managerLoanReviewService), a proxy service (LoanGateway1) and a WSDL (normalLoan) as resources required for the tutorial scenario.

You configured the proxy service with a base configuration. The next step ([Step 6: Configure the LoanGateway1 Proxy Service](#)) describes how to complete the configuration of the proxy service to add 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 will complete the following for the message flow:

- Add a route node
- Configure an action to do content based routing

To Configure the Routing Behavior of the Message Flow

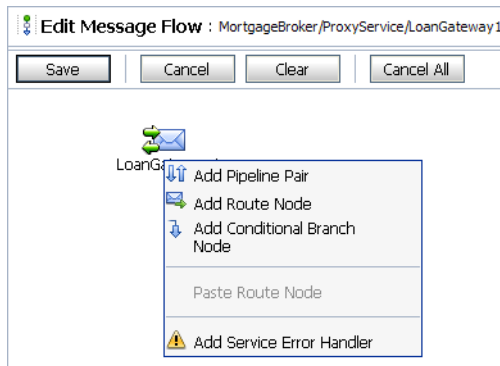
1. In Project Explorer, select the `ProxyService` folder from the `MortgageBroker` project tree.

Note: You must be in a session to edit resources. If you have not already done so, 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 `LoanGateway1` proxy service, click the **Edit Message Flow** icon  .

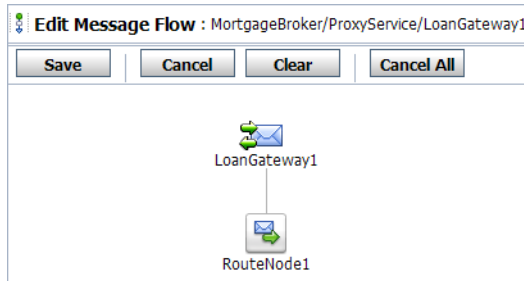
The Edit Message Flow page for the proxy service `LoanGateway1` is displayed.

Figure 3-19 Configure Message Flow for LoanGateway1 Proxy Service



3. Click **LoanGateway1**, then select **Add Route Node**.

In the configuration dialog, `RouteNode1` is added.

Figure 3-20 Add Route Node

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

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

A stage is an element of a pipeline and it is 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.

5. Click **Add an Action**, then select **Routing Table**.

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

You will configure the routing table to route messages to business services based on evaluating the interest rate element of the incoming message. You must create an XQuery expression to do this. You can do so using the XQuery Expression Editor.

6. In the Routing table, click <Expression>. The XQuery Expression Editor page is displayed.
7. Accept the default selection of **XQuery Text** in the page.
8. Click **Variable Structures**. The Variable Structures pane is displayed.
9. From the **Select Structure** drop-down list in the Variable Structures pane, select **body**.

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

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


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

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

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

Tutorial 1. Routing a Loan Application

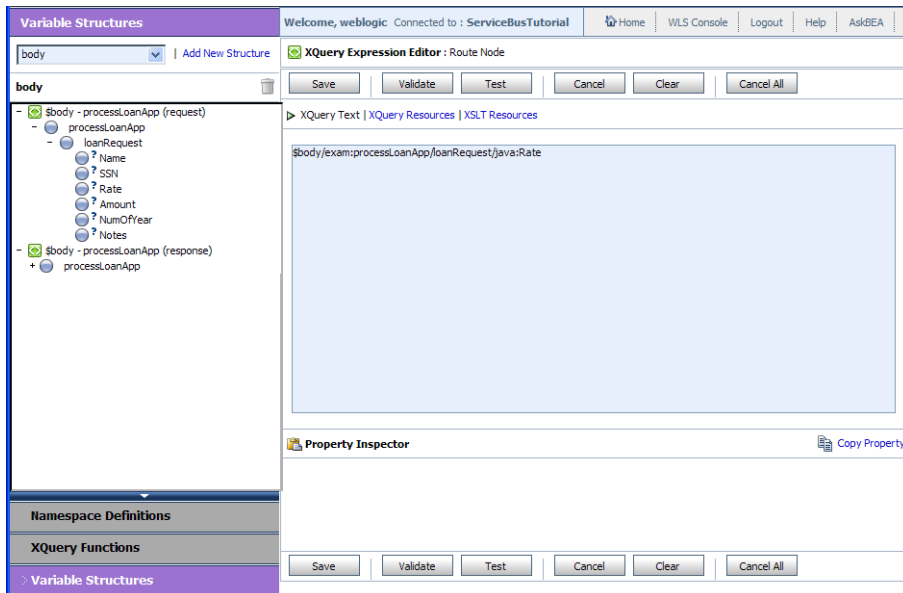
At run time, the proxy service makes its routing decision based on the value in the `rate` element of the message.

- c. Select the circle icon  associated with the `Rate` node element, drag and drop it in the XQuery Expression text box.

The `$body/exam:processLoanApp/loanRequest/java:Rate` 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 `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 3-21 XQuery Expression Editor



10. 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.

11. Click **Save**.

You are returned to the routing table on the Edit Stage Configuration page. `<Expression>` is replaced by the expression that returns the value of the `rate` element in the message.

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

13. 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 then route according to the routing table configuration.

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


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

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

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

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

To Add a Default (Else) Condition to the Routing Table

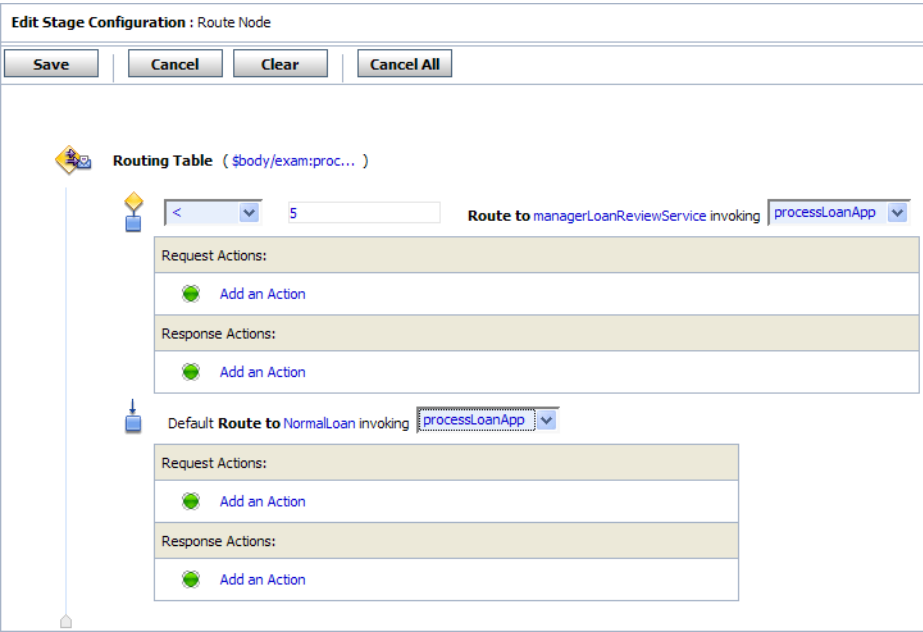
1. Click the **Case** icon , then select **Insert Default Case**. The else condition is added to the routing table.
2. In the else condition, click the **Service** link. The Select Service browser window is displayed.
 - a. Select the `NormalLoan` business service.
 - b. Click **Submit**.
3. Select the `processLoanApp` operation for the service.

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

Tutorial 1. Routing a Loan Application

After completing the above steps, the Edit Stage Configuration page should look like the following figure.

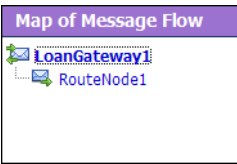
Figure 3-22 Routing Behavior of the Message Flow



- 4. Click **Save** on the Edit Stage Configuration page to save the configuration. You are returned to the Edit Message Flow page.
- 5. Click **Save**. You are returned to the MortgageBroker/ProxyServices page.
- 6. In the Change Center, click **Activate**, then **Submit** to save the session and to instantiate all the configuration changes made during the session.

You can view the map of the message flow of the proxy service as shown in the following figure.

Figure 3-23 Map of Message Flow of the LoanGateway1 Proxy Service



Summary

By completing “[Step 6: Configure the LoanGateway1 Proxy Service](#)” on page 3-22, you have completed the configuration of the logic in the Routing Table that determines the run-time routing behavior for messages. A message is routed to the `managerLoanReviewService` business service if the rate specified in the loan application is less than five, otherwise the message is routed to the `normalLoan` business service.

Step 7: Test Your Loan Application Routing Configuration

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

- A value of 4.9 or less causes the message to be routed to the `managerLoanReviewService` business service.
- A value of 5.0 or greater causes the message to be routed to the `normalLoan` service.

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

This section includes the following test scenarios:

- [To test the managerLoanReviewService Business Service](#)
- [To Test the normalLoan Business Service](#)

To run this test correctly, ensure that WebLogic Server is started in the BEA AquaLogic Service Bus 2.1 (ServiceBusTutorial) domain.

To test the managerLoanReviewService 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\weblogic91\samples\servicebus\examples\src\
```

Tutorial 1. Routing a Loan Application

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 line prompt.

3. Change to the following directory:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

4. To test the routing behavior of the proxy service, you must edit the `build.xml` file *run* target:

a. Open `build.xml`.

b. Scroll to the end of the `build.xml` file looking for the string:

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

c. Replace the string found at this location with 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 for the interest rate.

The remainder of the input parameters specified on the `arg line` are described in the following table.

Table 3-5 Input Parameters

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

5. Ensure that the business services are deployed:

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

```
cd
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

- b. Enter `ant`, then press **Enter**. The `BUILD SUCCESSFUL` message should appear as output.
6. To run the test, on the command line, enter:

```
ant run
```

This tests the routing logic of the proxy service. A response message from the Web service is output to the command line.

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

```
[java] Rate: 4.9
```

```
BUILD SUCCESSFUL
```

The response message indicates that the *Manager Loan Application Processing Service* processed the request because the interest rate requested was 4.9 (less than 5.0).

To Test the normalLoan Business Service

1. Follow the procedure [To test the managerLoanReviewService Business Service](#) to edit the `build.xml` file in the following directory:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

Note: In this case (to test the `normalLoan` business service), enter a value of 5.1 for the interest rate in the `build.xml` file (see [Table 3-5](#)), then save the changes.

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

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

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

- b. Enter `ant`, then press **Enter**. The `BUILD SUCCESSFUL` message should appear as output.
3. To test the routing logic of the proxy service, on the command line, enter:

```
ant run
```

A response message from the Web service is output to the command line.

Tutorial 1. Routing a Loan Application

```
[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 Application Processing Service* processed the request because the interest rate requested was 5.1.

Tutorial 2. Transforming a Loan Application

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

This section includes the following topics:

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

Prerequisites

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

Tutorial Goals

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

Using the AquaLogic Service Bus Console you will build on what you learned in [Chapter 3](#), [“Tutorial 1. Routing a Loan Application,”](#) to:

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

Definition of the Scenario

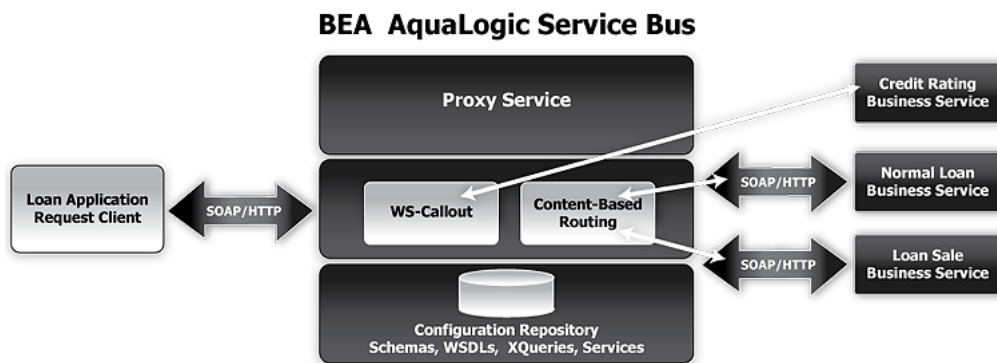
A primary mortgage company uses BEA AquaLogic Service Bus to identify and re-route loan applications that can be sold to secondary loan companies. Loan applications with a principal request of greater than 25,000,000.00 are candidates for sale to a secondary loan company. When AquaLogic Service Bus receives a loan application meeting these criteria, the applicant's credit rating information is retrieved (by making a callout to a Web service) and added to the loan application, then the application is forwarded to the secondary mortgage company's Web service to be processed.

Loan applications with a principal request equal to or less than 25,000,000.00 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 BEA 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 25000000, the application is routed to the `LoanSaleProcessor` business service. If the amount is less than or equal to 25000000, the application is routed to the `NormalLoan` business service. When the loan amount is more than 25000000, the request pipeline makes a service callout to the `CreditRatingService` and

receives the credit rating of the applicant using the `$creditRating` variable. To satisfy the interface requirements of the secondary loan company service, the message body is transformed by adding the credit rating details. The transformed message (`$body`) is routed to a business service that handles applications for large loan amounts. The service returns a response of:

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

Required Resources

You will use the Project folder, `MortgageBroker`, and the directory structure you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in the following table.

Table 4-1 Routing a Loan Application Tutorial Resources

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

Steps in This Tutorial

In this tutorial, you will build on what you learned in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#)—you will import additional WSDL resources, register new business and proxy services, and configure the routing behavior for the proxy service. Configuring the routing behavior includes:

- Adding a Service Callout to get a credit rating
- Changing the namespace and inserting a new `CreditRating` element in the outbound (request) message
- Reverting the namespace change and removing the `CreditRating` element on the response message.

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

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

Step 1: Prepare Your Environment

Ensure that AquaLogic Service Bus is running in the domain you created for the tutorial and that you have completed the steps described in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#).

Step 2: Create a Session and Set up a Project

For this tutorial, you use the project folder, `MortgageBroker`, and the directory structure you created in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#) to hold the project artifacts.

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation panel, select **Project Explorer**.
The Project Explorer pane is opened in the navigation panel and a project page is displayed in the console.
3. In Project Explorer, expand the `MortgageBroker` project tree to expose the sub folders containing the project artifacts.:
 - ProxyService
 - BusinessServices
 - WSDL

Step 3: Create the Resources

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

- [Create the WSDL Resources](#)
- [Create a Proxy Service](#)
- [Create a Business Service](#)

Create the WSDL Resources

Because the WSDLs are the basis on which you create the business services and the proxy service, you must create the WSDL resources (**LoanSale** and **CreditRatingService**) before creating the other resources required for this scenario.





To import the appropriate WSDLs and create the WSDL resources, follow the steps described in [“To Import a WSDL” on page 3-7](#), but for this instance, name your resources and base them on the WSDLs as described in the following table.

Table 4-2 WSDL Configuration Settings

Resource Name	Location of the WSDL
LoanSale	<i>BEA_HOME</i> \weblogic91\servicebus\examples\src\examples\webse rvices\jws_basic\large\LargeLoanPurchasingService.wsdl
CreditRatingService	<i>BEA_HOME</i> \weblogic91\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

Name ▾	Resource Type ▴
 CreditRatingService	WSDL
 LoanSale	WSDL
 managerApproval	WSDL
 normalLoan	WSDL

Create a Proxy Service

In this step, you 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 get the credit rating of the requestor if the loan amount requested is greater than 25 000 000.00. (Note that no units are assigned to the loan amount; it can be dollars or any other unit of currency.)

To create a new proxy service, follow the steps described in [“To Create the Proxy Service” on page 3-10](#), but for this instance, use the proxy service name and the Endpoint URI described in the following table.

Table 4-3 Proxy Service Configuration Settings

Proxy Service Name	LoanGateway2
Service Type	Select the WSDL port option. Select the normalLoan WSDL in the Select a WSDL browser window. Select the helloPort port in the Select a WSDL Definition page.
Endpoint URI	/loan/gateway2

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

Figure 4-3 LoadGateway2 Proxy Service - Configuration Settings Summary

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

When you complete this step, the MortgageBroker/ProxyService folder contains the proxy services you created in this tutorial and in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#) as shown in the following figure.

Figure 4-4 Proxy Service Resources

Name ▾	Resource Type ▴
 LoanGateway1	Proxy Service
 LoanGateway2	Proxy Service

Create a Business Service

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

- **CreditRatingService**—returns the customer's credit rating when a loan application meeting specified criteria is received. This is implemented using a Web service callout (Service Callout).

- **NormalLoan**—is the secondary mortgage company's business service. This service is invoked when a loan request is made for amounts of 25 000 000 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 25 000 000.00.

You created the Normal Loan service in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#). To create the `LoanSaleProcessor` and `CreditRatingService` business services for this scenario, follow the steps described in [“To Create the Normal Loan Business Service” on page 3-16](#). However, in this case, configure your services using the names, service types, and endpoint URIs described in the following tables.

Table 4-4 Configuration Settings for the LoanSaleProcessor Business Service

Business Service Name	LoanSaleProcessor
Service Type	Select the WSDL port option. Select the LoanSale WSDL in the Select a WSDL browser window. Select the helloPort port in the Select a WSDL Definition page.
Endpoint URI	Ensure that the pre-populated value for the Endpoint URI in the Existing URIs field is <code>http://host:port/njws_basic_ejb/LargeSimpleBean</code> where <code>host:port</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\weblogic91\samples\servicebus\examples\src\examples.properties</code> For more information, see “Edit the examples.properties File” on page 2-4 .





Table 4-5 Configuration Settings for the CreditRatingService Business Service

Business Service Name	CreditRatingService
-----------------------	---------------------

Service Type	<p>Select the <code>WSDL</code> port option.</p> <p>Select the <code>CreditRatingService</code> WSDL in the Select a WSDL browser window.</p> <p>Select the <code>helloPort</code> port in the Select a WSDL Definition page.</p>
Endpoint URI	<p>Ensure that the pre-populated value for the Endpoint URI in the Existing URIs field is <code>http://host:port/njws_basic_ejb/CreditSimpleBean</code> where <code>host:port</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\weblogic91\samples\servicebus\examples\src\examples.properties</code></p> <p>For more information, see “Edit the examples.properties File” on page 2-4.</p>

When you complete this step, the `MortgageBroker/BusinessService` folder contains the business services 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

Name ▾	Resource Type ▲
 CreditRatingService	Business Service
 LoanSaleProcessor	Business Service
 managerLoanReviewService	Business Service
 NormalLoan	Business Service

Summary

By completing [Step 1: Prepare Your Environment](#) to [Step 3: Create the Resources](#), you created the resources required for this scenario. You configured the proxy service with a base configuration. Subsequent steps ([Step 4: Configure the Routing for the LoanGateway2 Proxy Service](#) to [Step 6: Configure the Routing Table Response Actions for the LoanGateway2 Proxy Service](#)) describes how to complete the configuration of the proxy service to add the routing, transformation, and Service Callout behavior for the loan application messages.

Step 4: Configure the Routing for the LoanGateway2 Proxy Service

The proxy service is implemented in 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 panel, select **Resource Browser**.

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

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 `LoanGateway2` proxy service, click the **Edit Message Flow** icon .

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

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

In the configuration dialog, `RouteNode1` is added.

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

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

5. Click **Add an Action**, then select **Routing Table**.

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

You will configure the routing table to route messages to business services based on evaluating the `amount` element of the incoming message. You must create an XQuery expression to do this. You can do so using the XQuery Expression Editor.

Configure the Routing Expression

1. In the Routing table, click the `<Expression>` link. The XQuery Expression Editor page is displayed.
2. Accept the default selection of **XQuery Text** in the page.
3. Click **Variable Structures**. The Variable Structures pane is displayed.
4. From the **Select Structure** drop-down list in the Variable Structures pane, select `body`.

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


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

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

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

- c. Select the circle icon  associated with the `Amount` node element, drag and drop it in the XQuery Expression text box.

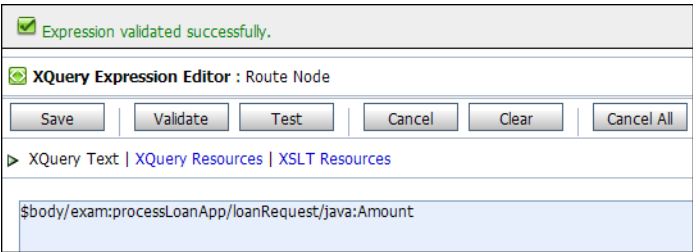
The `$body/exam:processLoanApp/loanRequest/java:Amount` XQuery expression is written in the text box.

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

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

It is good practise 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 4-6 Validate Routing Expression



6. Click **Save**.

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

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

The routing table now contains an expression that determines the routing behavior. If the value in the `Amount` field is greater than 25000000, then messages are routed according to the routing table configuration.

8. Click the `Service` link to define the service to which you want to route when the amount is greater than 25000000. The Select Service browser window is displayed.

a. Select the **LoanSaleProcessor** business service

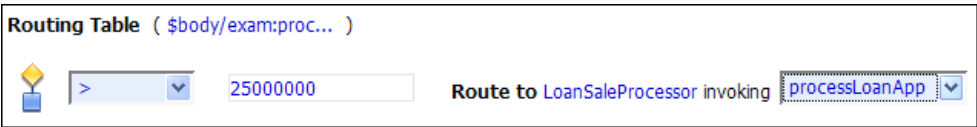
b. Click **Submit**.

9. In the Routing Table Operation drop-down list, select the `processLoanApp` operation.

This is the operation on the **LoanSaleProcessor** service business service that is invoked at run time if the amount of the loan requested in the loan application is greater than 25000000.

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

Figure 4-7 Routing the Loan Application to LoanSaleProcessor



When a loan application with the amount of the loan greater than 25000000 is identified, a Web service callout (Service Callout) is performed to retrieve the customer's credit rating. The credit rating information is added to the loan application and the application is then forwarded to the secondary mortgage company's business service to be processed. The next section ([Step 5: Configure the Routing Table Request Actions for the LoanGateway2 Proxy Service](#)) describes how to configure the proxy service to do the Service Callout and transform the message appropriately for the target service.

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

To configure the Routing Table Request actions, execute the following steps:

- [Configure the Service Callout Input Parameter](#)
- [Add a Service Callout Action](#)
- [Configure the Message Transformation](#)

Configure the Service Callout Input Parameter

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

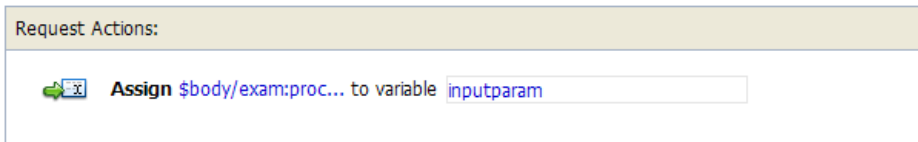
1. Ensure that the Edit Stage Configuration page for the routing table you created in “[Step 4: Configure the Routing for the LoanGateway2 Proxy Service](#)” on page 4-10 is open in the AquaLogic Service Bus Console.
2. In the Request Actions panel of the Routing Table, click **Add an Action**, then select **Assign**.
The Assign action is added to the Request Action pipeline in the Routing Table.
3. Click the `<Expression>` link. The XQuery Expression Editor page is displayed.
 - a. Accept the default selection of **XQuery Text** in the page.
 - a. Select `body` from the Variables Structures drop-down list in the Variables Structures pane, then expand the element list and drag the `loanRequest` element and drop it into the XQuery expression text box. The XQuery expression is written in the text box:


```
$body/exam:processLoanApp/loanRequest
```
 - b. Click **Validate** to validate the XQuery, then **Save**.

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

4. Enter `inputparam` in the variable text box.

Figure 4-8 Configure Service Callout Input Parameter



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

Add a Service Callout Action

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

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

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

2. Click the Service Callout **Service** link. The Select Service browser window is displayed.
 - a. In the Select Service window, select `CreditRatingService`.
 - b. Click **Submit**.
3. From the Operation drop-down list, select `processLoanApp`. The fields that allow you to configure the request and response parameters for the Service Callout are displayed:
 - a. In the **Service Callout Request Parameters** section, set the Loan request parameter to `inputparam` (the message context variable you defined in the preceding section).
 - b. In the **Service Callout Response Parameters** section, set the return parameter name to `creditRating`.

Figure 4-9 Configure Service Callout Action

The screenshot shows a configuration window titled "Service Callout to CreditRatingService" with a dropdown menu set to "processLoanApp". The window is divided into several sections:

- Request Parameters:** A table with one row: "loanRequest" mapped to "inputparam".
- Response Parameters:** A table with one row: "return" mapped to "creditRating".
- SOAP Request Header:** An empty text input field.
- SOAP Response Header:** An empty text input field.


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

Configure the Message Transformation

In this section, you will configure the transformation of the message to match the public contract (interface requirements) of the `loanSaleProcessor` Service. Specifically, this section describes how to:

- Configure the proxy service to rename the namespace for the message to the namespace required by the target service. You do this using XPath expressions. This configuration is described in [“To Rename the Namespace” on page 4-15](#).
- Configure the proxy service to insert a new element into the message. To do this, you must add an element action in the configuration. This configuration is described in [“To Insert a New Element into the Message” on page 4-16](#).

To Rename the Namespace

1. Click the **Service Callout** icon , then select **Add an Action→Rename**.
The Rename action configuration fields are added to the Request Actions panel.
2. In the **Rename** statement, click the `<XPath>` link. The XPath Expression Editor is displayed.
 - a. In the XPath Expression text box, enter the following XPath expression:

```
./java:*
```

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

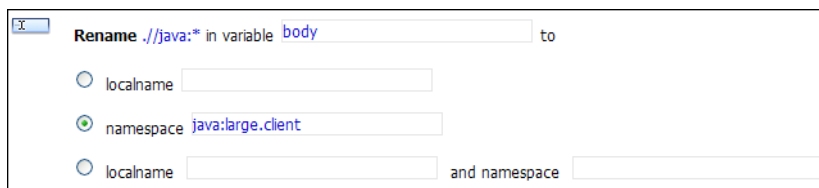
- b. Click **Save**. You are returned to the Edit Stage Configuration page.
3. In the variable text box, enter `body`.

The `body` context variable holds the body of the message. By completing these steps, you created a condition in which the XPath expression finds all the namespaces with the `java` prefix in the `body` context variable. The next step specifies the namespace with which to replace the namespaces identified by the XPath expression.
4. Select the namespace option and enter the fully qualified namespace to be substituted in the message. The new namespace is `java:large.client`.

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


The Rename action should now look like the following figure.

Figure 4-10 Rename Namespace



The screenshot shows a configuration window for the 'Rename' action. At the top, it says 'Rename `./java:*` in variable `body` to'. Below this, there are three radio button options: 'localname', 'namespace', and 'localname and namespace'. The 'namespace' option is selected, and its corresponding text box contains the value `java:large.client`. The other text boxes are empty.

To Insert a New Element into the Message

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

The Insert action configuration fields are added to the Request Actions panel.
2. In the Insert action, click `<Expression>`.

The XQuery Expression Editor page is displayed. You will add a new namespace on this page.
3. In the **Namespace Definitions** palette, locate to the **User Defined Namespaces** area.

Figure 4-11 User Defined Namespace

- 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.

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>
```

- The element you add to the message must be associated with the `java:large.client` namespace. Therefore you specify the namespace for the element with `lg:.`
- `{ }` tells the XQuery engine that the content between the `{ }` is not XML and must be interpreted.

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

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

Note: It is good practice is to validate your expressions before submitting them.

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

7. Click the <XPath> link. The XPath Expression Editor page is displayed:
 - a. Click **Variable Structures**. The Variable Structures pane is displayed.
 - b. In the **Select Structure** drop-down list in the Variables Structures pane, select `body`.
 - c. Expand the element list and drag the `Notes` element and drop it into the XPath Expression text box.

The following XPath expression is written to the text box

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

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

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

```
./exam:processLoanApp/loanRequest/lg:Notes
```
9. Click **Validate**, then **Save** to return to the Edit Stage Configuration page.
10. Enter `body` in the **Variable** text field (the last field in the expression). This is the context variable into which the new `<CreditRating>` element is inserted at run time.

The Insert action should now look like the following figure.

Figure 4-12 Insert New Element into Message



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

After completing the above steps, the configuration of the outbound message should look like the following figure.

Figure 4-13 Configuration of Outbound Message

The screenshot shows the 'Routing Table' configuration for a message with body path `$body/exam:proc...`. The configuration includes the following sections:

- Request Actions:**
 - Assign:** Assign `$body/exam:proc...` to variable `inputparam`.
 - Service Callout:** Service Callout to `CreditRatingService` invoking `processLoanApp`.

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

SOAP Request Header:
 SOAP Response Header:

Transport Headers		
Name	Action	Options
Add Header		
- Rename:** Rename `./java:*` in variable `body` to:
 - ☐ localname
 - ☒ namespace `java:large.client`
 - ☐ localname and namespace
- Insert:** Insert `<lg:CreditRatin...` `after` `./exam:process...` in variable `body`.

The next step describes how to configure the response actions for the `LoanGateway2` proxy service.

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

This step describes how to 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). Specifically, this section describes how to:

- Configure the proxy service to remove the `CreditRating` element from the response message. This configuration is described in [“To Delete an Element”](#) on page 4-20.

- Configure the proxy service to revert the namespace to the namespace in the original message: `java:normal.client`. This configuration is described in [“To Add a Rename Action” on page 4-21](#).

To Delete an Element

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

The **Delete** action configuration fields are added to the Response Actions in the Routing Table.

3. Select the **<XPath> in variable** option, then click **<XPath>**. The XPath Expression Editor page is displayed.

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

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

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

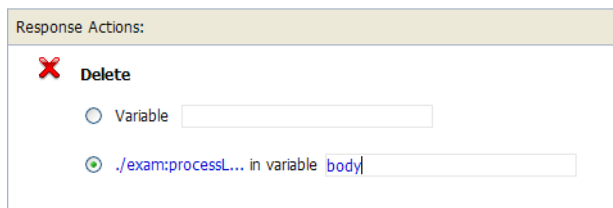
- b. Click **Validate**, then **Save**. You are returned to the Edit Stage Configuration page.

4. Enter `body` in the variable text field in the Delete action.

The configuration for the Delete action is therefore:

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

Figure 4-14 Delete Element



Response Actions:

Delete


☐ Variable

☒ XPath

By completing these steps, you have specified that AquaLogic Service Bus removes the credit rating element from the response message as the message is processed in the response pipeline.

To Add a Rename Action

This section describes how to rename the namespace to the namespace required by the client, that is `java:normal.client`. (Recall that you configured the request message to the `loanSaleProcessor` business service to change the namespace to that required by that service. (See [“To Rename the Namespace” on page 4-15.](#))

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

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

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**. You are returned to the Edit Stage Configuration page.
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`.

By completing these steps, 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 4-15 Add Rename Action



The screenshot shows a configuration window for a 'Rename' action. At the top, it says 'Rename ./lg:* in variable `body` to'. Below this, there are three radio button options: 'localname', 'namespace' (which is selected), and 'localname and namespace'. The 'namespace' option has a text field next to it containing the value `java:normal.client`.

After completing the above steps, the configuration of the response action of the proxy service should look like the following figure.

Figure 4-16 Configuration of Response Actions of Proxy Service

Response Actions:

Delete

☐ Variable

☒ ./exam:processL... in variable

Rename in variable to

☐ localname

☒ namespace

☐ localname and namespace

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


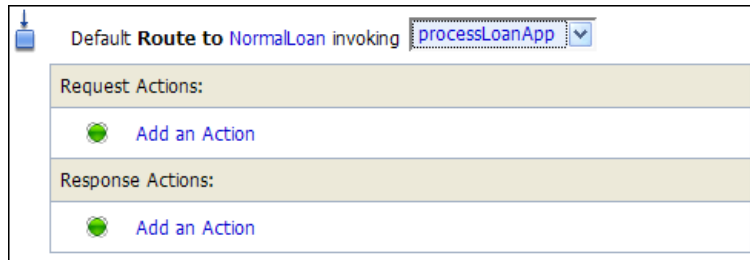
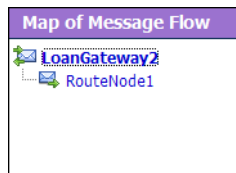
1. Click the routing table **Case** icon , then select **Insert Default Case**.
The default case is added after the routing table on the Edit Stage Configuration page.
2. Click **<Service>** to display the Select Services browser window.
 - a. Select the `NormalLoan` business service.
 - b. Click **Submit**.
3. In the Default case configuration fields, select the operation `processLoanApp` as shown in the following figure.

Figure 4-17 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**, then **Submit** to save the session configuration information.

You can view the map of the message flow of the proxy service as shown in the following figure.

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

Step 8: 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 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 25000000 causes the message to be routed to the `normalLoan` business service.
- A value greater than 25000000 causes the message to be routed to the `LoanSaleProcessor` business service (the large loan processing service).

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

This section includes the following test scenarios:

- [To Test the LoanSaleProcessor \(the Large Loan Purchasing Service\) Business Service](#)
- [To Test the NormalLoan Business Service](#)

To Test the LoanSaleProcessor (the Large Loan Purchasing Service) 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\weblogic91\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 line prompt.

3. Change to the following directory:

```
BEA_HOME\weblogic91\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:

- a. Open `build.xml`.

- b. Scroll to the end of the `build.xml` file looking for the string:

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

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

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

This changes the URI to that of the proxy service to that of the deployed Web service (gateway2) running in the tutorial domain, and specifies a value of 25000001 for the loan amount. (For more information about the input parameters for the test, see [Table 3-5](#).)

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

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

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

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

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

```
ant run
```

This tests the routing logic of the proxy service. The message from the Web service is output to the command line.

```
[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 25000001 (greater than 25000000).

To Test the NormalLoan Business Service

1. Follow the procedure “[To Test the LoanSaleProcessor \(the Large Loan Purchasing Service\) Business Service](#)” on page 4-24 to edit the build.xml file in the following directory:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

In this case (to test the normalLoan business service), in the build.xml file, enter a value of 25000000 for the loan amount (see [Table 3-5](#)), then save the changes.

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

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

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

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

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

```
ant run
```

A response message from the Web service is output to the command line.

Tutorial 2. Transforming a Loan Application

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

```
[java] Rate: 5.1
```

```
BUILD SUCCESSFUL
```

The response message indicates that the *Normal Loan Application Processing Service* processed the request because the amount of the loan requested was 25000000.

Best Practices

In this tutorial, you configure the transformation and replace 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 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 better practice to add a single transformation to a request pipeline rather than configure five identical sets of transformations for each routing configuration in each route node. The run-time execution cost is the same, but the cost of maintaining, configuring, and understanding the latter implementation is higher and more work intensive.

Tutorial 3. Validating a Loan Application

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

Validation can happen in a number of ways:

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

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

This section includes the following topics:

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

Prerequisites

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

Tutorial Goals

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

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

Features Highlighted

Developing this tutorial exposes you to the following features of AquaLogic Service Bus

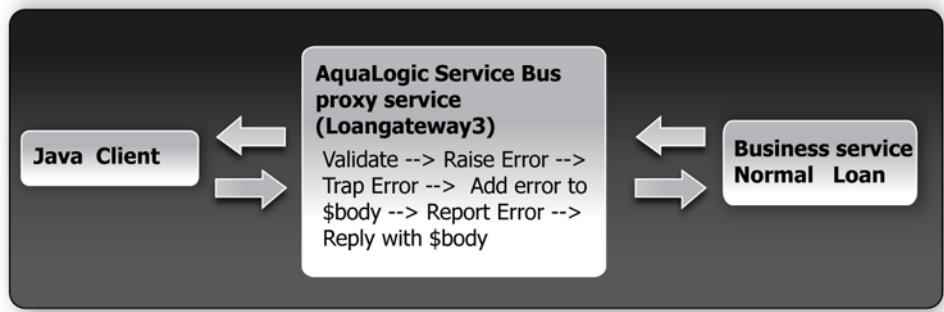
- Configuring a request response pipeline
- Using the validate action to raise a validation exception
- Showing how a stage error handler can trap an error and return a customized error message

Definition of the Scenario

A primary mortgage company routes every loan application through a BEA AquaLogic Service Bus proxy service which validates the message. If the application is incomplete, it is written to an error directory and an error message is returned to the client. If complete, it is routed to a business service for review. If 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 where AquaLogic Service Bus fits in your enterprise 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 Process in AquaLogic Service Bus

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

In this scenario, one of the following actions takes place:

- The message is returned by the response pipeline to the client along with the validation error message.
- The message is forwarded to a business service, `NormalLoanApprovalService`, which returns a reply. The loan application was accepted.

Required Resources

You will use the Project folder, `MortgageBroker`, and the directory structure you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in the following table.

Table 5-1 Routing a Loan Application Tutorial Resources

Resource Name	Description
normalLoan	This is the WSDL resource
NormalLoan Web service	This is an external business service used by AquaLogic Service Bus
LoanGateway3	This is the AquaLogic Service Bus proxy service

Steps in This Tutorial

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

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

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

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

- [Step 1: Prepare Your Environment](#)
- [Step 2: Create a Session and Set up a Project](#)
- [Step 3: Create the Resources](#)
- [Step 4: Configure the LoanGateway3 Proxy Service](#)
- [Step 5: Add a Default Routing Case](#)
- [Step 6: Test Your Loan Application Routing Configuration](#)
- [Step 7: Using Reporting and Monitoring](#)

Step 1: Prepare Your Environment

Ensure that AquaLogic Service Bus is running in the domain you created for the tutorial and that you have completed the steps described in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#).

Step 2: Create a Session and Set up a Project

For this tutorial, you use the project folder, `MortgageBroker`, and the directory structure you created in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#) to hold the project artifacts.

1. Click **Create** in the Change Center to create a new session.
2. In the AquaLogic Service Bus Console navigation panel, select **Project Explorer**.
The Project Explorer pane is opened in the navigation panel and a project page is displayed in the console.
3. In **Project Explorer** expand the `MortgageBroker` project tree to expose the sub folders containing the project artifacts.:
 - ProxyService
 - BusinessServices
 - WSDL

Step 3: Create the Resources

In this step, you will create the proxy service, **LoanGateway3**. You will use the existing `normalLoan` service WSDL resource as the basis on which to validate the content of the message. This WSDL resource was created in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#) in the section [Create the normalLoan Service WSDL Resource](#). If the loan application is valid, that is, it meets the expected criteria, it is routed to the existing `normalLoan` business service. This business service was created in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#) in the section [To Create the Normal Loan Business Service](#).

Create a Proxy Service

In this step, you create a proxy service. The proxy service is used to route the loan application to the appropriate business service. Base this proxy service on the proxy service you previously created in Tutorial 1.

To create a new proxy service, follow the steps described in [“To Create the Proxy Service” on page 3-10](#), but for this instance use the proxy service name and the Endpoint URI described in the following table.

Table 5-2 Proxy Service Configuration Settings

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

Review the configuration settings prior to registering the proxy service.

When you complete this step, the `MortgageBroker/ProxyService` folder contains the proxy services you created in this tutorial and in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#):

- `LoanGateway1`
- `LoanGateway2`
- `LoanGateway3`

Step 4: Configure the LoanGateway3 Proxy Service


The proxy service is implemented in AquaLogic Service Bus as a message flow, which includes request and response pipelines. AquaLogic Service Bus Message Flows define the implementation of proxy services. 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 behavior of the `LoanGateway3` proxy service, you will complete the following for the message flow:

- [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 panel, select **Project Explorer**.
The Project Explorer pane is opened in the navigation panel.

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

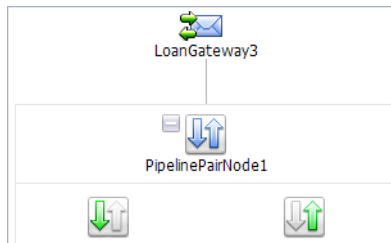
2. In the **Actions** column associated with the `LoanGateway3` proxy service, click the **Edit Message Flow** icon .

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

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


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

Figure 5-2 Add Pipeline Pair



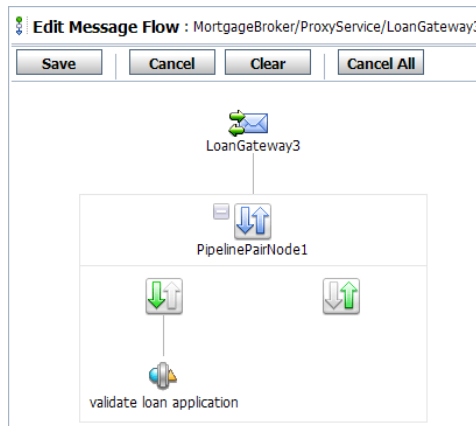
Configure the Request Actions for the LoanGateway3 Proxy Service

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

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


The screen should appear as shown in the following figure.

Figure 5-3 Add Stage to Request Pipeline



Configure an Action to do Message Validation

1. To edit the stage, click **validate loan application**, then select **Edit**→**Stage**. The Edit Stage Configuration page is displayed.
2. Click the **Add an Action** link and select **Validate** from the drop-down list. The Validate action is added to the stage.
3. In the Validate statement, click the **XPath** link to edit the XPath expression. The XPath Expression Editor page is displayed.
4. Click **Variable Structures**. The Variable Structures pane is displayed.
5. In the Variables Structure pane, from the **Select Structure** drop-down list, select `body`.
A structural representation of the `body` element is displayed in the Variables Structure pane.
 - a. Click + beside the **processLoanApp** element to expand the element.
The **processLoanApp** element is within the **\$body - processLoanApp (request)** element.
 - b. Click + to expand the **loanRequest** element.
A graphical representation of the structure of the loan request document is displayed.

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

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

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

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

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

7. Click **Save**.

You are returned to the Edit Stage Configuration page. `<XPath>` is replaced by the XQuery expression.

8. In the **Variable** text field, enter `body`.
9. Select `resource`, then select **WSDL** from the list. The Select a WSDL browser window is displayed.

The proxy service is based on the WSDL resource you originally created, hence you must validate the message against this resource.

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

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

- a. In the Select WSDL Definitions panel, from the Types category, select the WSDL type for this WSDL as `loanStruct`.
- b. Click **Submit** to complete your selection.

11. Select the **Raise Error on validation failure** option.

The validate action should appear as shown in the following figure.

Figure 5-4 Configure Action for Message Validation

12. Click **Save**.

Note: Validation of a message should happen as early as possible in a message flow. This prevents unnecessary processing of the message in a case where the content of the message may be invalid. If an exception is raised, further processing is stopped and a response is immediately returned to the client.

Messages can be validated at the route node or using a separate Validate action (as in this case). Creating a Validation stage make the processing of the message more modular and logically breaks out the actions in the message flow. In a case where multiple validations may be performed on a message, it allows for logical expansion of the validate action.

The configuration is saved.

Add a Stage Error Handler using a Replace Action

Add a Replace action to the stage and configure this action to replace the `Notes` element value with the `fault` context variable `reason` text. This results in the `fault` description being returned to the client.

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

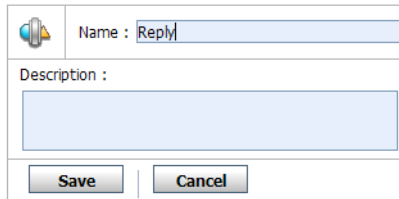

Figure 5-5 Edit Name and Description of Stage


Figure 5-5 shows a dialog box for editing a stage. It has a 'Name' field containing the text 'Reply' and a 'Description' text area below it. At the bottom, there are 'Save' and 'Cancel' buttons.

5. To edit the stage, click **reply**, then select **Edit**→**Stage**. The Edit Stage Configuration page is displayed as shown in the following figure.

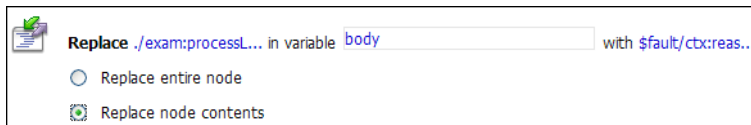
Figure 5-6 Edit ErrorHandler page

6. Click the **Add an Action** link, then select **Replace**.
7. In the **Replace** statement, click the XPath link to edit the XPath expression. The XPath Expression Editor page is displayed.
8. From the **Select Structure** drop-down list in the Variable Structures pane, select **body**.
A structural representation of the **body** element is displayed in the Variable Structures pane.
 - a. Click + beside the **processLoanApp** element to expand the element.
The **processLoanApp** element is within the **\$body - processLoanApp (request)** element.
 - b. Click + to expand the **loanRequest** element.
 - c. Select the circle icon  associated with the **Notes** element of **\$body**, drag and drop it on the XPath Expression text box. The following XPath expression is written in the text box.
`./exam:processLoanApp/loanRequest/java:Notes`

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


9. Click **Validate** to validate the expression, then click **Save**.
10. In the Variable text field, enter `body`.
11. Click the `<Expression>` link to display the XQuery Expression Editor page and accept the default selection of **XQuery Text** in the page.
12. In the XQuery expression text box, enter:
`$fault/ctx:reason/text()`
13. Click **Validate**, then **Save**.
14. Select the **Replace node contents** option. You do not want to replace the entire node. The replace action should appear as shown in the following figure.


Figure 5-7 Add Error Handler Using Replace Action



Add a Report Action to Report the Error

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

1. Click the **Replace** action  and select **Add an Action→Report**. The Report action is added to the page.
2. In the Report statement, click the `<Expression>` link. The XQuery Expression Editor page is displayed.
3. Accept the default selection of **XQuery Text** in the page.


4. In the **Variable Structures** pane, select `body - processLoanApp` from the **Select Structure** drop-down list.
 - a. Select the icon  associated with the `$body` element and drag and drop it on the XQuery Expression text box.

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

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

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

You return to the Edit Stage Configuration page.

5. In the key-value pair table, select the link `Add a key`.
 - a. Enter a name for the key. In this case, enter `errorCode`.
 - b. In the key value field, click XPath.
6. In the **Variable Structures** pane, select `fault` from the **Select Structure** drop-down list.
 - a. Select the icon  associated with the `errorCode` and drag and drop it on the XPath Expression text box.

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

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


This step allows the reporting data to be indexed by `errorCode`. The keys are a convenient way to identify a message. You can use the `errorCode` to search for reporting entries in the reporting module.



7. Click **Validate** to validate the XQuery, then click **Save**.
8. In the **Variable** text field, enter `fault`.

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

The report action should appear as shown in the following figure.

Figure 5-8 Add Report Action


 **Report** \$body with search keys:

Key Name	Key Value	Options
 <code>errorCode</code>	<code>./ctx:errorCode</code> in variable <code>fault</code>	

Add a Reply Action to Reply with a Failure Code.

1. Click the **Report** action and select **Add an Action->Reply**. The Reply action is added to the page.
2. Select the **With Failure** option as shown in the following figure.

Figure 5-9 Reply Action to Reply with Failure Code

 **Reply**

☐ With Success

☒ With Failure


3. Click **Save** the stage configuration.
4. Save the error handler configuration.

Now, the Edit Stage Configuration page should appear as shown in the following figure.

Figure 5-10 Request Actions for LoanGateway3 Proxy Service


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



Save Cancel Clear Cancel All


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

☐ Replace entire node

☒ Replace node contents

 **Report** `$body` with search keys:

Key Name	Key Value	Options
 <code>errorCode</code>	<code>./ctx:errorCode</code> in variable <code>fault</code>	

 **Reply**

☐ With Success

☒ With Failure

Step 5: Add a Default Routing Case

Configure the Route Node

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

1. Click **PipelinePairNode1**, then select **Add Route Node**.
2. Click **RouteNode1** and select Edit→**Route Node**. The Edit Stage Configuration page is displayed.
3. Click **Add an Action**, then select **Routing**.
The Route action is displayed in the page.
4. In the Route statement, click the `Service` link to define the service to which you want to route the messages. The Select Service page is displayed.
 - a. Select the `NormalLoan` business service.
 - b. Click **Submit**.

You are returned to the Edit Stage Configuration page.

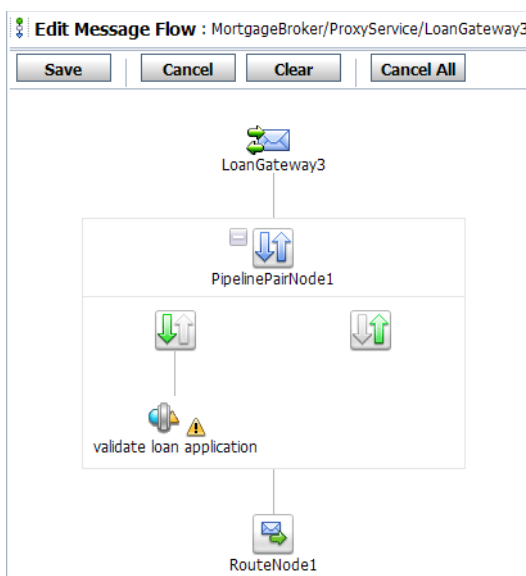
5. In the **Operation** drop-down list, select `processLoanApp`.

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

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

The screen should appear as shown in the following figure.

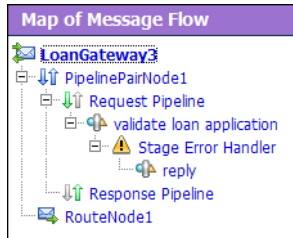
Figure 5-11 Configure Message Flow of LoanGateway3 Proxy Service



6. Click **Save**, then **Save** to save the configuration. You are returned to the MortgageBroker/ProxyService page.
7. Click **Activate**, the **Submit**.

You can view the map of the message flow of the proxy service as shown in the following figure.

Figure 5-12 Map of Message Flow of the LoanGateway3 Proxy Service



Step 6: Test Your Loan Application Routing Configuration

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

- A non-integer value raises an error and returns an error message in the Notes field of the message.
- An integer value causes the message to be routed to the `NormalLoanApprovalService` business service.

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

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

This section includes the following test scenarios:

- [To Test the Validation Exception](#)
- [To Test the NormalLoanApprovalService Business Service](#)

To run this test correctly, ensure that WebLogic Server is started in the BEA AquaLogic Service Bus 2.1 (ServiceBusTutorial) domain.

To Test the Validation Exception

- 1. Open a Command window.
- 2. If you have not run the `setEnv.cmd` to set up your environment, you must do so now.
 - a. Change to the following directory:

```
BEA_HOME\weblogic91\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 line prompt.

- 3. Change to the following directory:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

- 4. To test the routing behavior of the proxy service, in this case, you must edit the `build.xml` file *run* target:
 - a. Open `build.xml`.
 - b. Scroll to the end of the `build.xml` file looking for the string:

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

- c. Replace the string found at this location with:

```
<arg line="http://${wls.hostname}:${wls.port}/loan/gateway3 steve 555 2500 20.5 4.9 notes" />
```

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

The remainder of the input parameters specified on the `arg line` are described in the following table.

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

5. Ensure that the business services are deployed:

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

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

- b. Enter `ant`, then press **Enter**.

6. To run the test, on the command line, enter:

```
ant run
```

This tests the routing logic of the proxy service. A response message from the Web service is output to the command line.

```
[java] Loan Application Response: ALSB Validate action failed validation
[java] Rate: 4.9
```

```
BUILD SUCCESSFUL
```

The response message indicates that a validation exception occurred as the value entered in the loan duration field was a non-integer value.

To Test the NormalLoanApprovalService Business Service

1. Follow the procedure [To Test the Validation Exception](#) to edit the `build.xml` file in the following directory:

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

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

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

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

Tutorial 3. Validating a Loan Application

```
BEA_HOME\weblogic91\samples\servicebus\examples\src\examples\webservices\jws_basic\normal
```

- a. Enter ant, then press **Enter**.
3. To test the routing logic of the proxy service, on the command line, enter:

```
ant run
```

A response message from the Web service is output to the command line.

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

```
[java] Rate: 4.9
```

```
BUILD SUCCESSFUL
```

The response message indicates that the `NormalLoan` business service processed the request because the loan duration requested was 20, an integer value.

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 Your 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. If you do not wish to use the out-of-the-box reporting provider, you can create your own reporting provider using the Reporting Service Provider Interface (SPI).

Reporting automatically happens in the AquaLogic Service Bus Console. Reporting features are available from the Dashboard. Information is presented in a drill-down format. The more you explore the links presented, the more information is made available to you.

For more information, see:

- [Reporting](#) in the *AquaLogic Service Bus User Guide*.
- [Reporting](#) in the *AquaLogic Service Bus Console Online Help*.

Monitoring

BEA AquaLogic Service Bus provides the capability to 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 the *AquaLogic Service Bus User Guide*.
- [Monitoring](#) in the *AquaLogic Service Bus Console Online Help*.

Tutorial 3. Validating a Loan Application