



# BEA AquaLogic® Service Bus

## Tutorials

Version: 3.0  
Revised: February 2008



# Contents

## 1. Introduction to the ALSB Tutorials

ALSB Concepts .....	1-2
Proxy Services and Business Services .....	1-3
Message Flows and Pipelines .....	1-4
Pipeline Pairs .....	1-4
Overview of the Tutorials .....	1-5
How to Use the Tutorials .....	1-6

## 2. Getting Started with the ALSB Tutorials

Scenarios in the Tutorials .....	2-1
Setting Up the Tutorials .....	2-2
Install ALSB .....	2-3
Create an ALSB Domain .....	2-3
Start ALSB .....	2-10
Load the Client Application .....	2-11
Log in to ALSB Console .....	2-17
Where to Go from Here .....	2-18

## 3. Tutorial 1. Routing a Loan Application

Prerequisites .....	3-1
Tutorial Objectives .....	3-1
Definition of the Scenario .....	3-2
Overview of the Run-Time Process in ALSB .....	3-2

Required Resources . . . . .	3-3
Tasks in This Tutorial . . . . .	3-3
Prepare the Environment . . . . .	3-3
Create a Session and Set up a Project . . . . .	3-3
Create the WSDL Resources . . . . .	3-6
Create a Proxy Service . . . . .	3-8
Create the Business Services . . . . .	3-16
Configure the LoanGateway1 Proxy Service. . . . .	3-23
Test Your Loan Application Routing Configuration Using the Test Console . . . . .	3-32
Where to go From Here . . . . .	3-36

## 4. Tutorial 2. Transforming a Loan Application

Prerequisites . . . . .	4-1
Tutorial Objectives . . . . .	4-1
Definition of the Scenario . . . . .	4-2
Overview of the Run-time Process in ALSB . . . . .	4-2
Required Resources . . . . .	4-3
Tasks in This Tutorial . . . . .	4-3
Prepare the Environment . . . . .	4-3
Create a Session . . . . .	4-4
Create the Resources . . . . .	4-4
Configure Routing for LoanGateway2 Proxy Service . . . . .	4-10
Configure Routing Request Actions for LoanGateway2 Proxy Service . . . . .	4-16
Configure Routing Response Actions for LoanGateway2 Proxy Service . . . . .	4-29
Add a Default Routing Case . . . . .	4-31
Test the Loan Application Routing Configuration Using the Test Console. . . . .	4-34
Best Practices . . . . .	4-36

# 5. Tutorial 3. Validating a Loan Application

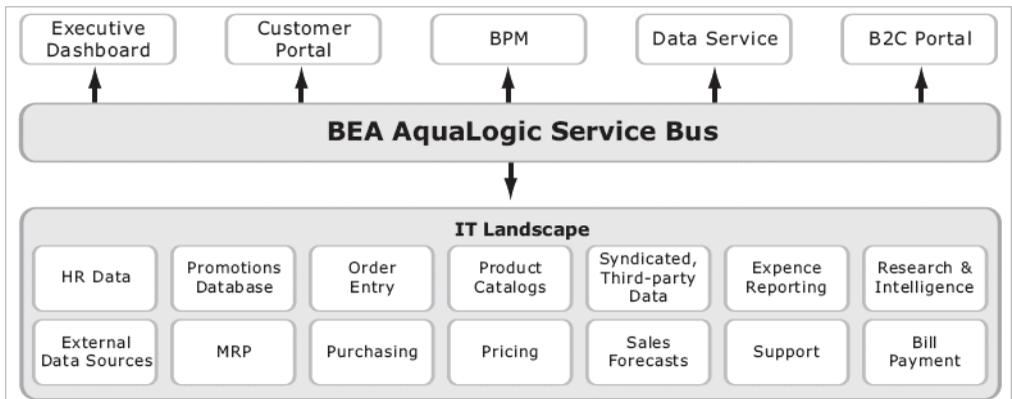
- Prerequisites . . . . . 5-2
- Tutorial Objectives . . . . . 5-2
  - Features Highlighted . . . . . 5-2
- Definition of the Scenario . . . . . 5-2
  - Overview of the Run-Time Process in ALSB . . . . . 5-3
  - Required Resources . . . . . 5-4
- Tasks in This Tutorial . . . . . 5-4
  - Prepare Your Environment . . . . . 5-4
  - Create a Session . . . . . 5-5
  - Create the Resources . . . . . 5-5
  - Configure the LoanGateway3 Proxy Service . . . . . 5-7
  - Add a Default Routing Case . . . . . 5-26
  - To Test the Loan Application Routing Configuration Using Test Console . . . . . 5-29
  - Step 7: Using Reporting and Monitoring . . . . . 5-33



# Introduction to the ALSB Tutorials

AquaLogic Service Bus is a market-leading enterprise service bus built from the ground up for Service Oriented Architecture (SOA) lifecycle management. It provides foundation capabilities for service discovery and intermediation, rapid service provisioning and deployment, and governance. This service-infrastructure software adheres to the SOA principles of building coarse grained, loosely coupled, and standards-based services, creating a neutral container in which business functions may connect service consumers and back-end business services, regardless of underlying infrastructure. [Figure 1-1](#) illustrates the role of ALSB as a service intermediary in an enterprise IT SOA architecture.

**Figure 1-1 ALSB Intermediary**



ALSB Console enables you to:

configure a proxy service to suit your needs and to manage web services by controlling the service and policy configurations and by monitoring system and operations tasks. ALSB relies on WLS run-time features. For more information, see [AquaLogic Service Bus Concepts and Architecture](#).

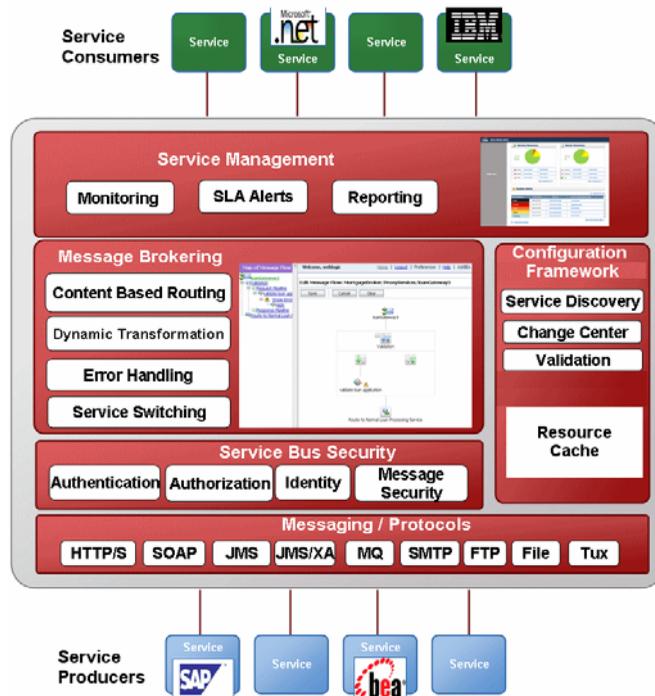
This topic consists of the following sections:

- [ALSB Concepts](#)
- [Overview of the Tutorials](#)
- [How to Use the Tutorials](#)

## ALSB Concepts

ALSB allows you to manage web services and deliver authentic message brokering through the configuration of proxy services in ALSB Console. The underlying concepts of ALSB are briefly described in this section. For more information about ALSB, see [AquaLogic Service Bus Concepts and Architecture](#). ALSB manages the routing and transformation of messages in an enterprise system to promote seamless application integration. [Figure 1-2](#) illustrates the functional components of ALSB.

Figure 1-2 ALSB Architecture



## Proxy Services and Business Services

ALSB 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 can configure using ALSB Console. Proxy services are ALSB definitions of intermediary web services that ALSB implements locally on WebLogic Server. With ALSB message brokering, service clients exchange messages with an intermediary proxy service rather than working directly with a business service.

ALSB allows you to implement proxy services independently and configure them dynamically, as driven by your business needs without requiring costly infrastructure development efforts. The configuration functions are separated from the management functions in ALSB Console. A proxy service can route messages to multiple business services; you can choose to configure a proxy service with an interface that is independent of the business services with which the proxy service communicates. In such cases, you can configure a message flow definition to route a message to

the appropriate business service and map the message data into the format required by the business service's interface.

For more information on proxy services, see [Proxy Services](#) in AquaLogic Service Bus Concepts and Architecture. For information about how to use the ALSB Console to configure proxy services and other resources, see [Using the AquaLogic Service Bus Console](#).

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

## Message Flows and Pipelines

In ALSB, a message flow is the implementation of a proxy service. You configure the logic for the manipulation of messages using proxy service message flow definitions. This logic includes such activities as transformation, publishing, and reporting, which are implemented as individual actions within the stages of a pipeline.

Pipelines are one-way processing paths that include no branching. A pipeline is a named sequence of stages containing actions, representing a non-branching one-way processing path. It is used to specify the message flow for service requests and responses. A stage is a user-configured processing step. Messages fed into the pipelines are accompanied by a set of message context variables that contain the message contents. They can be accessed or modified by actions in the pipeline stages.

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

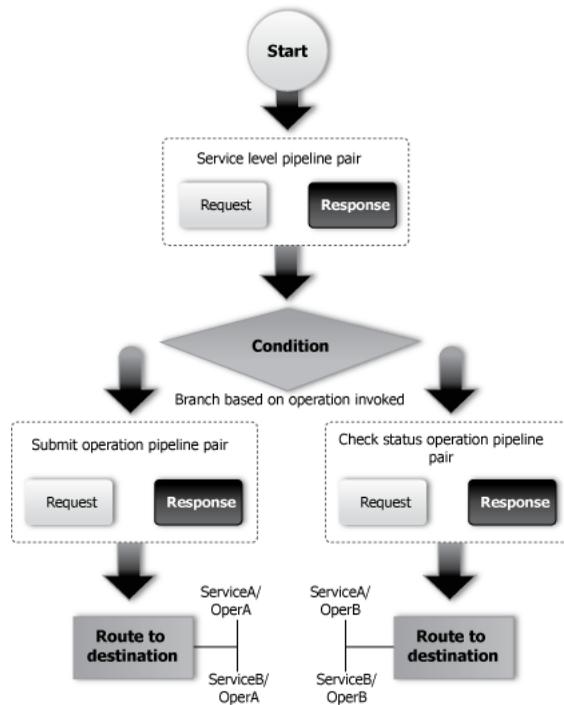
## Pipeline Pairs

Pipeline pairs are request and response pipelines. The request pipeline definition specifies the actions that ALSB 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 ALSB performs on responses from the business or proxy service that the proxy service invokes before returning a response to a client.

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

Figure 1-3 illustrates sample operation pipelines in a proxy service.

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



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

## Overview of the Tutorials

This tutorial shows how ALSB is used to route a loan application to appropriate business services based on the configurations set in the proxy service. ALSB Console comprises JSP-based portlets that support the configuration and design of the proxy service message flows and the associated resources.

In the tutorials, you will use the console to create the proxy service that will interact with the service client and business services. As you work through the tutorial, you will:

- Learn how to configure proxy services to process the input and output messages in a message flow.
- Create, read, and update configuration objects (such as proxy services and web services Development Language (WSDLs)).
- See how ALSB uses session management to allow configuration changes locally, and then instantiate the changes in the core environment.
- Configure resources such as WSDLs, Service Callouts, XQueries, and business services.
- Route messages according to XQuery-based policies or callouts to external web services.
- See how ALSB 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 ALSB is driven by metadata specified as the *message flow definition* for a proxy service in the ALSB Console.
- Test proxy services and business services.

## How to Use the Tutorials

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

- Provide you an understanding of the concepts underlying ALSB.
- Show you how to navigate through the ALSB Console and demonstrate the capabilities of the console.
- Provide the basic tasks that will help you to configure specific scenarios.

Complete the first tutorial, Routing a loan application, to become familiar with ALSB, then proceed to the other tutorials in any order. The following tutorials are provided as part of the tutorial suite:

- [Tutorial 1. Routing a Loan Application](#) is based on scenarios that demonstrate how ALSB facilitates the routing of messages within an enterprise.

- [Tutorial 2. Transforming a Loan Application](#) describes how a proxy service is configured with a routing node, transformations, and a web service callout (Service Callout) to allow content-based routing and message enrichment through ALSB.
- [Tutorial 3. Validating a Loan Application](#) describes how a ALSB 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

- [AquaLogic Service Bus Examples](#)
- [AquaLogic Service Bus Installation Guide](#)
- [AquaLogic Service Bus Concepts and Architecture](#)
- [Using the AquaLogic Service Bus Console](#)

## Introduction to the ALSB Tutorials

# Getting Started with the ALSB Tutorials

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

This topic consists of the following sections:

- [Scenarios in the Tutorials](#)
- [Setting Up the Tutorials](#)

After completing this section, you will know how to deploy a typical ALSB environment in which you can define resources and design web services. You must complete all the tasks in this section before you start the tutorials.

## Scenarios in the Tutorials

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

In the first tutorial, you will become familiar with the basic functionality of ALSB that facilitates web service mediation, including message routing and web service resource creation. In

subsequent tutorials you will develop and customize web services for routing, transforming and validating a loan application.

## Setting Up the Tutorials

Do the following to set up and run the tutorials:

1. Install ALSB
2. Create an ALSB Domain
3. Change Domain Configuration Settings
4. Start ALSB
5. Build and Deploy Business Services
6. Log into ALSB Consoles

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

```
BEA_HOME\alsb_3.0\samples\servicebus\examples
```

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

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

Filename/Location	Description
/src/setEnv.cmd	This file sets the environment properties.
/src/examples.properties	This file contains business service build and deployment properties.
For each business service:	
BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webservices\jws_basic\normal	The files contained in this directory are those required by the normalLoan business service. The same file structure exists for the other three business services used in this tutorial (ManagerLoanReview, LoanSaleProcessor, and CreditRating).

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

Filename/Location	Description
<code>SimpleBean.java</code>	The <code>jws-181</code> web service definition.
<code>/client/LoanStruct.java</code>	The definition of the message.
<code>build.xml</code>	The build script that is used to build each of the services using <code>ant</code> .
<code>*.wsdl</code>	A web service Definition Language (WSDL) file defines each of the services.
<code>/client/Main.java</code>	The Java client with which each of the business services interacts.

## Install ALSB

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

## Create an ALSB Domain

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

1. To start the Configuration Wizard in graphical mode, choose the Configuration Wizard option in the BEA Products group from the Windows Start menu:

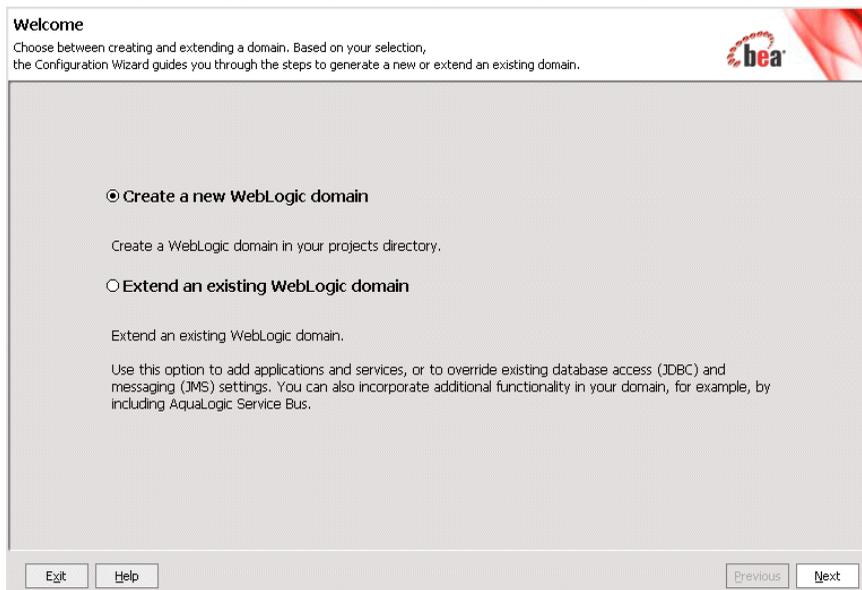
**Start > All Programs > BEA Products > Tools > Configuration Wizard**

2. Using the Configuration Wizard, select the following product components:
  - WebLogic Server
  - AquaLogic Service Bus

For more information, see [Creating WebLogic Domains Using the Configuration Wizard](#).

- a. In the **Welcome** page, the Create a new WebLogic domain radio button is selected by default. Accept the default selection and click **Next**.

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



- b. In the **Select Domain Source** page, select AquaLogic Service Bus check box. Click **Next**.

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

Select Domain Source

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

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

- WebLogic Server (Required)
- Workshop for WebLogic Platform
- AquaLogic Service Bus
- Workshop for WebLogic 10.2

**Base this domain on an existing template**

Template location:

- c. In the **Configure Administrator Username and Password** page, configure a new administrator username and password. You can use the default weblogic username, but you must create a unique password. Click **Next** to proceed to the **Configure Server Start Mode and JDK** page.

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

Configure Administrator Username and Password

Create a user to be assigned to the Administrator role.  
This user is the default administrator used to start development mode servers.

\*User name:

\*User password:

\*Confirm user password:

Description:

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

Figure 2-4 Configure Server Start Mode and JDK

**Configure Server Start Mode and JDK**

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

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

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

**WebLogic Domain Startup Mode**

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

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

**JDK Selection**

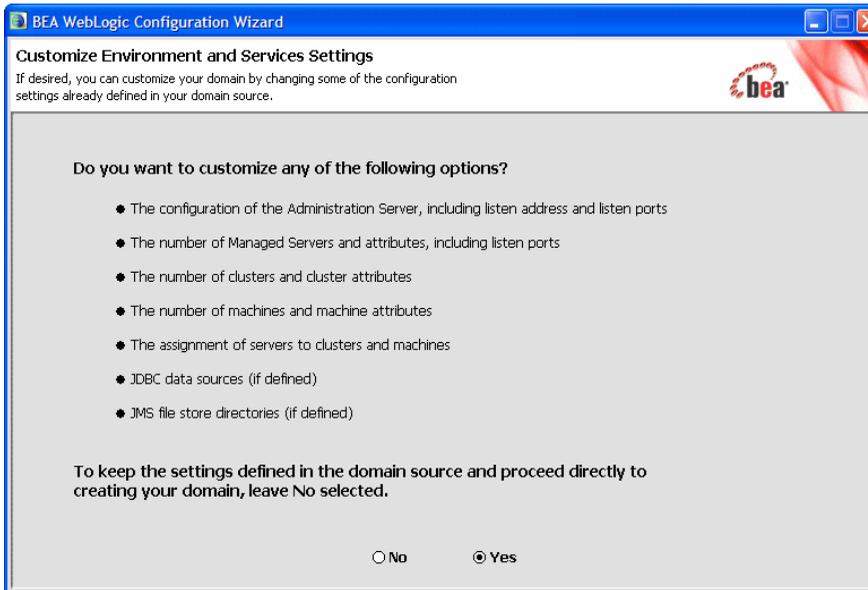
**BEA Supplied JDKs**

JRockit SDK 1.5.0\_11 @ D:\beascreenNew\jrockit\_150\_11  
Sun SDK 1.5.0\_11 @ D:\beascreenNew\jdk150\_11

**Other JDK**  
Location:

- e. In the **Customize Environment and Services Settings** page, the default customization option for the environment and services settings is No. Click **Yes** to change the **Listen Port** in **Configure the Administrator Server**.

Figure 2-5 Customize Environment and Service Settings



- f. Set the value for **Listen Port** to 7001.

Figure 2-6 Configure the Administrator Server

**Configure the Administration Server**

Enter administration server configurations. Each WebLogic Server domain must have one Administration Server. The Administration Server hosts the Administration Console which is used to perform administrative tasks.

Discard Changes

\*Name:

Listen address:

Listen port:

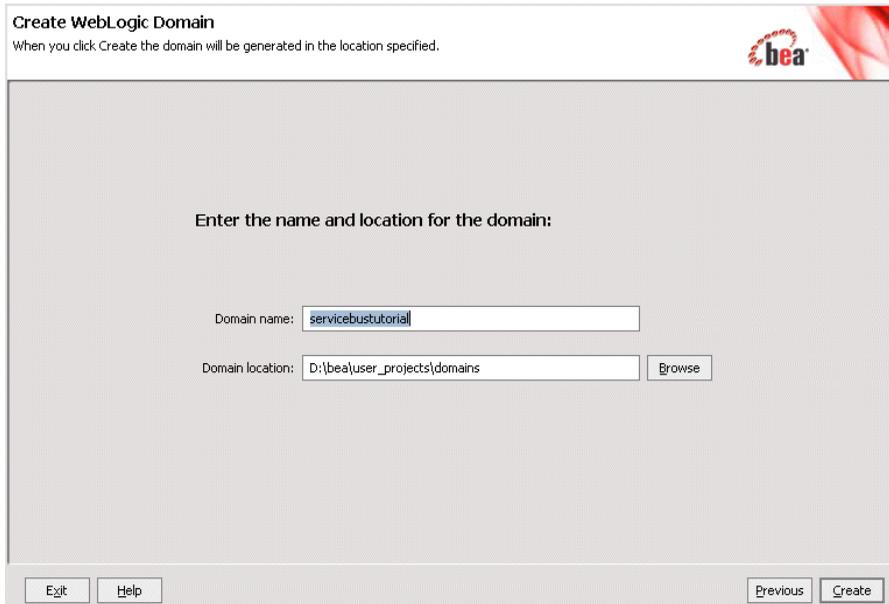
SSL listen port:

SSL enabled:

Exit Previous Next

- g. Click **Next** to proceed **Configure Managed Servers** page. Accept the default options for:
- Configure Managed Server
  - Configure Machines
  - Configure JDBC Data Sources
  - Run Database Scripts
  - Configure JMS Stores
- Click **Next** on **Review WebLogic Domain** to go to **Create WebLogic Domain** page.
- h. In the **Create Weblogic Domain** page specify the domain name as ServiceBusTutorial. Click **Create**.

**Figure 2-7 Configuration Wizard - Create Weblogic Domain**



- i. When the domain is created successfully, Do not click **Start Admin Server**. Click **Done** to finish.

## Start ALSB

You can start ALSB using one of the following methods:

- On Windows systems, from the Windows Start menu, select:  
**Start > All Programs > BEA Products > User Projects > *domain\_name* > Start Server.**  
For example, to start the server for the Servicebustutorial domain select **Start > All Programs > BEA Products > User Projects > Servicebustutorial > Start Server**
- On Windows systems, invoke startWebLogic.cmd from the root of the ALSB domain.
- On UNIX systems, invoke startWebLogic.sh from the root of the ALSB domain.

When ALSB is started, a server command console window will display status information about WebLogic Server.

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

## Load the Client Application

To test the loan application using the test console, you must first deploy the client jars in the WLS console, which are available at `{BEA_HOME}\alsb_3.0\samples\servicebus\examples\build\webservices`. The following client jars are available:

- `creditLoan_jws_basic_ejb`
- `largeLoan_jws_basic_ejb`
- `manager_jws_basic_ejb`
- `normalLoan_jws_basic_ejb`

To deploy `creditLoan_jws_basic_ejb` client jar:

1. Click **WLS Console** on the dashboard to go to the WLS Console.
  - a. In left pane click **Lock and Edit** in the **Change Center**.
  - b. Click on **Deployments in Domain Structure** to go to the **Summary of Deployments** page (see [Figure 2-8](#)).

**Figure 2-8 Summary of Deployments**

**Summary of Deployments**

Control [Monitoring](#)

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

**Deployments**

Showing 1 - 10 of 66 Previous | Next

<input type="checkbox"/>	Name	State	Type	Deployment Order
<input type="checkbox"/>	ALDSP Transport Provider	Active	Web Application	161
<input type="checkbox"/>	aldsp_transport-110n	Active	Library	160
<input type="checkbox"/>	ALSB Cluster Singleton Marker Application	Active	Enterprise Application	100
<input type="checkbox"/>	ALSB Domain Singleton Marker Application	Active	Enterprise Application	101
<input type="checkbox"/>	ALSB Framework Starter Application	Active	Enterprise Application	102
<input type="checkbox"/>	ALSB Logging	Active	Web Application	440
<input type="checkbox"/>	ALSB Publish	Active	Web Application	430
<input type="checkbox"/>	ALSB Resource	Active	Web Application	500
<input type="checkbox"/>	ALSB Routing	Active	Web Application	410
<input type="checkbox"/>	ALSB Subscription Listener	Active	Enterprise Application	123

Showing 1 - 10 of 66 Previous | Next

- c. Click **Install**. Use **Install Application Assistant** (see [Figure 2-9](#)) to install the client jar you want to deploy.

Figure 2-9 Install Application Assistant

- d. Use this to navigate to  
`BEA_HOME\alsb_3.0\samples\servicebus\examples\build\webservices`
- e. Click the radio button associated with `creditLoan_jws_basic_ejb` (see Figure 2-10).  
 Click **Next** to choose the targeting style.

Figure 2-10 Choose Client Jar

- f. In **Choose Targeting Style** page accept defaults. Click **Next**.

**Figure 2-11 Choose Targeting Styles**

The screenshot shows a dialog box titled "Install Application Assistant". At the top, there are four buttons: "Back", "Next", "Finish", and "Cancel". Below the buttons is a dashed line. The main content area is titled "Choose targeting style" and contains the following text: "Targets are the servers, clusters, and virtual hosts on which this deployment will run. There are several ways you can target an application." There are two radio button options: "Install this deployment as an application" (which is selected) and "Install this deployment as a library". Below the second option is a descriptive sentence: "Application libraries are deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications." At the bottom of the dialog box, there are four buttons: "Back", "Next", "Finish", and "Cancel".

- g. In **Choose Targeting Style** (see [Figure 2-11](#)) page accept defaults. Click **Next** to go to **Optional Settings** page.
- h. In **Optional Settings** (see [Figure 2-12](#)) page accept defaults and click **Finish**. Click **Activate the Session**.

Figure 2-12 Optional Settings

**Install Application Assistant**

Back Next Finish Cancel

**Optional Settings**  
You can modify these settings or accept the defaults

**General**

What do you want to name this deployment?

Name: creditLoan\_jws\_basic\_ejb

**Security**

What security model do you want to use with this application?

DD Only: Use only roles and policies that are defined in the deployment descriptors.

Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.

Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.

Advanced: Use a custom model that you have configured on the realm's configuration page.

**Source accessibility**

How should the source files be made accessible?

Use the defaults defined by the deployment's targets

Recommended selection.

Copy this application onto every target for me

During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

I will make the deployment accessible from the following location

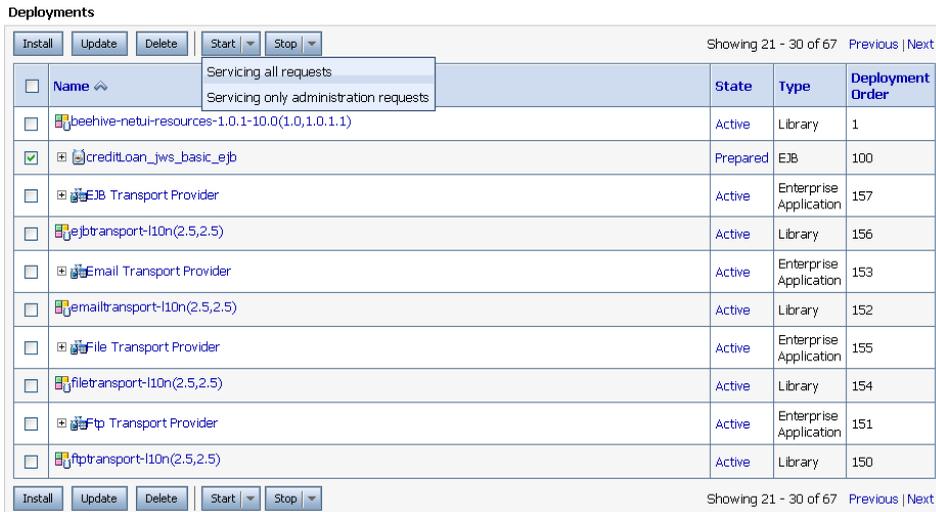
Location: D:\bea\wls\server\_10.0\samples\servicebus\examples\build

Provide the location from where all targets will access this application's files. This is often a shared directory. You must ensure the application files exist in this location and that each target can reach the location.

Back Next Finish Cancel

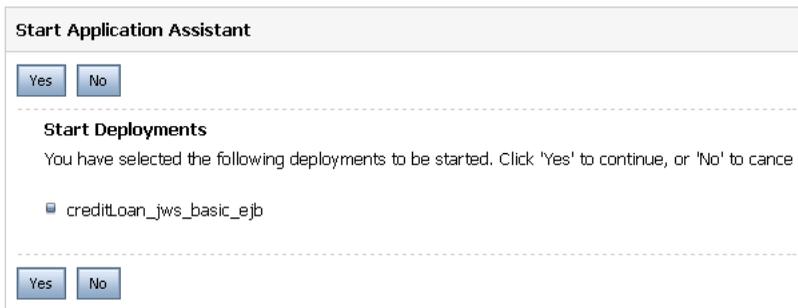
2. Click **Lock and Edit** to create a new session. In the **Summary of Deployments** page locate the client jar you have installed in Deployments table. Click on the radio button associated with the client jar and click **Start**. Choose Servicing all requests from the drop-down-list (see [Figure 2-13](#)).

Figure 2-13 Deployments



3. In **Start Deployments** page click **Yes** (see [Figure 2-14](#)).

Figure 2-14 Start Deployments



For more information on deploying and starting client jars in WLS console, see [Enterprise Applications](#) in *Administration Console Online Help*.

You must repeat [step 1](#) to [step 3](#) to deploy `largeLoan_jws_basic_ejb`, `manager_jws_basic_ejb`, and `normalLoan_jws_basic_ejb` in the WLS console.

## Log in to ALSB Console

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

### To log in to ALSB Console

1. Open a browser window and enter the following URL:

`http://host:port/sbconsole`

where `host:port` represents the name and port number of the machine on which WebLogic Server is running. The default port for WLS is 7001.

Enter `http://localhost:7001/sbconsole` URL in your browser to open the ALSB console for the Servicebustutorial domain.

2. The ALSB Console login page is displayed. Enter the username and password that you specified when you created the ALSB tutorial domain.

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

**Start > All Programs > BEA Products > User Projects > Servicebustutorial > AquaLogic Service Bus Admin Console** for ALSB Admin Console

## Where to Go from Here

After you complete the tasks required to set up the tutorials, you can proceed to [Tutorial 1. Routing a Loan Application](#), which describes how you can configure AquaLogic Service Bus with the resources required for the loan application routing scenario. Each of the tutorials include instructions to design and configure the ALSB resources, and procedures you can use to test the completed configurations.

# Tutorial 1. Routing a Loan Application

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

This section includes the following topics:

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

## Prerequisites

You must complete [Getting Started with the ALSB Tutorials](#) before beginning this tutorial.

## Tutorial Objectives

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

- Importing Web Service Definition Language (WSDL) files

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

This tutorial introduces you to:

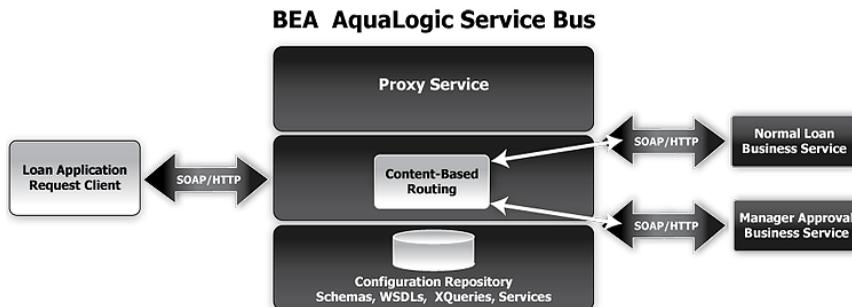
- Using ALSB Console
- Implementation details of proxy services: message flows and actions
- Editing XQuery expressions using the XQuery Expression Editor
- Working with context variables to manipulate the content of the message

## Definition of the Scenario

A primary mortgage company uses ALSB 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 the appropriate business service for processing.

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

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



## Overview of the Run-Time Process in ALSB

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 application is routed to the ManagerLoanReview business service; otherwise it is routed to the NormalLoan business service. The target business service returns a response similar to that shown in [Figure 3-42](#).

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

## Required Resources

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

**Table 3-1 Routing Loan Application Tutorial Resources**

Resource Name	Description
normalLoan	This is the WSDL resource.
NormalLoan	This is business service used by ALSB.
ManagerLoanReview	This is business service used by ALSB.
LoanGateway1	This is the ALSB proxy service.

## Tasks in This Tutorial

In this tutorial, ALSB is used to route a loan application within a mortgage company to a target a web service depending on the interest rate requested. Perform the following tasks to design and configure a proxy service and the associated resources in ALSB to implement this use case scenario.

### Prepare the Environment

Ensure that ALSB is running in the domain that you have created for the tutorial and that you have completed the tasks described in [Setting Up the Tutorials](#).

### Create a Session and Set up a Project

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

The Change Center in the console allows you to create and manage sessions. All updates to the system configuration during the current session are saved as temporary files. These changes are effective only when the current session is activated. [Figure 3-2](#) illustrates the Change Center pane available on ALSB Console.

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



In addition to creating and activating sessions, the Change Center allows you to perform the functions summarized in the [Table 3-2](#).

**Table 3-2 Change Center Options**

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

**Note:** It is recommended that you click **Create** before modifying any system configuration settings on the console. It is a good practice to save your configuration changes

incrementally. To save and activate configuration changes in the current session, you must click **Activate** after you have completed making a set of changes.

## To create the MortgageBroker Project and File Folders

In the ALSB system environment, system configuration entities are grouped into user-defined projects. On ALSB Console, the Project Explorer page displays a default project and the various user defined projects under the Projects folder. This page allows you to perform all project configuration tasks. Each project is represented by a project folder on the console. You can add new project folders and navigate the project trees. You can collapse and expand folders as required by clicking them.

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

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

- ProxyService
- BusinessService
- WSDL

### To Start a Session and Create a Project

1. In the ALSB console Change Center click **Create** to create a new session.
2. In the ALSB Console navigation pane, select the **Project Explorer**.

The project explorer opens in the navigation pane and a Project page is displayed on the console.

3. In the **Enter New Project Name** field in the Projects section, type `MortgageBroker` as shown in [Figure 3-3](#).

**Figure 3-3 Create a Project**



Projects	
▶ Enter New Project Name:	MortgageBroker <span style="float: right;">Add Project</span>

4. Click **Add Project**.

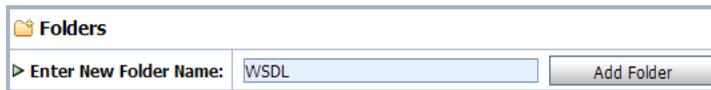
The MortgageBroker project is created and listed in the Project Explorer under Projects.

You must create a project folder WSDL 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 the project explorer, click the MortgageBroker project to open the associated project page.
2. On the Project page, in **Folders**, enter the folder name in the field provided. In this case, enter WSDL as displayed in [Figure 3-4](#).

**Figure 3-4 Create a Folders**



3. Click Add Folder.  
The WSDL folder is displayed in the list of project folders for the MortgageBroker project.
4. Repeat [step 2](#) and [step 3](#) to create the following additional folders: ProxyService and BusinessService.
5. When all three folders are created, click **Activate**, then **Submit** to save the project directory structure. The Change Center environment setting options are listed in [Table 3-2](#).

## Create the WSDL Resources

ALSB 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.

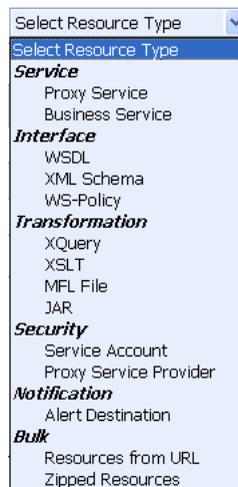
### To Create the normalLoan WSDL Resource

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

## To Import a WSDL

1. In the ALSB console Change Center click **Create** to create a new session.
2. In ALSB Console navigation pane, select the Project Explorer.
3. In the project explorer, click `MortgageBroker` to expand the folder, then click `WSDL` to open the WSDL folder.
4. In the Resources pane, from the Select Resource Type drop-down list, select `WSDL`.

**Figure 3-5 Select a Resource Type**



The Create a New WSDL Resource page is displayed.

5. Enter the following information in the Create a New WSDL Resource page (see [Figure 3-6](#)):
  - a. Enter `normalLoan` as the resource name.
  - b. Click **Browse...** . Select the WSDL associated with the normal loan process:
 

```
BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webservices\jws_basic\normal\NormalLoanApprovalService.wsdl
```
  - c. Optionally, enter information in the Resource Description field to describe the resource.
  - d. The resource is now displayed in the resources pane. The displayed **Create a New WSDL Resource** page must be similar to the page shown in [Figure 3-6](#).

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

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

This step completes the creation of the normalLoan WSDL resource in the WSDL folder. Activate the session. For more information on how to activate a session, see [Table 3-2](#).

## Create the managerApproval WSDL Resource

Complete tasks in [step 1](#) to [step 5](#), to create a WSDL resource associated with the ManagerApproval Service, using the procedure in [To Import a WSDL](#) section. To import the ManagerApprovalService WSDL resource, you must use the configuration parameters listed in the following table.

**Table 3-3 ManagerApproval Service WSDL**

Configuration Parameter	Value
Resource Name	managerApproval
ManagerApprovalService WSDL	BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webservices\jws_basic\manager\ManagerApprovalService.wsdl

## Create a Proxy Service

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

### To Create the Proxy Service

1. In the ALSB console Change Center click **Create** to create a new session.

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

The Create a Proxy Service - General Configuration page is displayed as shown in [Figure 3-7](#).

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

Create a Proxy Service (MortgageBroker/ProxyServices/)	
General Configuration	
<b>Service Name*</b>	<input type="text" value="LoanGateway1"/>
<b>Description</b>	<input type="text"/>
<b>Service Type*</b>	<p><b>Create a New Service</b></p> <p><input checked="" type="radio"/> WSDL Web Service <input type="text" value="MortgageBroker/WSDL/normalLoan"/> <input type="button" value="Browse..."/></p> <p><input type="text" value="helloPort"/> (port)</p> <p><input type="radio"/> Messaging Service</p> <p><input type="radio"/> Any SOAP Service <input type="text" value="SOAP 1.1"/></p> <p><input type="radio"/> Any XML Service</p> <p><b>Create From Existing Service</b></p> <p><input type="radio"/> Business Service <input type="text"/> <input type="button" value="Browse..."/></p> <p><input type="radio"/> Proxy Service <input type="text"/> <input type="button" value="Browse..."/></p>
<b>Content Streaming</b>	<p><input type="checkbox"/> Enabled</p> <p><input type="radio"/> Memory Buffer</p> <p><input type="radio"/> Disk Buffer</p>
<b>Publish To Registry</b>	<input type="checkbox"/>
<input type="button" value="Next &gt;&gt;"/>   <input type="button" value="Last &gt;&gt;"/>   <input type="button" value="Cancel"/>	

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

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

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

**Figure 3-8 Select WSDL Definition**

Select a WSDL definition

Search: Name:  Path:    [Adv. Search](#)

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

Description:

▶ Select WSDL definitions

```
Bindings
  NormalLoanApprovalServiceSoapBinding
Ports
  helloPort
```

<< Back | Submit | Cancel

- a. In Select WSDL Definitions pane, from the Ports category, Click `helloPort`, which is the WSDL port for the normalLoan WSDL (helloPort).
- b. Click **Submit**. The port name is displayed in the WSDL port text field as shown in [Figure 3-9](#).



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

Create a Proxy Service (Mortgage Broker /Proxy Service/LoanGateway1)					
Transport Configuration					
Protocol*	http				
Endpoint URI*	Format: /someName /loan/gateway1				
Get All Headers	<input type="radio"/> Yes <input checked="" type="radio"/> No Header <input type="text"/> <input type="button" value="Add"/> <table border="1"> <thead> <tr> <th>HEADER</th> <th>ACTION</th> </tr> </thead> <tbody> <tr> <td colspan="2">There are no headers configured.</td> </tr> </tbody> </table>	HEADER	ACTION	There are no headers configured.	
HEADER	ACTION				
There are no headers configured.					

<< Prev.      Next >>      Last >>      Cancel

13. Accept the default settings for the HTTP transport configuration in **HTTP Transport Configuration** page (see [Figure 3-11](#)).

**Figure 3-11 HTTP Transport Configuration**

Create a Proxy Service (MortgageBroker/ProxyService/LoanGateway1)	
HTTP Transport Configuration	
HTTPS required	<input type="checkbox"/>
Authentication	<input checked="" type="radio"/> None <input type="radio"/> Basic <input type="radio"/> Client Certificate <input type="radio"/> Custom Authentication (See Advanced Settings)
Dispatch Policy	default
Request Encoding	<input type="text"/>
Response Encoding	<input type="text"/>
Advanced Settings <input type="button" value="v"/>	

<< Prev.      Next >>      Last >>      Cancel

The Create a Proxy Service - Operation Selection Configuration page is displayed as shown in [Figure 3-12](#). On this page you will select the metadata accompanying the message, which identifies the web service operation to be invoked.

**Figure 3-12 Operation Selection Configuration of Proxy Service**

Create a Proxy Service (Mortgage Broker/Proxy Service/LoanGateway1)	
Operation Selection Configuration	
Enforce WS-I Compliance	<input type="checkbox"/>
Selection Algorithm	<input type="radio"/> Transport Header <input type="radio"/> SOAPAction Header <input type="radio"/> WS-Addressing <input type="radio"/> SOAP Header <input checked="" type="radio"/> SOAP Body Type
<input type="button" value=" &lt;&lt; Prev."/> <input type="button" value=" Next &gt;&gt;"/> <input type="button" value=" Last &gt;&gt;"/> <input type="button" value=" Cancel"/>	

14. Accept the default selection algorithm (SOAP Body Type). Click **Next**. The **Create a Proxy Service–Summary** page is displayed as shown in the [Figure 3-13](#). This page shows a summary of configuration settings for the proxy service.

**Note:** The default service is an RPC web service. As a result, the SOAP body contains the service operation selection information. An operation in a Java web service WSDL corresponds to a public method in the Java web service. If there are multiple public methods accessible to a client, each public method type will have an operation definition in the WSDL. For a SOAP based RPC service, the SOAP body indicates the operation (method) selected by the client. The SOAP header can also specify the operation. However, by convention, the SOAP body defines this.

**Figure 3-13 Proxy Service - Configuration Settings Summary**

Create a Proxy Service - Summary (MortgageBroker/ProxyServices/LoanGateway1)	
<b>General Configuration</b> 	
Service Name	LoanGateway1
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Content Streaming	Disabled
<b>Transport Configuration</b> 	
Protocol	http
Endpoint URI	/loan/gateway1
Get All Headers	No
Headers	
<b>HTTP Transport Configuration</b> 	
HTTPS required	No
Authentication	None
<b>Operation Selection Configuration</b> 	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type
<input data-bbox="212 963 354 987" type="button" value=" &lt;&lt; Prev. "/> <input data-bbox="401 963 544 987" type="button" value=" Save "/> <input data-bbox="591 963 733 987" type="button" value=" Cancel "/>	

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

- After reviewing the proxy service configuration settings, click **Save** to register the service.

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

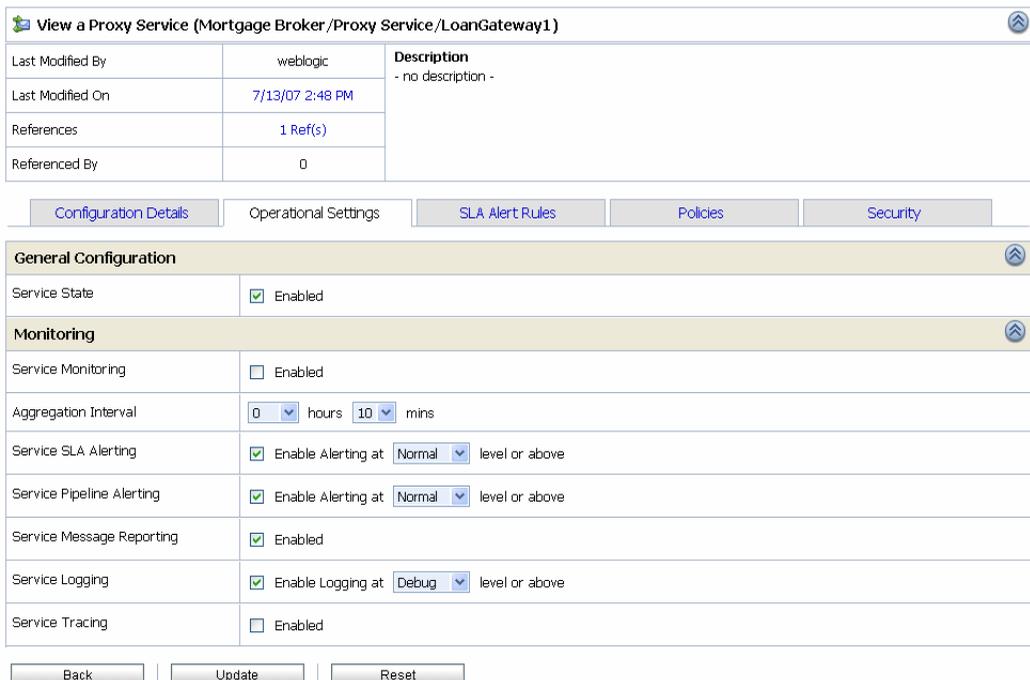
The LoanGateway1 proxy service is displayed in the **Resources** pane in the ProxyService folder of the MortgageBroker project. Click the LoanGateway1 link in the Resources pane of MortgageBroker/ProxyServices to go the View a Proxy Service page.

**Figure 3-14 Proxy Services in Resource Pane**



16. The View a Proxy Service page is displayed as shown in [Figure 3-15](#). The Operational Settings tab on this page has options to enable monitoring for the proxy service.

**Figure 3-15 Monitoring Configuration of Proxy Service**



To enable monitoring:

- a. Click the **Operational Settings** tab.
- b. Select the **Enabled** check box for the **Monitoring** setting.

- c. Choose an aggregation interval for the service. The aggregation interval is the period over which aggregated statistics related to the service are computed for display in the ALSB Console dashboard. You can also accept the default value for the Aggregation Interval, which is set to five minutes.

For more information about using the ALSB Dashboard and monitoring services, see: [Monitoring](#) in Using the AquaLogic Service Bus Console and AquaLogic Service Bus Operations Guide.

- d. Click **Update**.
- e. Click the **Back** button at the bottom of the **View a Proxy Service** page, to return to the **MortgageBroker/ProxyService** folder page.

This step completes the configuration of the LoanGateway1 proxy service in ALSB Console. Activate the session. For more information, see [Table 3-2](#).

## Create the Business Services

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

Complete this step to:

- [Create the NormalLoan Business Service](#)
- [Create the ManagerLoanReview Business Service](#)

### Create the NormalLoan Business Service

1. In the ALSB console **Change Center** click **Create** to create a new session
2. In the Project Explorer, select the **BusinessService** folder from the **MortgageBroker** project tree.
3. In the Select Resource Type drop-down list, select **Business Service**.

The Create a Business Service - General Configuration page is displayed as shown in [Figure 3-16](#).

Figure 3-16 Create Business Service

**Create a Business Service (MortgageBroker/BusinessService/)**

**General Configuration**

**Service Name\*** NormalLoan

**Description**

**Service Type\***

**Create a New Service**

WSDL Web Service MortgageBroker/WSDL/normalLoan   
helloPort (port)

Transport Typed Service

Messaging Service

Any SOAP Service SOAP 1.1

Any XML Service

**Create From Existing Service**

Business Service

Proxy Service

4. Enter a name for the service. In this case, enter NormalLoan.
5. For Service Type, select WSDL web service, then click **Browse...** . Select the WSDL and port on which to base the business service:
  - a. In the WSDL Browser, select the normalLoan WSDL
  - b. Select the port (helloPort) for the WSDL.
  - c. Click **Submit**.

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

6. Click **Next**.

The Create a **Business Service - Transport Configuration** page is displayed as shown in [Figure 3-17](#).

**Figure 3-17 Transport Configuration of a Business Service**

Create a Business Service (default/normal)					
Transport Configuration					
Protocol*	http				
Load Balancing Algorithm	none				
Endpoint URI*	Format: http://host:port/someService <input type="text" value="http://"/> <input type="button" value="Add"/> <table border="1"> <thead> <tr> <th>EXISTING URIS</th> <th>OPTIONS</th> </tr> </thead> <tbody> <tr> <td>http://localhost:7001/njws_basic_ejb/NormalSimpleBean</td> <td> <input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/> </td> </tr> </tbody> </table>	EXISTING URIS	OPTIONS	http://localhost:7001/njws_basic_ejb/NormalSimpleBean	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>
EXISTING URIS	OPTIONS				
http://localhost:7001/njws_basic_ejb/NormalSimpleBean	<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="✖"/>				
Retry Count	0				
Retry Iteration Interval	30				
Retry Application Errors	<input checked="" type="radio"/> Yes <input type="radio"/> No				
<input type="button" value=" &lt;&lt; Prev."/> <input type="button" value=" Next &gt;&gt;"/> <input type="button" value=" Last &gt;&gt;"/> <input type="button" value=" Cancel"/>					

- Accept the default protocol, http.
- Select none for the Load Balancing Algorithm, since load balancing is relevant only when a service specifies multiple end points.

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

- The End Point URI of the business service is the endpoint URI on the server on which the service is deployed. Ensure that the pre-populated value for the End point URI in the Existing URIs field is http://<host:port>/njws\_basic\_ejb/NormalSimpleBean.

The host and port values specified for <host:port> represent the machine and port on which your ALSB server is running.

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

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

The Create a Business Service-HTTP Transport Configuration page is displayed as shown in [Figure 3-18](#).

Figure 3-18 HTTP Transport Configuration of a Business Service

Create a Business Service (Mortgage Broker/Business Service/NormalLoan)	
HTTP Transport Configuration	
Timeout	<input type="text" value="0"/>
HTTP Request Method	<input checked="" type="radio"/> POST <input type="radio"/> GET
Authentication	<input checked="" type="radio"/> None <input type="radio"/> Basic <input type="radio"/> Client Certificate
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=" &lt;&lt; Prev."/> <input type="button" value=" Next &gt;&gt;"/> <input type="button" value=" Last &gt;&gt;"/> <input type="button" value=" Cancel"/>	

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

The Create a Business Service-SOAP Binding Configuration page is displayed as shown in [Figure 3-19](#).

Figure 3-19 SOAP Binding Configuration of Business Service

Create a Business Service (Mortgage Broker/Business Service/NormalLoan)	
SOAP Binding Configuration	
Enforce WS-I Compliance	<input type="checkbox"/>
<input type="button" value=" &lt;&lt; Prev."/> <input type="button" value=" Next &gt;&gt;"/> <input type="button" value=" Last &gt;&gt;"/> <input type="button" value=" Cancel"/>	

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

The Create a Business Service-Summary page is displayed as shown in [Figure 3-20](#). You can now check if all the settings of the business service have been implemented correctly.

**Figure 3-20 Business Service - Configuration Settings Summary**

Create a Business Service - Summary (MortgageBroker/BusinessService/NormalLoan)	
<b>General Configuration</b>	
Service Name	NormalLoan
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/BusinessService/normalLoan, port="helloPort")
<b>Transport Configuration</b>	
Protocol	http
Load Balancing Algorithm	none
Endpoint URI	http://localhost:7001/njws_basic_ejb/NormalSimpleBean.
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes
<b>HTTP Transport Configuration</b>	
Timeout	0
HTTP Request Method	POST
Authentication	None
Follow HTTP redirects	ENABLED
<b>SOAP Binding Configuration</b>	
Enforce WS-I Compliance	No
<input data-bbox="185 1038 323 1058" type="button" value=" &lt;&lt; Prev. "/> <input data-bbox="373 1038 512 1058" type="button" value=" Save "/> <input data-bbox="561 1038 700 1058" type="button" value=" Cancel "/>	

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

To turn on monitoring for the NormalLoan business service, click NormalLoan business service in the **Resources** pane. The View a Business Service page is displayed as shown in [Figure 3-21](#). The Operational Settings tab on this page has options to enable monitoring for the business service.

Figure 3-21 View Business Service Page

View a Business Service (MortgageBroker/BusinessService/NormalLoan)		
Last Modified By	weblogic	Description - no description -
Last Modified On	12/5/07 3:39 PM	
References	1 Ref(s)	
Referenced By	0	
<div style="display: flex; justify-content: space-between;"> <span>Configuration Details</span> <span>Operational Settings</span> <span>SLA Alert Rules</span> <span>Policies</span> </div>		
General Configuration		
State	<input checked="" type="checkbox"/> Enabled	
Offline Endpoint URIs	<input type="checkbox"/> Enable with Retry Interval <input type="text" value="0"/> hours <input type="text" value="0"/> mins <input type="text" value="0"/> secs	
Monitoring		
Monitoring	<input type="checkbox"/> Enabled	
Aggregation Interval	<input type="text" value="0"/> hours <input type="text" value="10"/> mins	
SLA Alerting	<input checked="" type="checkbox"/> Enable Alerting at <input type="text" value="Normal"/> level or above	
Throttling		
Throttling State	<input type="checkbox"/> Enabled	
Maximum Concurrency	<input type="text" value="0"/>	
Throttling Queue	<input type="text" value="0"/> messages	
Message Expiration	<input type="text" value="0"/> msecs	
<div style="display: flex; justify-content: space-around;"> <span>Back</span> <span>Update</span> <span>Reset</span> </div>		

To enable monitoring:

- a. Click the **Operational Settings** tab.
- b. Select the **Enable Monitoring** check box.
- c. Choose an aggregation interval for the service. The aggregation interval is the period over which aggregated statistics related to the service are computed for display in the ALSB Console dashboard. You can also accept the default value for the Aggregation Interval, which is set to five minutes.

For more information about using the ALSB Dashboard and monitoring services, see [Monitoring](#) in Using the AquaLogic Service Bus Console and [Monitoring ALSB at Runtime](#) AquaLogic Service Bus Operations Guide.

Click **Update**.

- d. Click **Back** in the **View a Business Service** page to return your browser to the MortgageBroker project, BusinessService folder page.

You have completed the configuration of the NormalLoan business service on ALSB Console. Activate the session. For more information, see [Table 3-2](#).

### Create the ManagerLoanReview Business Service

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

Follow the same tasks as described in the preceding section: [Create the NormalLoan Business Service](#), but use the configuration parameters in the following table.

**Table 3-4 ManagerLoanReview Business Service**

Configuration Parameter	Value
Business Service Name	ManagerLoanReview
ManagerApprovalService WSDL	Select the managerApproval WSDL you created based on: BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webervices\jws_basic\manager\ManagerApprovalService.wsdl  <b>Note:</b> Select the helloPort port.
EndpointURI	http://<host:port>/mjws_basic_ejb/ManagerSimpleBean
Load Balancing Algorithm	none

### Summary

When you complete all the tasks from [Prepare the Environment](#) to [Create the Business Services](#), you have created two business services (NormalLoan and ManagerLoanReview), a proxy service (LoanGateway1) and two WSDLs (normalLoan and ManagerApprovalService) that are the resources required for this tutorial.

Once you have configured the proxy service with a base configuration you can proceed to the next task ([Configure the LoanGateway1 Proxy Service](#)) to complete the configuration of the proxy service by adding the routing behavior for the loan application.

## Configure the LoanGateway1 Proxy Service

ALSB 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.

### Configure the Routing Behavior of the Message Flow

1. In the ALSB console **Change Center** click **Create** to create a new session.
2. In the Project Explorer, select the ProxyService folder from the MortgageBroker project tree.
 

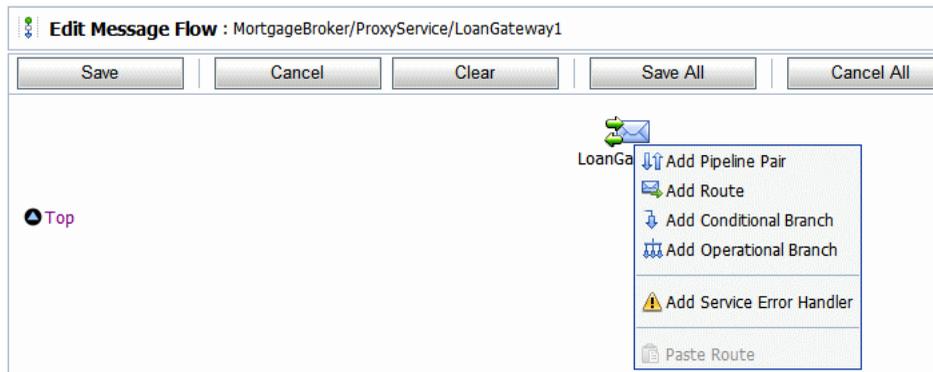
**Note:** You must be in a session to edit resources. If you have not already done so, click **Create** in the change center to begin a session to configure the proxy service and edit the message flow.
3. In the Actions column associated with the LoanGateway1 proxy service, double click the Edit Message Flow icon. The **Edit Message Flow** page for the proxy service LoanGateway1 is displayed as shown in [Figure 3-22](#).

**Figure 3-22 Edit Message Flow for LoanGateway1 Proxy Service**



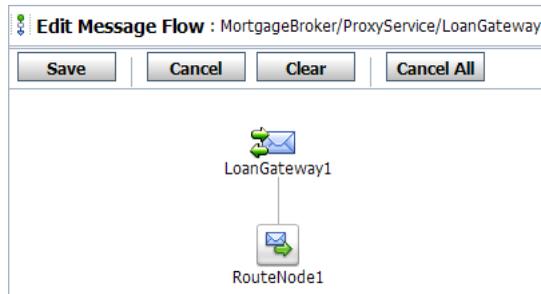
4. Click LoanGateway1, and select **Add Route** from the short-cut menu options.

Figure 3-23 Configure Message Flow for LoanGateway1 Proxy Service



RouteNode1 is added in the configuration page as shown in [Figure 3-24](#).

Figure 3-24 Add a Route Node

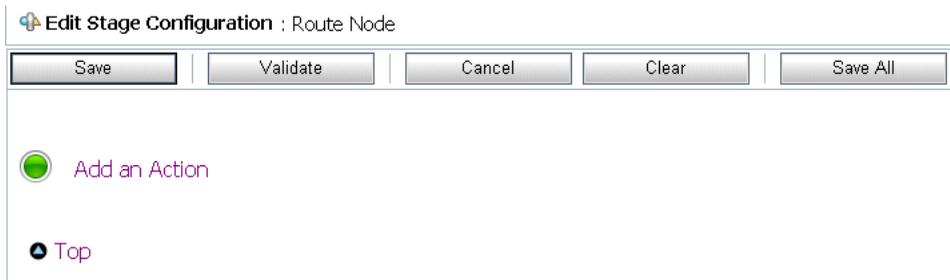


5. Click RouteNode1 and select **Edit > Route** from the short-cut menu options.

The Edit Stage Configuration: Route Node page is displayed as shown in [Figure 3-25](#). This page contains a single Add an Action link.

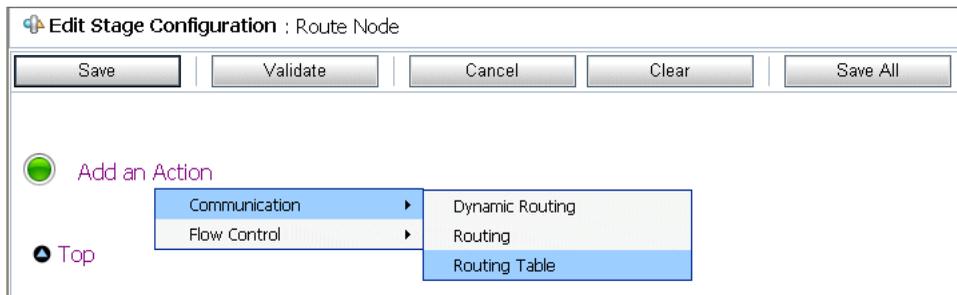
**Note:** A stage is an element of a pipeline and also a container for actions defined in a pipeline. Actions are the elements of a pipeline stage that define the handling of messages as they flow through a proxy service at run time.

**Figure 3-25 Add an Action Link**



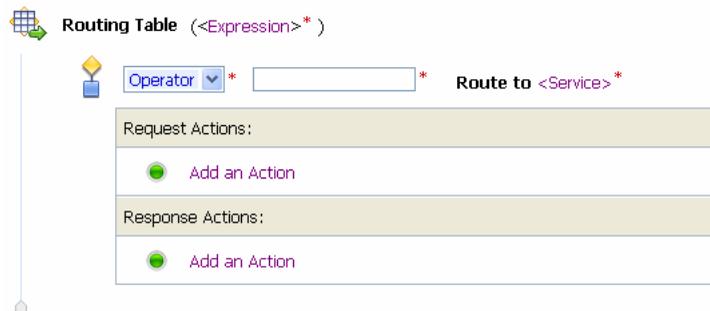
6. Click the **Add an Action** link, and select the **Communication > Routing Table** from the short-cut menu options.

**Figure 3-26 Add a Routing Table Element**



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

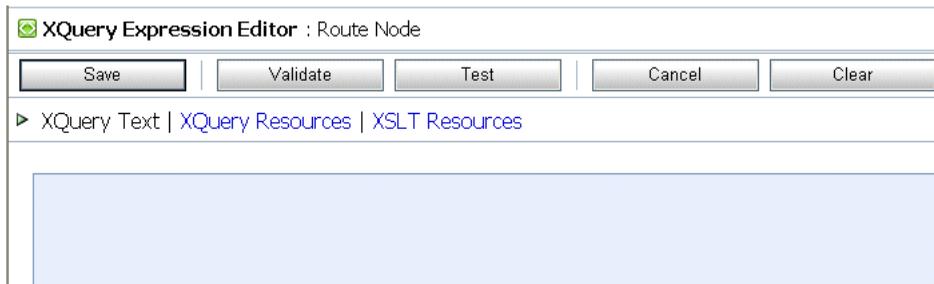
**Figure 3-27 Routing Table Configuration Page**



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

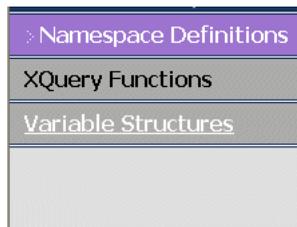
7. In the Routing table, click **<Expression>**. The XQuery Expression Editor is displayed.

**Figure 3-28 XQuery Expression Editor**



8. In the left pane, click **Variable Structures**.

**Figure 3-29 Namespace Definitions Pane**



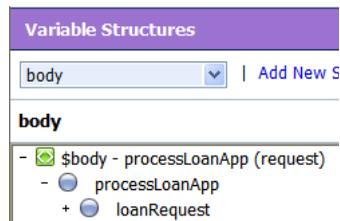
9. The Variable Structures pane is displayed. Select body from the Select Structure drop-down list in the Variable Structures pane.

**Figure 3-30 Variable Structures Pane**



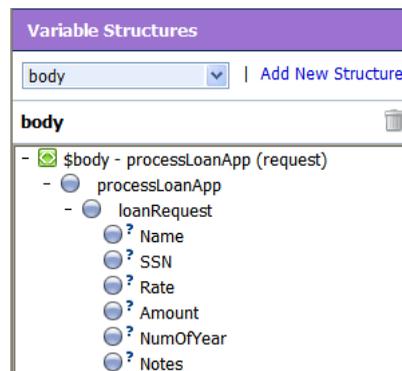
10. A structural representation of the body element is displayed in the Variable Structures pane.
11. Click + to expand the processLoanApp element, within the \$body - processLoanApp (request) element.

**Figure 3-31 processLoanApp Element**



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

**Figure 3-32 loanRequest Element**

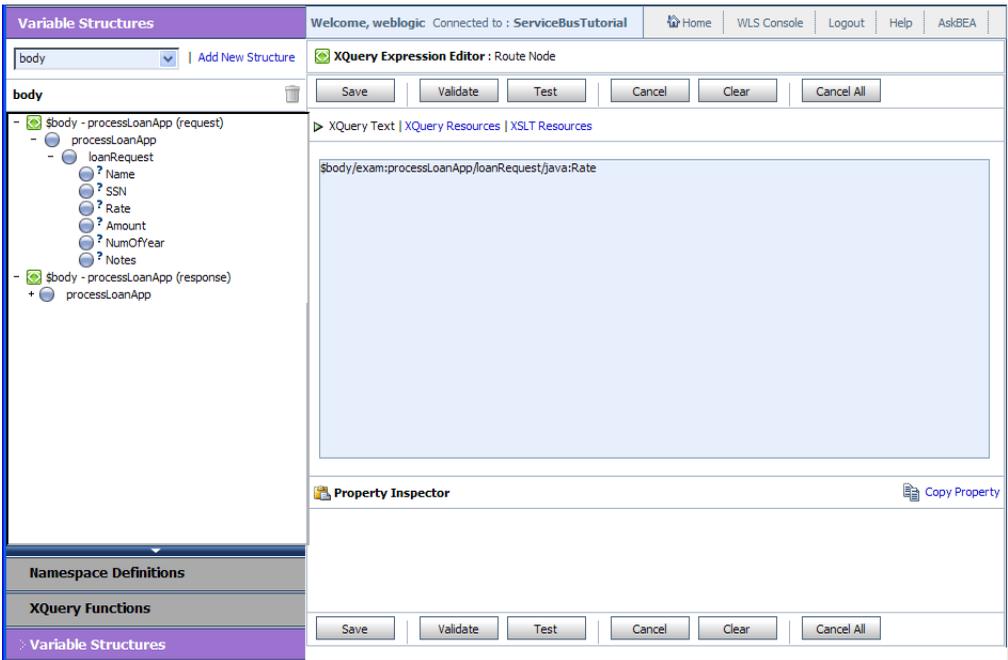


13. Drag-and-drop the circle associated with the rate element in the XQuery Expression text box. The following XQuery expression is written in the text box:

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

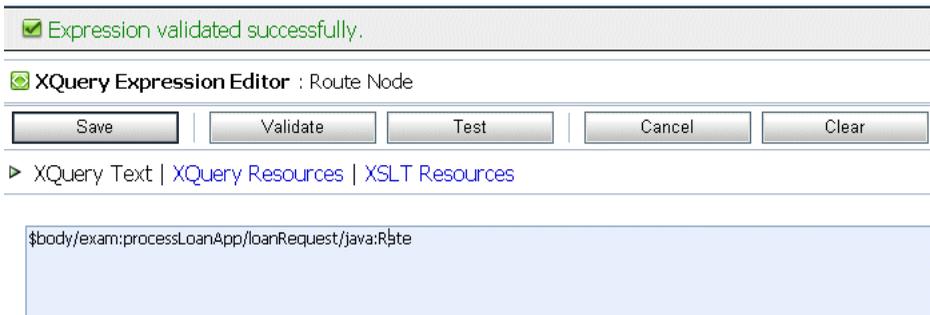
**Note:** The drag-and-drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the Rate element in the Variable Structures pane. The expression is displayed in the Property Inspector palette. Place the cursor in the XQuery expression text box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

Figure 3-33 XQuery Expression Editor



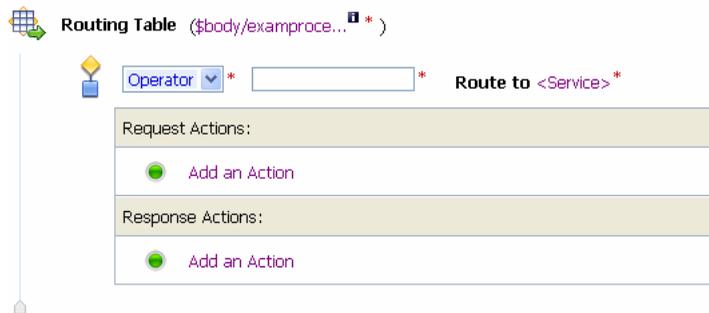
14. Click **Validate** to validate the XQuery. It is a good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button.

Figure 3-34 XQuery Validation



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

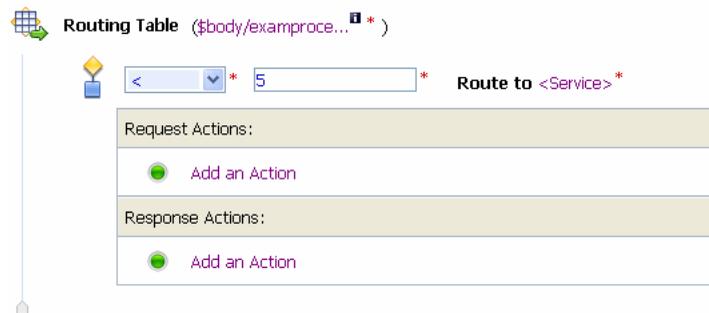
**Figure 3-35 Routing Table View - Expression**



16. In the Operator drop-down list, select `<`. In the associated text field enter the number 5.

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

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



17. Click the **Service** link to define the service to which you want to route messages when the rate is less than five. The Select Service page is displayed.
  - a. Select the ManagerLoanReview business service.
  - b. Click **Submit**.

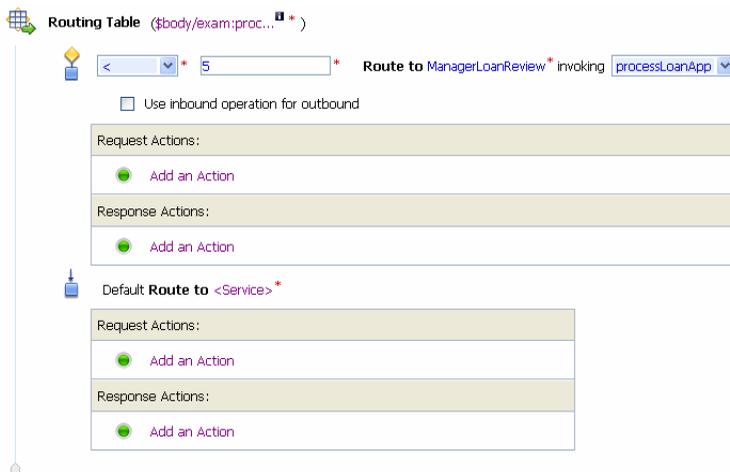
18. In the Routing Table Operation drop-down list, select the processLoanApp operation. This is the operation on the ManagerLoanReview business service that is invoked at run-time if the rate requested in the loan application is less than 5.

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

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

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

Figure 3-37 Routing Table Default Condition



20. In the Default (else) condition, click the **Service** link. The Select Service page is displayed.

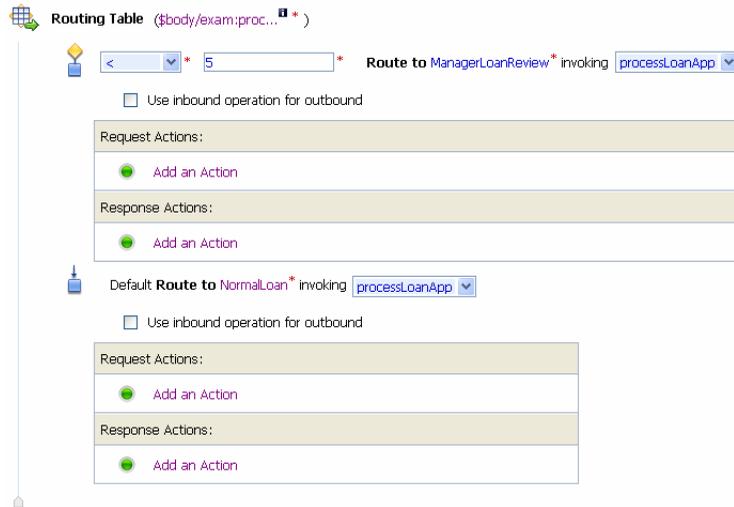
- a. Select the NormalLoan business service.
- b. Click **Submit**.

21. In the Default operation drop-down list, select the processLoanApp operation for the service.

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

After you have completed these tasks, the Edit Stage Configuration page is displayed as shown in [Figure 3-38](#).

**Figure 3-38 Routing Behavior of the Message Flow**

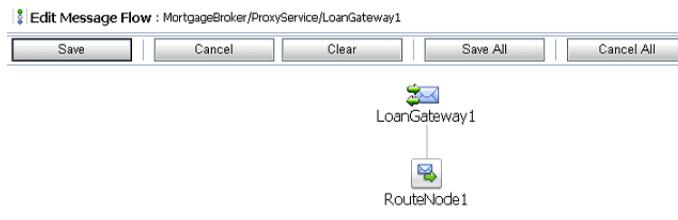


22. Click **Validate** to validate the RouteNode1 routing configuration.

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

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

**Figure 3-39 LoanGateway1 Proxy Service Message Flow Map**



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

25. In the Change Center, click **Activate** the session. For more information, see [Table 3-2](#).

## Summary

[Configure the LoanGateway1 Proxy Service](#) completes the configuration of the run-time message routing behavior in the LoanGateway1 Routing Table. A message is routed to the ManagerLoanReview business service if the rate specified in the loan application is less than 5. Otherwise the message is routed to the NormalLoan business service.

## Test Your Loan Application Routing Configuration Using the Test Console

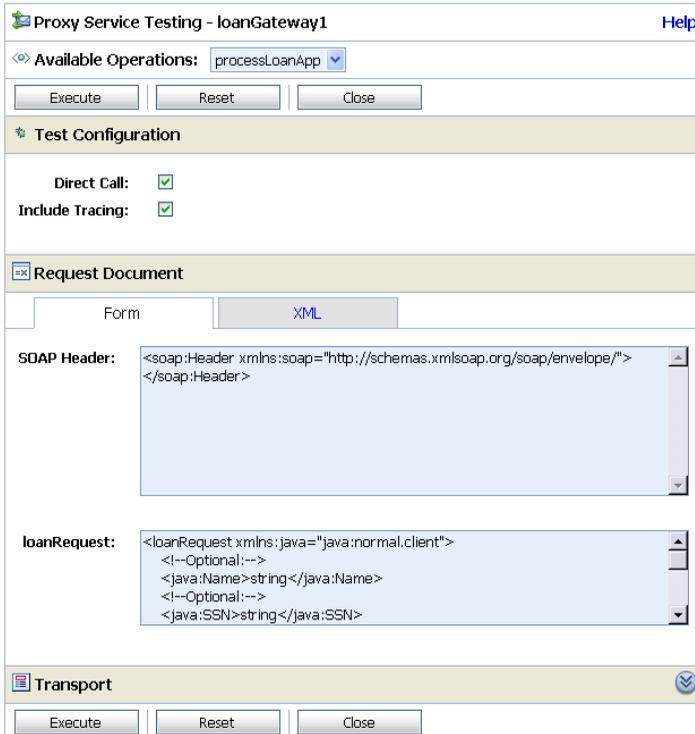
After you have configured ALSB to work with the client and the target business services, you can test the configuration. The routing behavior for the proxy service LoanGateway1 in this scenario is based on the interest rate requested in the loan application message. A change in the value of the interest rate results in a change in the routing behavior as follows:

- A value of 4.9 or less causes the message to be routed to the ManagerLoanReview business service.
- A value of 5.0 or greater causes the message to be routed to the NormalLoan business service.

### To test the Routing of the Loan Application ManagerLoanReviewService

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page.
4. Click on **Launch Test Console** icon in **Actions** column for LoanGateway1 to launch the test console (see [Figure 3-40](#)).

Figure 3-40 Test Console for LoanGateway1



5. Replace the default payload in loanRequest field with the following code:

```
<loanRequest xmlns:java=" java:normal.client ">
  <!--Optional:-->
  <java:Name>Smith</ java:Name>
  <!--Optional:-->
  <java:SSN>1234567</ java:SSN>
  <!--Optional:-->
  <java:Rate>4.1</ java:Rate>
  <!--Optional:-->
  <java:Amount>9000000</ java:Amount>
  <!--Optional:-->
  <java:NumOfYear>10</ java:NumOfYear>
```

## Tutorial 1. Routing a Loan Application

```
<!--Optional-->
<java:Notes>Manager Loan Application Review Service</java:Notes>
</loanRequest>
```

6. Accept other default settings and click **Execute**. The response as in [Figure 3-41](#) is obtained.

**Figure 3-41 Response from the ManagerLoanApplicationReview Service**



The response message indicates that the proxy service LoanGateway1 routes the request to ManagerLoanReview business service because the interest rate requested is 4 (less than 5).

## To test the Routing of the Loan Application Through normalLoanProcessor

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page.
4. Click on **Launch Test Console** icon in **Actions** column for LoanGateway1 to launch the test console (see [Figure 3-40](#))
5. Replace the default payload in loanRequest field with the following code:

```
<loanRequest xmlns:java="java:normal.client">
  <!--Optional-->
```

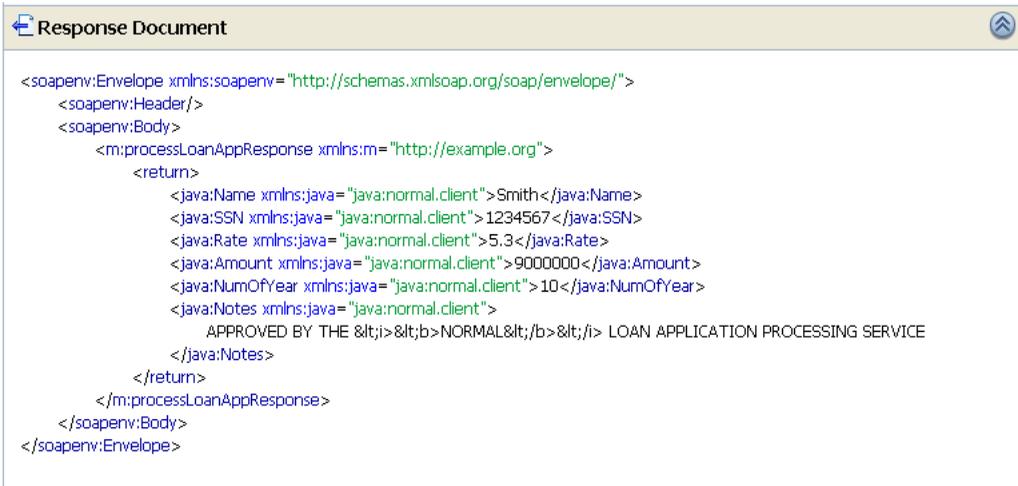
```

    <java:Name>Smith</java:Name>
    <!--Optional:-->
    <java:SSN>1234567</java:SSN>
    <!--Optional:-->
    <java:Rate>5.3</java:Rate>
    <!--Optional:-->
    <java:Amount>9000000</java:Amount>
    <!--Optional:-->
    <java:NumOfYear>10</java:NumOfYear>
    <!--Optional:-->
    <java:Notes>Manager Loan</java:Notes>
  </loanRequest>

```

Accept other default settings and click **Execute**. The response as in [Figure 3-42](#) is obtained.

**Figure 3-42 Response From NormalLoanProcessor**



```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <m:processLoanAppResponse xmlns:m="http://example.org">
      <return>
        <java:Name xmlns:java="java:normal.client">Smith</java:Name>
        <java:SSN xmlns:java="java:normal.client">1234567</java:SSN>
        <java:Rate xmlns:java="java:normal.client">5.3</java:Rate>
        <java:Amount xmlns:java="java:normal.client">9000000</java:Amount>
        <java:NumOfYear xmlns:java="java:normal.client">10</java:NumOfYear>
        <java:Notes xmlns:java="java:normal.client">
          APPROVED BY THE &lt;i>&lt;b>NORMAL&lt;/b>&lt;/i> LOAN APPLICATION PROCESSING SERVICE
        </java:Notes>
      </return>
    </m:processLoanAppResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

The response message indicates that the proxy service LoanGateway1 routes the request to NormalLoanProcessor business service because the interest rate requested is 5.3(greater than 5).

## Where to go From Here

After completing [Tutorial 1. Routing a Loan Application](#) proceed to [Tutorial 2. Transforming a Loan Application](#).

# Tutorial 2. Transforming a Loan Application

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

This section includes the following topics:

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

## Prerequisites

You must complete [Tutorial 1. Routing a Loan Application](#) before beginning this tutorial.

## Tutorial Objectives

This tutorial provides the tasks to create and test a routing and transformation scenario developed using the graphical environment provided in ALSB Console. Using the ALSB Console you will build on what you learned in [Tutorial 1. Routing a Loan Application](#) to perform the following tasks:

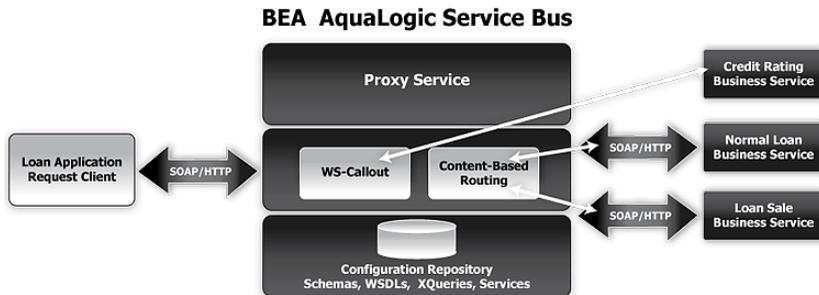
- Configure content-based routing.
- Use an XPath expression to change the content in the message body.

## Definition of the Scenario

A primary mortgage company uses ALSB to identify and re-route loan applications that can be sold to secondary loan companies. Loan applications with a principal request of greater than US \$25 million are candidates for sale to a secondary loan company. When ALSB receives a loan application meeting these criteria, the applicant’s credit rating information is retrieved (by making a callout to a web service). The credit rating information is added to the loan application. The application is then forwarded to the secondary mortgage company web service to be processed. Loan applications with a principal request equal to or less than US \$25 million are routed to a different business service for processing. The target business services respond indicating whether the loan application is approved or rejected.

Figure 4-1 illustrates where ALSB 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 ALSB



## Overview of the Run-time Process in ALSB

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

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

## Required Resources

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

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

Resource Name	Description
creditRating, loanSale	These are the WSDL resources.
NormalLoan, LoanSaleProcessor, CreditRating	These are the external business services used by ALSB.
LoanGateway2	This is the ALSB proxy service.

## Tasks in This Tutorial

In this tutorial, you will perform the following tasks:

- Import additional WSDL resources
- Register new business services and proxy services
- Configure the routing behavior for the proxy service, by executing the following tasks:
  - Adding a Service Callout to get a credit rating
  - Changing the namespace and inserting a new `<CreditRating>` element in the outbound (request) message
  - Reverting the namespace change and removing the `<CreditRating>` element on the response message.

## Prepare the Environment

Ensure that ALSB is running in the domain that you created for the tutorial and that you have completed the tasks described in [Tutorial 1. Routing a Loan Application](#).

## Create a Session

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

1. In the ALSB console Change Center click **Create** to create a new session.
2. In the ALSB Console navigation pane, select the Project Explorer.  
The project explorer pane is opened in the navigation pane and a project page is displayed in the console.
3. In the project explorer, expand the MortgageBroker project tree to expose the subfolders containing the project artifacts:
  - ProxyService
  - BusinessService
  - WSDL

## Create the Resources

In this section , you will learn how to import the WSDL resource, and create the proxy service and the business services required for the tutorial.

### Create the WSDL Resources

You must create the WSDL resources (loanSale and creditRating) before creating the other resources required for this scenario. WSDL resources are the building blocks for creating the business and proxy services. To import the appropriate WSDLs and create the WSDL resources, follow the tasks described in [To Create the normalLoan WSDL Resource](#) section in [Tutorial 1](#).

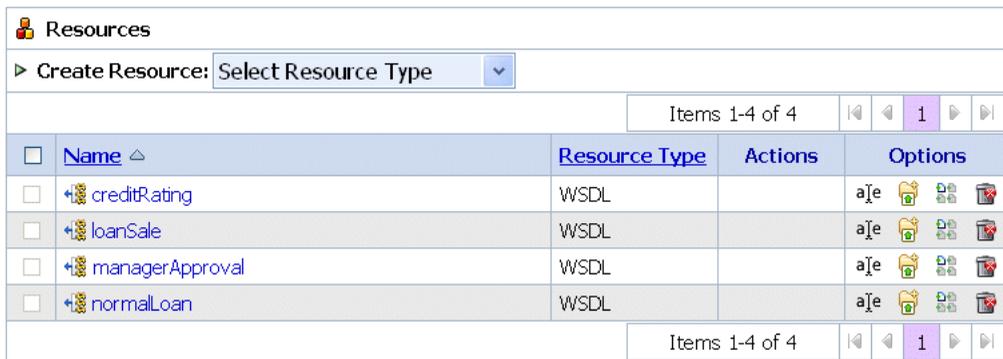
[Routing a Loan Application](#). However, for this instance, name your resources and base them on the WSDLs listed in the following table.

**Table 4-2 WSDL Configuration Settings**

Resource Name	Location of the WSDL
loanSale	BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webservices\jws_basic\large\LargeLoanPurchasingService.wsdl
creditRating	BEA_HOME\alsb_3.0\samples\servicebus\examples\src\examples\webservices\jws_basic\credit\CreditLoanApprovalService.wsdl

When you complete this task, 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 [Figure 4-2](#).

**Figure 4-2 WSDL Resources**



## Create a Proxy Service

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

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

To create a new proxy service, follow the tasks described in [To Create the Proxy Service](#). For this instance, use the proxy service name and the endpoint URI listed in the following table.

**Table 4-3 LoanGateway2 Proxy Service Configuration Settings**

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

After you complete this task, a summary of the configuration settings for the proxy service is displayed as shown in [Figure 4-3](#). You must review the configuration settings before to registering the proxy service.

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

Create a Proxy Service - Summary (MortgageBroker/ProxyService/LoanGateway2)	
<b>General Configuration</b>	
Service Name	LoanGateway2
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/normalLoan, port="helloPort")
Content Streaming	Disabled
<b>Transport Configuration</b>	
Protocol	http
Endpoint URI	loan/gateway2
Get All Headers	No
Headers	
<b>HTTP Transport Configuration</b>	
HTTPS required	No
Authentication	None
<b>Operation Selection Configuration</b>	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type
<input style="margin-right: 10px;" type="button" value=" &lt;&lt; Prev. "/> <input style="margin-right: 10px;" type="button" value=" Save "/> <input style="margin-right: 10px;" type="button" value=" Cancel "/>	

When you complete this task, the MortgageBroker/ProxyService folder contains the proxy services that you created in this tutorial and in [Tutorial 1. Routing a Loan Application](#) as shown in [Figure 4-4](#).

**Figure 4-4 Proxy Service Resources**

Resources				
Create Resource: <input type="text" value="Select Resource Type"/>				
Items 1-3 of 3				
<input style="margin-right: 5px;" type="button" value=" &lt; "/> <input style="margin-right: 5px;" type="button" value=" &lt;&lt; "/> <input style="background-color: #e0e0e0; margin-right: 5px;" type="button" value=" 1 "/> <input style="margin-right: 5px;" type="button" value=" &gt;&gt; "/> <input style="margin-right: 5px;" type="button" value=" &gt; "/>				
<input type="checkbox"/>	Name <sup>△</sup>	Resource Type	Actions	Options
<input type="checkbox"/>	LoanGateway1	Proxy Service		a e
<input type="checkbox"/>	LoanGateway2	Proxy Service		a e

## Create a Business Service

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

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

You created the NormalLoan service in [Tutorial 1. Routing a Loan Application](#). To create the LoanSaleProcessor and CreditRatingService business services for this scenario, follow the tasks described in [Create the NormalLoan Business Service](#). You must configure your services using the names, service types, and endpoint URIs listed in the following two tables.

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

Business Service Name	LoanSaleProcessor
Service Type	Select the WSDL web service as the Service Type option. Select the LoanSale WSDL in the <b>Select a WSDL</b> page. Select the helloPort port in the <b>Select a WSDL Definition</b> page.
Endpoint URI	Ensure that the pre-populated value for the Endpoint URI in the Existing URIs field is <code>http://&lt;host:port&gt;/ljws_basic_ejb/LargeSimpleBean</code> where <host:port> represents the machine and port on which the ALSB server is running.
Load Balancing Algorithm	none

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

Business Service Name	CreditRating
Service Type	Select the WSDL port option. Select the creditRating WSDL in the <b>Select a WSDL</b> page. Select the helloPort port in the <b>Select a WSDL Definition</b> page.
Endpoint URI	Ensure that the pre-populated value for the endpoint URI in the Existing URIs field is <code>http://&lt;host:port&gt;/crejws_basic_ejb/CreditSimpleBean</code> where <host:port> represents the machine and port on which your ALSB server is running.
Load Balancing Algorithm	none

When you complete this task, the MortgageBroker/BusinessService folder contains the business services that you created in this tutorial and in [Tutorial 1. Routing a Loan Application](#) as shown in [Figure 4-5](#).

**Figure 4-5 Business Service Resources**

<input type="checkbox"/>	Name	Resource Type	Actions	Options
<input type="checkbox"/>	CreditRatingService	Business Service		a e
<input type="checkbox"/>	loanSaleProcessor	Business Service		a e
<input type="checkbox"/>	managerLoanReviewService	Business Service		a e
<input type="checkbox"/>	normalLoanProcessor	Business Service		a e

Items 1-4 of 4

## Summary

Upon completing all the tasks from [Prepare the Environment](#) to [Create the Resources](#) you have created the resources required for this scenario. You have also configured the proxy service with a base configuration. In the following three tasks you will configure the proxy service by adding the routing, transformation, and Service Callout behavior for the loan application messages.

## Configure Routing for LoanGateway2 Proxy Service

A proxy service is implemented in ALSB as a message flow, which includes request and response pipelines. This task includes the following tasks:

- [Create a Routing Table](#)
- [Configure the Routing Expression](#)

### Create a Routing Table

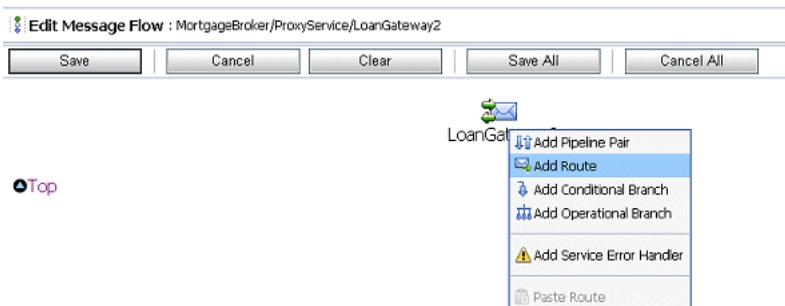
1. In the ALSB Console navigation pane, select **Resource Browser**.

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

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

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

Figure 4-6 Edit Message Flow -LoanGateway2



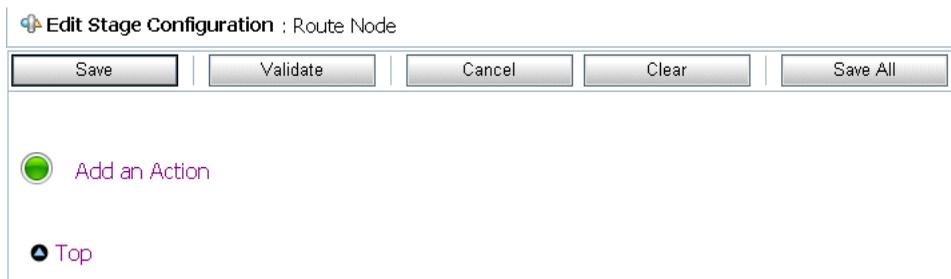
3. Click LoanGateway2, then select **Add Route** from the short-cut menu. In the configuration dialog, RouteNode1 is added.

**Figure 4-7 Edit Message Flow - LoanGateway2 RouteNode1**



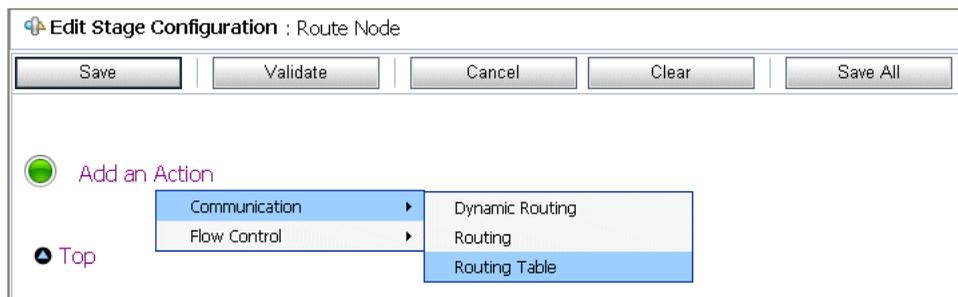
4. Click **RouteNode1** and select **Edit Route** from the short-cut menu. The **Edit Stage Configuration** page is displayed and contains a single link, **Add an Action**.

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



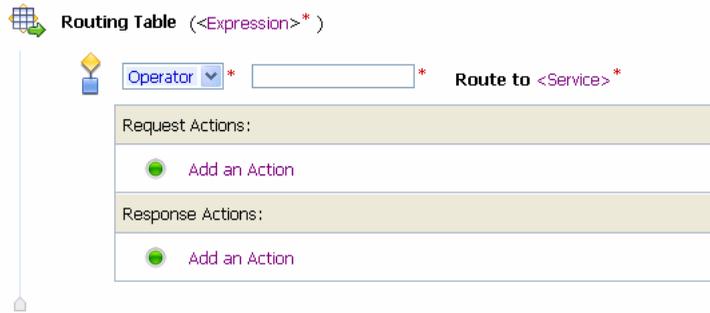
5. Click **Add an Action**, then select **Communication > Routing Table** from the short-cut menu.

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



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

Figure 4-10 Routing Table Configuration Page

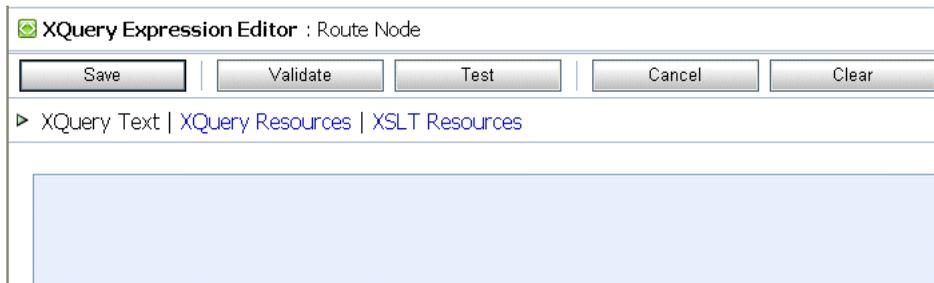


To configure the routing table to route messages to business services based on the amount element of the incoming message, you must create an XQuery expression using the XQuery Expression Editor.

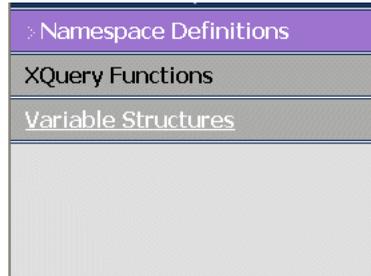
## Configure the Routing Expression

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

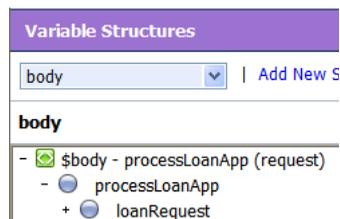
Figure 4-11 XQuery Expression Editor



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

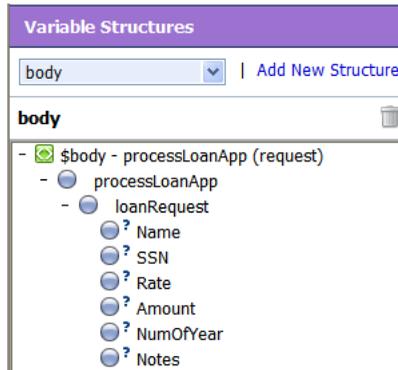
**Figure 4-12 Namespace Definitions Pane**

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

**Figure 4-13 Variable Structures Pane**

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

Figure 4-14 loanRequest Element



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

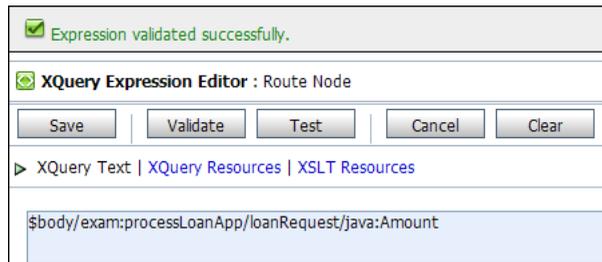
```
$body/exam:processLoanApp/loanRequest/java:Amount
```

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

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

It is good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button. In this case, the expression is valid as shown in [Figure 4-15](#).

**Figure 4-15 Validate Routing Expression**



9. Click **Save**.

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

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

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

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

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

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

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

You have now defined the case for routing the loan application to the LoanSaleProcessor business service as shown in [Figure 4-16](#).

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



When a loan application with a loan amount greater than US \$25 million is identified, a web service callout (Service Callout) is performed to retrieve the customer's credit rating. The credit rating information is added to the loan application. The application is then forwarded to the secondary mortgage company's business service to be processed.

## Configure Routing Request Actions for LoanGateway2 Proxy Service

This section describes how to configure the proxy service to do the Service Callout and transform the message appropriately for the target service.

### Configure the Service Callout Input Parameter

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

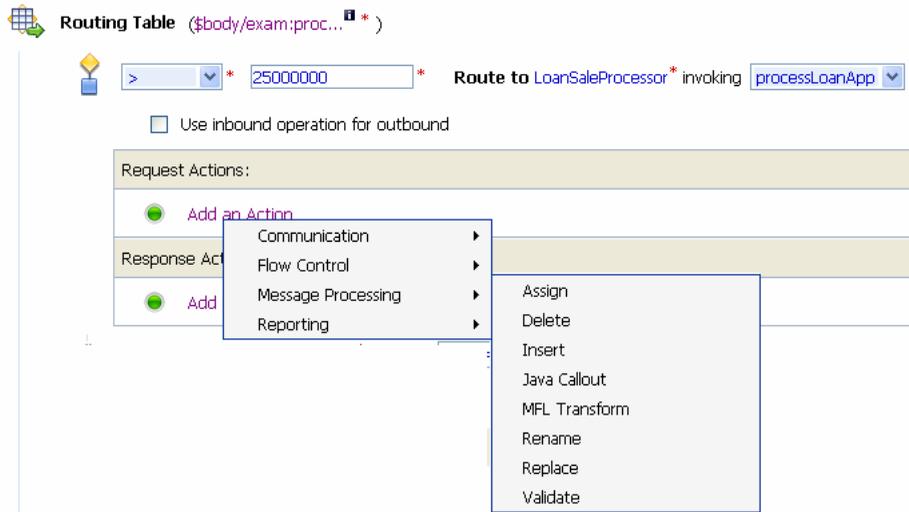
- Delete the “xsi:type” attribute from the message
- Assign an input parameter for the Service Callout action
- Rename the namespace for the Service Callout input parameter

### Add a Delete Action

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

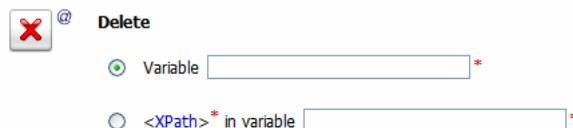
1. In the Routing Table Request Actions pane, select **Add an Action > Message Processing > Delete**.

Figure 4-17 Add a Delete Action



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

Figure 4-18 Initial Delete Action



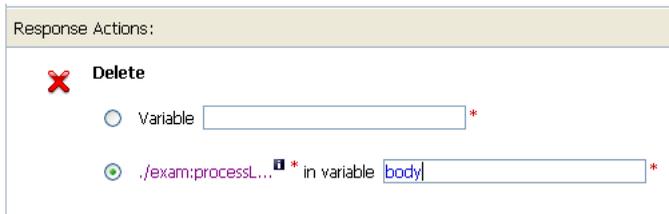
2. Select the **<XPath>** in variable option, then click **<XPath>**. The **XPath Expression** editor is displayed.
  - a. In the XPath Expression text box, enter the following expression:
 

```
./exam:processLoanApp/loanRequest/@xsi:type
```
  - b. Click **Validate**, then Save. You will return to the **Edit Stage Configuration** page.
3. Enter body in the variable text field of the **<XPath>** in variable option in the Delete action.

The configuration for the Delete action is therefore:

Delete `./exam:processLoanApp/loanRequest/@xsi:type` in variable body as shown in [Figure 4-19](#).

Figure 4-19 LoanGateway2 Routing Request Delete Action



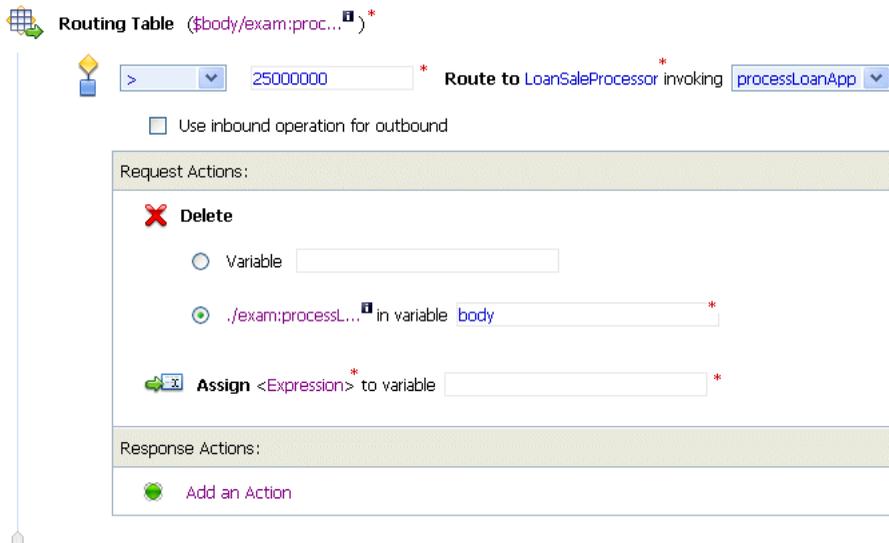
### Assign a Service Callout Input Parameter

You must now assign an input parameter for the service callout action as follows:

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

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

Figure 4-20 Initial Assign Action

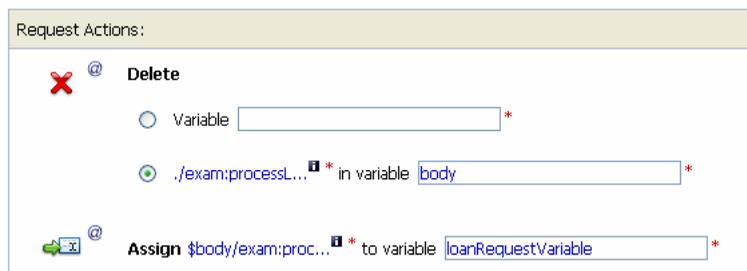


3. Click the **<Expression>** link. The **XQuery Expression** editor is displayed.
4. Accept the default selection of **XQuery Text** in the editor.

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

```
$body/exam:processLoanApp/loanRequest
```
7. Click **Validate** to validate the XQuery, then **Save**. The routing table is displayed on the **Edit Stage Configuration** page. **<Expression>** is now replaced by the expression that returns the content of the message to be sent to the **Service Callout** service.
8. Enter `loanRequestVariable` in the variable text box as shown in [Figure 4-21](#).

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



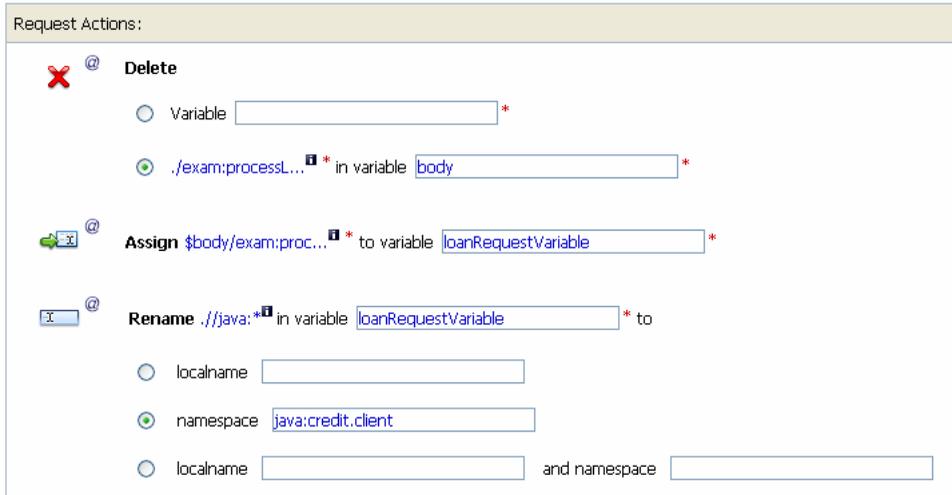
This task completes the assignment of value returned by the XQuery expression (`$body/exam:processLoanApp/loanRequest`) to the `loanRequestVariable` variable.

## Rename Input Variable Namespace

You must now rename the namespace of the input parameter you have assigned for the Service Callout using the tasks below:

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

Figure 4-22 Initial Input Variable Rename Action



3. In the Rename statement, click the <XPath> link. The **XPath Expression Editor** is displayed.
4. In the XPath Expression text box, enter the following XPath expression:  

```
./java:*
```

This expression identifies every instance of a namespace with a prefix of java.
5. Click **Save** to save the XPath expression and return to the **Edit Stage Configuration** page.
6. In the variable text box, enter `loanRequestVariable`.  

You have completed creating a condition in which the XPath expression finds all the namespaces with the java prefix in the `loanRequestVariable` context variable. The next task specifies the namespace to replace the namespaces identified by the XPath expression.
7. Select the namespace option and enter the fully qualified new namespace to be substituted in the message. The new namespace is `java:credit.client`.  
**Note:** The default namespace that you are replacing in this case is `java:normal.client`.  
The Rename action is displayed as shown in [Figure 4-23](#).

**Figure 4-23 LoanGateway2 Routing Request Rename Action**

 **Delete**

Variable  \*

./exam:processL...<sup>1</sup> \* in variable  \*

 **Assign** \$body/exam:proc...<sup>1</sup> \* to variable  \*

 **Rename** ./java:\*<sup>1</sup> in variable  \* to

localname

namespace

localname  and namespace

## Add a Service Callout Action

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

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

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

Figure 4-24 Initial Service Callout Action

The screenshot shows a configuration interface for a service callout action. It consists of several sections:

- Delete:** A red 'X' icon with a '@' symbol. Below it are two radio buttons: 'Variable' (unselected) and './exam:process...' (selected) in variable 'body'.
- Assign:** A green arrow icon with a '@' symbol. Below it is the text 'Assign \$body/exam:proc...' to variable 'loanRequestVariable'.
- Rename:** A blue 'i' icon with a '@' symbol. Below it is the text 'Rename ./java:\*' in variable 'loanRequestVariable' to:
  - localname (empty field)
  - namespace 'java:credit.client' (selected)
  - localname (empty field) and namespace (empty field)
- Service Callout to <Service>:** A green globe icon with a '@' symbol. Below it is the text 'Service Callout to <Service>'.

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

The service callout action will be displayed as shown in [Figure 4-25](#).

Figure 4-25 Service Callout Action with Configured Service

The screenshot shows the configuration page for a service callout action. At the top, it says 'Service Callout to creditRating\*' invoking 'Operation' (selected in a dropdown menu). Below this is a table with two sections:

Request Actions:
<input checked="" type="radio"/> Add an Action
Response Actions:
<input checked="" type="radio"/> Add an Action

4. From the **Operation** drop-down list, select **processLoanApp**. The fields that allow you to configure the request and response parameters for the service callout are displayed as shown in [Figure 4-26](#).

Figure 4-26 Service Callout Action - Configured Operation

Service Callout to **creditRating\*** invoking **processLoanApp** \*

Configure Soap Body
  Configure Payload Parameters

Request Parameters:

loanRequest  \*

Response Parameters:

return  \*

SOAP Request Header:

SOAP Response Header:

Request Actions:

Add an Action

Response Actions:

Add an Action

- In the service callout **Request Parameters** section, set the **loanRequest** parameter to `loanRequestVariable` (the message context variable you defined in the preceding section).
- In the service callout **Response Parameters** section, set the **return** parameter name to `creditRating` as shown in [Figure 4-27](#).

Figure 4-27 Service Callout Action - Configured Parameters

Service Callout to **CreditRatingService\*** invoking **processLoanApp** \*

Configure Soap Body
  Configure Payload Parameters

Request Parameters:

loanRequest  \*

Response Parameters:

return  \*

This task completes the service callout configuration to invoke the `creditRating` web service. The credit rating returned by this web service is assigned to the `$creditRating` context variable.

## Configure Message Transformation

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

- Rename the namespace for the message to the namespace required by the target service using XPath expressions as described in [Rename the Message Namespace](#).
- Insert a new element into the message by adding an element action in the configuration as described in [Insert a New Element in the Message](#).

### Rename the Message Namespace

1. Click the **Service Callout** icon and select Add an Action > Message Processing > Rename. The Rename action configuration fields are added to the Request Actions pane.

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

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

```
./java:*
```

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

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

5. In the variable text box, enter `body`. The body context variable contains the body of the message.

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

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

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

The Rename action is displayed as shown in [Figure 4-28](#).

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

 **Rename** `./java:*` in variable  \* to

localname

namespace

localname  and namespace

**Insert a New Element in the Message**

1. Click the **Rename** icon and select **Add an Action > Message Processing > Insert**.

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

2. In the Insert action, click **<Expression>**. The XQuery Expression editor is displayed. You will add a new namespace on this page in the subsequent tasks.
3. In the Namespace Definitions palette, navigate to the User Defined Namespaces area.
  - a. Click **Add Namespace**.
  - b. In the Prefix field, enter `lg`.
  - c. In the URI field, enter `java:large.client`
  - d. Click **Add**.

The new namespace is displayed in the User Defined Namespaces area in the Namespace Definitions palette as shown in [Figure 4-29](#).

**Figure 4-29 User Defined Namespace**

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

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

where:

## Tutorial 2. Transforming a Loan Application

- lg: is the namespace of the element that you add to the message that should be associated with the java:large.client namespace.
- {} indicates to the XQuery engine that the content between the {} is not XML and must be interpreted.

At run time, the \$creditRating variable is assigned a credit rating value by the CreditRating business service. (This is the business service to which we configured a Service Callout action in section [Add a Service Callout Action](#)).

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

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

6. In the Insert action, select `after` from the drop-down list.
7. Click the **<XPath>** link. The 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

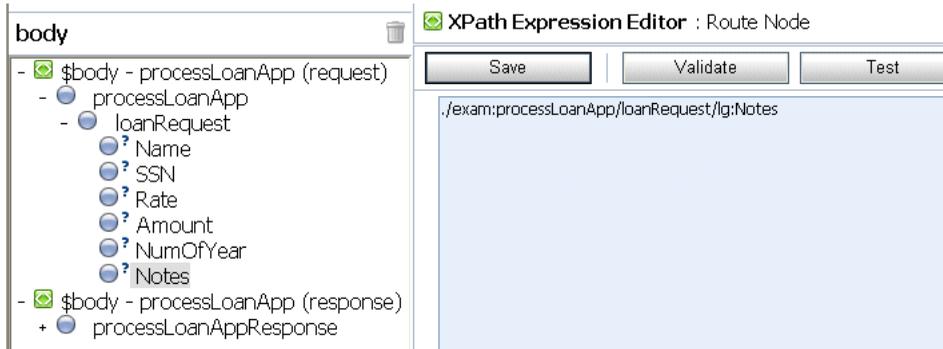
```
./exam:processLoanApp/loanRequest/java:Notes
```

**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 in the XPath Expression text box.

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

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

Figure 4-30 Configure Outbound Message



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 will be displayed as shown in [Figure 4-31](#).

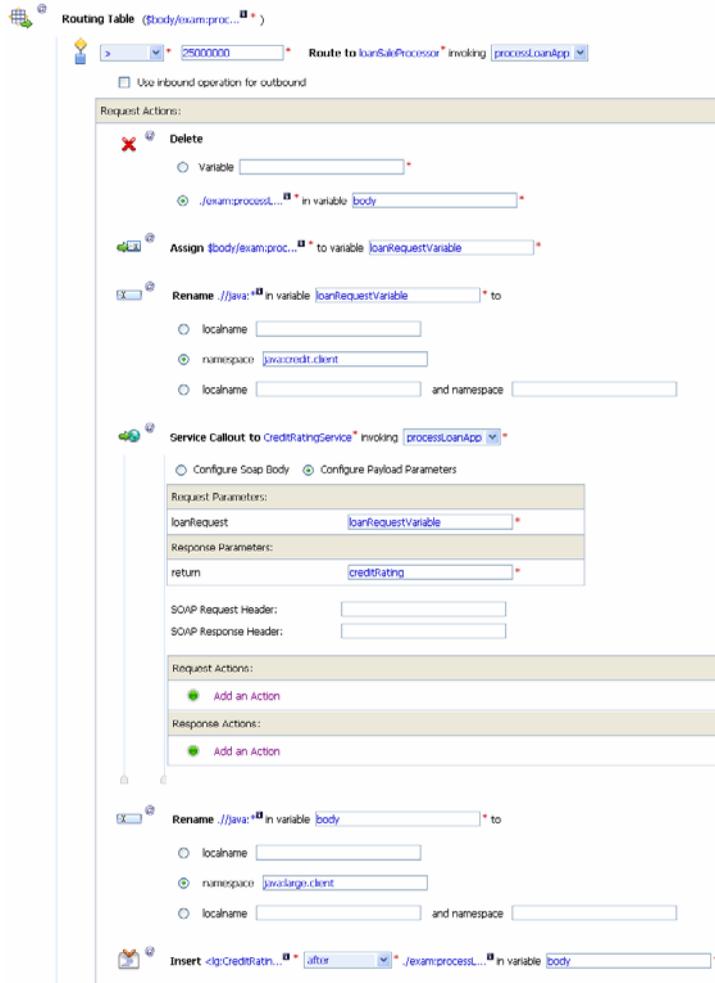
Figure 4-31 Insert New Element into Message



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

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

Figure 4-32 Configuration of Outbound Message



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

## Configure Routing Response Actions for LoanGateway2 Proxy Service

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

- Remove the <CreditRating> element from the response message as described in [Delete an Element](#).
- Revert the namespace to the namespace in the original message: java:normal.client as described in [Add a Rename Action](#).

### Delete an Element

1. Ensure that the **Edit Stage Configuration** page for the routing table that you created and configured in the preceding tasks is open in the ALSB Console.
2. In the routing table **Response Actions** pane, select **Add an Action > Message Processing > Delete**.

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

3. Select the <**XPath**> in variable option, then click <**XPath**>. The **XPath Expression Editor** page is displayed.
  - a. In the XPath Expression text box, enter the following expression:
 

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

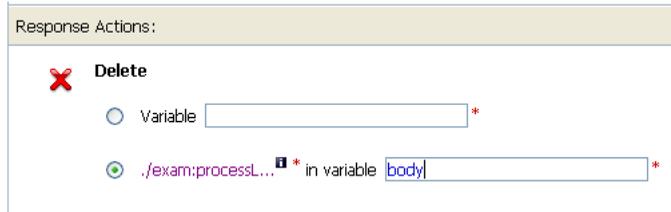
where <processLoanAppResponse> is the WSDL operation that was invoked with the string Response appended to it, and return is the WSDL part name.
  - b. Click **Validate**, then **Save**. **Edit Stage Configuration** page is displayed.

4. Enter `body` in the variable text field of the <**XPath**> in variable option in the Delete action.

The configuration for the Delete action is therefore:

Delete `./exam:processLoanAppResponse/return/lg:CreditRating` in variable body as shown in [Figure 4-33](#).

**Figure 4-33 Response Delete Action**



You have configured ALSB to remove the `<CreditRating>` element from the response message, when the message is processed in the response pipeline.

### Add a Rename Action

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

**Note:** Recall that you configured the request message to the `LoanSaleProcessor` business service to change the namespace to that required by that service. (See [Rename the Message Namespace](#)).

1. Click the **Delete** icon, then **Add an Action > Message Processing > Rename**.  
The Rename action configuration fields are added to the Response Actions pipeline.
2. In the Rename statement, click **<XPath>**. The XPath Expression Editor is displayed.
  - a. In the XPath Expression text box, enter the following expression:  

```
./lg:*
```

This XPath expression finds every namespace with a prefix of `lg:`
  - b. Click **Save**. The **Edit Stage Configuration** page is displayed.
3. In the Rename action's variable text field enter `body`.
4. Select the namespace option, and in the text field associated with the namespace, enter `java:normal.client`

When these tasks are completed, you have specified that ALSB replaces all namespaces with the prefix `lg:` in the response message with `java:normal.client` as shown in [Figure 4-34](#).

**Figure 4-34 Add Rename Action**

**Rename** `./lg:*` in variable  \*

localname

namespace

localname  and namespace

After these tasks are complete, the response action configuration of the proxy service will be displayed as shown in [Figure 4-35](#).

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

Response Actions:

**Delete**

Variable  \*

`./exam:processL...` in variable  \*

**Rename** `./lg:*` in variable  \*

localname

namespace

localname  and namespace

The last task 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.

## Add a Default Routing Case

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

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

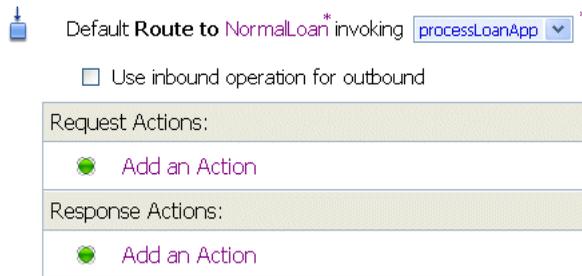
The default case is added after the routing table on the **Edit Stage Configuration** page.

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

## Tutorial 2. Transforming a Loan Application

- a. Select the NormalLoan business service.
- b. Click **Submit**.
3. In the default case configuration operation drop-down list, select the operation processLoanApp as shown in [Figure 4-36](#).

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



4. Click **Save** to save the route node configuration and return to the **Edit Message Flow** page for the LoanGateway2 proxy service.
  5. Click **Save** on each configuration page until you return to the **Edit Message Flow** page.
  6. Click **Activate**, and then **Submit** to save the session configuration information.
- The message flow map of the proxy service will be displayed as shown in [Figure 4-37](#).

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



[Figure 4-38](#) shows the complete route node for LoanGateway2.

Figure 4-38 Configuration of the Route Node for LoanGateway2

The screenshot displays the configuration for a route node in a service bus. The main configuration area is titled "Routing Table (tbody:exam.proc...)" and contains the following elements:

- Route Definition:** A dropdown menu is set to "Route to loanSaleProcessor" with an "invoking process:loanApp" dropdown.
- Request Actions:**
  - Delete:** A radio button is selected for "Variable" with the value ".exam.process..." in variable "body".
  - Assign:** A radio button is selected for "Assign tbody:exam.proc... to variable loanRequestVariable".
  - Rename:** A radio button is selected for "Rename //jws: in variable loanRequestVariable to" with "namespace" set to "jws:credit\_client".
  - Service Callout to CreditRatingService:** A radio button is selected for "Configure Payload Parameters".
    - Request Parameters:** "loanRequest" is mapped to "loanRequestVariable".
    - Response Parameters:** "return" is mapped to "creditRating".
    - SOAP Request Header:** and **SOAP Response Header:** fields are present but empty.
- Response Actions:**
  - Rename:** A radio button is selected for "Rename //lg: in variable body to" with "namespace" set to "jws:large\_client".
  - Delete:** A radio button is selected for "Variable" with the value ".exam.process..." in variable "body".
- Default Route:** A dropdown menu is set to "Default Route to normalLoanProcessor" with an "invoking process:loanApp" dropdown. It includes "Request Actions" and "Response Actions" sections, each with an "Add an Action" button.

At the bottom left of the interface, there is a "Top" button.

## Test the Loan Application Routing Configuration Using the Test Console

This section describes how to test the loan application using the test console. To test the loan, you must first deploy the client jars in the WLS console. For more information see, [Load the Client Application](#).

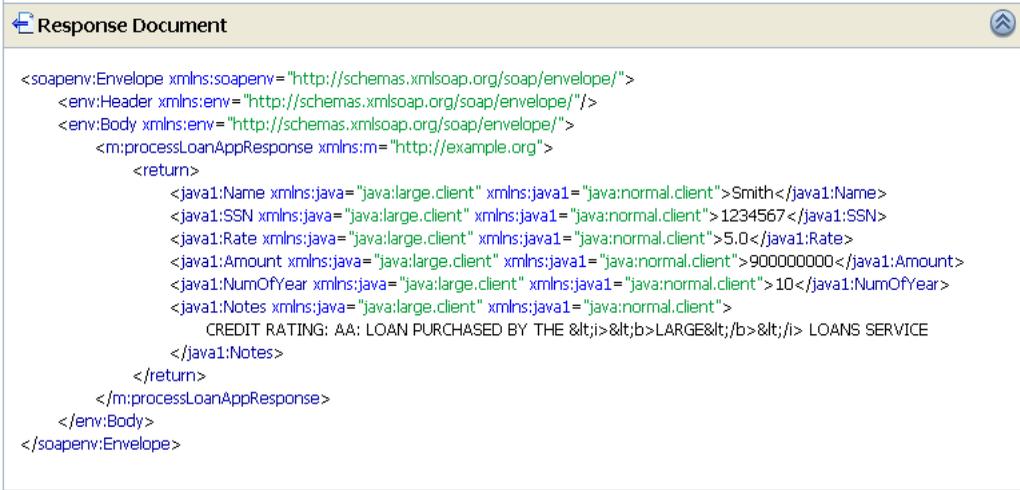
### To test the Routing of the Loan Application Using LoanSaleProcessor Service

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page. Click on **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see [Figure 3-40](#)).
4. Replace the default payload in loanRequest field with the following code:

```
<loanRequest xmlns:java="java:normal.client">
  <!--Optional:-->
  <java:Name>Smith</java:Name>
  <!--Optional:-->
  <java:SSN>1234567</java:SSN>
  <!--Optional:-->
  <java:Rate>5</java:Rate>
  <!--Optional:-->
  <java:Amount>900000000</java:Amount>
  <!--Optional:-->
  <java:NumOfYear>10</java:NumOfYear>
  <!--Optional:-->
  <java:Notes>Large Loan Processing</java:Notes>
</loanRequest>
```

Accept other default settings and click **Execute**. The response as in [Figure 4-39](#).

Figure 4-39 Response From LoanSaleProcessor Business Service



```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" />
  <env:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <m:processLoanAppResponse xmlns:m="http://example.org">
      <return>
        <java1:Name xmlns:java="java:large.client" xmlns:java1="java:normal.client">Smith</java1:Name>
        <java1:SSN xmlns:java="java:large.client" xmlns:java1="java:normal.client">1234567</java1:SSN>
        <java1:Rate xmlns:java="java:large.client" xmlns:java1="java:normal.client">5.0</java1:Rate>
        <java1:Amount xmlns:java="java:large.client" xmlns:java1="java:normal.client">900000000</java1:Amount>
        <java1:NumOfYear xmlns:java="java:large.client" xmlns:java1="java:normal.client">10</java1:NumOfYear>
        <java1:Notes xmlns:java="java:large.client" xmlns:java1="java:normal.client">
          CREDIT RATING: AA; LOAN PURCHASED BY THE &lt;i>&lt;b>LARGE&lt;/b>&lt;/i> LOANS SERVICE
        </java1:Notes>
      </return>
    </m:processLoanAppResponse>
  </env:Body>
</soapenv:Envelope>

```

The response message indicates that the proxy service LoanGateway2 routes the application to LoanSale Processor processed the request because the amount of the loan requested is greater than US \$25 million.

## Test the NormalLoan Business Service

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page. Click on **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see [Figure 3-40](#)).
4. Replace the default payload in loanRequest field with the following code:

```

<loanRequest xmlns:java="java:normal.client">
  <!--Optional:-->
  <java:Name>Smith</java:Name>
  <!--Optional:-->
  <java:SSN>1234567</java:SSN>
  <!--Optional:-->

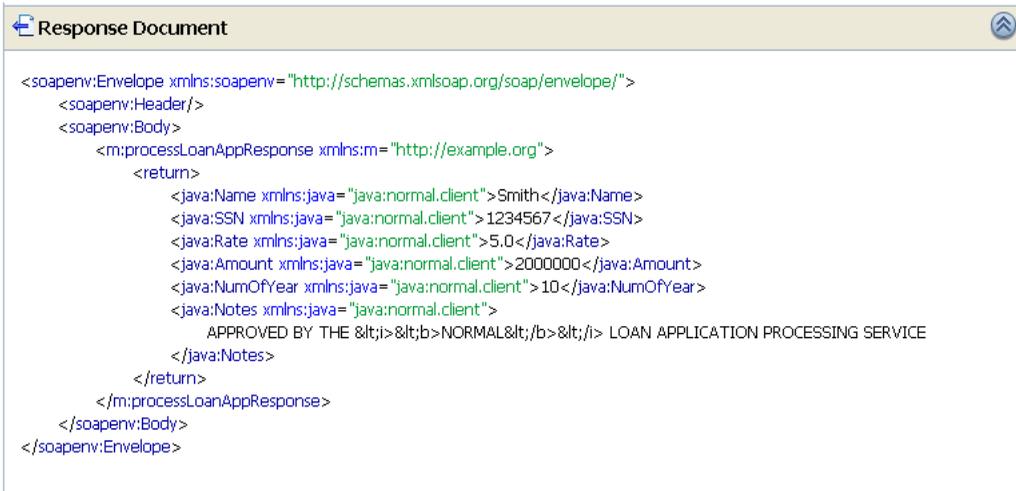
```

## Tutorial 2. Transforming a Loan Application

```
<java:Rate>5</java:Rate>
<!--Optional:-->
<java:Amount>2000000</java:Amount>
<!--Optional:-->
<java:NumOfYear>10</java:NumOfYear>
<!--Optional:-->
<java:Notes>Normal Loan</java:Notes>
</loanRequest>
```

Accept other default settings and click **Execute**. The response as in [Figure 4-40](#).

**Figure 4-40 Response from NormalLoanProcessor**



```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <m:processLoanAppResponse xmlns:m="http://example.org">
      <return>
        <java:Name xmlns:java="java:normal.client">Smith</java:Name>
        <java:SSN xmlns:java="java:normal.client">1234567</java:SSN>
        <java:Rate xmlns:java="java:normal.client">5.0</java:Rate>
        <java:Amount xmlns:java="java:normal.client">2000000</java:Amount>
        <java:NumOfYear xmlns:java="java:normal.client">10</java:NumOfYear>
        <java:Notes xmlns:java="java:normal.client">
          APPROVED BY THE &lt;i>&lt;b>NORMAL&lt;/b>&lt;/i> LOAN APPLICATION PROCESSING SERVICE
        </java:Notes>
      </return>
    </m:processLoanAppResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

The response message indicates that the proxy service `LoanGateway2` routes the application to `NormalLoanApproval` business service processed the request because the amount of the loan requested was less than US \$25 million.

## Best Practices

In this tutorial, you configured the transformation and replaced the namespaces in the `Route` node. Alternatively, it is possible to configure the same logic in the request pipeline in a pipeline pair.

The work you must do to accomplish the transformation and namespace replacement is the same regardless of the model that you choose.

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

The following scenario describes a situation in which the choice of the implementation is important:

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

## Tutorial 2. Transforming a Loan Application

# Tutorial 3. Validating a Loan Application

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

You can configure message validation logic in the following ways:

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

This tutorial includes the following topics:

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

## Prerequisites

You must complete [Tutorial 1. Routing a Loan Application](#) and [Tutorial 2. Transforming a Loan Application](#) before beginning this tutorial:

## Tutorial Objectives

The objective of this tutorial is to provide tasks to configure ALSB resources that implement message validation and error handling, in ALSB Console. This tutorial includes the following:

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

## Features Highlighted

This tutorial exposes you to the following features of ALSB:

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

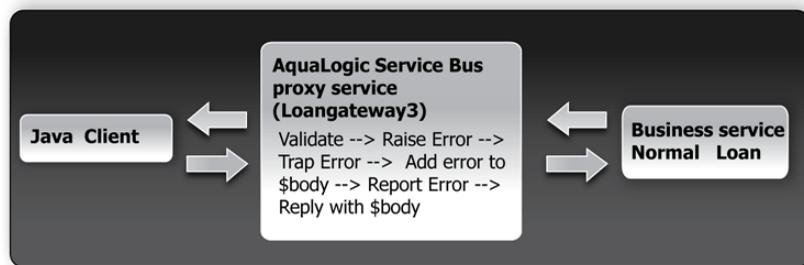
## Definition of the Scenario

A primary mortgage company routes every loan application through an ALSB proxy service that validates the message. If the application is:

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

[Figure 5-1](#) summarizes the logical architecture to support this scenario and illustrates how ALSB is positioned in the enterprise system environment to mediate the messaging between the client and the business services.

Figure 5-1 Validating a Loan Application using ALSB



## Overview of the Run-Time Process in ALSB

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

In this scenario, the message routing is as follows:

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

## Required Resources

Use the MortgageBroker project folder and the directory structure that you created in the previous tutorial to hold the project artifacts. The resources required for this scenario are described in the following table.

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

Resource Name	Description
normalLoan	The WSDL resource.
NormalLoan	The external business service used by ALSB.
LoanGateway3	The ALSB proxy service.

## Tasks in This Tutorial

In this tutorial, ALSB 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, ALSB raises a validation exception, and returns an error message to the client.

You will create a proxy service, LoanGateway3, and configure it to route a message to the NormalLoan business service. You will configure the behavior of the proxy service by performing the following tasks:

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

## Prepare Your Environment

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

## Create a Session

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

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

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

## Create the Resources

In this section, you will create the proxy service, LoanGateway3. You will use the existing normalLoan WSDL resource to validate the content of the message. If the loan application is valid and meets the expected criteria it is routed to the existing NormalLoan business service. The normalLoan WSDL resource and the NormalLoan business service were created in [To Create the normalLoan WSDL Resource](#) section in [Chapter 3, “Tutorial 1. Routing a Loan Application”](#).

## Create a Proxy Service

In this task, you create a proxy service that is used to route the loan application to the appropriate business service. For creating this proxy service instance, follow the tasks described in [To Create the Proxy Service](#), and use the proxy service name and the endpoint URI specified in [Table 5-2](#).

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

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

Figure 5-2 LoanGateway3 Proxy Service

Create a Proxy Service (MortgageBroker/ProxyServices/)	
General Configuration	
<b>Service Name*</b>	LoanGateway3
<b>Description</b>	<input type="text"/>
<b>Service Type*</b>	<p><b>Create a New Service</b></p> <p><input checked="" type="radio"/> WSDL Web Service <input type="text" value="MortgageBroker/WSDL/normalLoan"/> <input type="button" value="Browse..."/></p> <p><input type="radio"/> Messaging Service <input type="text" value="helloPort"/> (port)</p> <p><input type="radio"/> Any SOAP Service <input type="text" value="SOAP 1.1"/> <input type="button" value="v"/></p> <p><input type="radio"/> Any XML Service</p> <p><b>Create From Existing Service</b></p> <p><input type="radio"/> Business Service <input type="text"/> <input type="button" value="Browse..."/></p> <p><input type="radio"/> Proxy Service <input type="text"/> <input type="button" value="Browse..."/></p>
<b>Content Streaming</b>	<input type="checkbox"/> Enabled <ul style="list-style-type: none"> <li><input type="radio"/> Memory Buffer</li> <li><input type="radio"/> Disk Buffer</li> </ul>
<b>Publish To Registry</b>	<input type="checkbox"/>
<input type="button" value="Next &gt;&gt;"/>   <input type="button" value="Last &gt;&gt;"/>   <input type="button" value="Cancel"/>	

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

Figure 5-3 Create a Proxy Service–Summary of LoanGateway3

General Configuration	
Service Name	LoanGateway3
Description	
Service Type	Web Service - SOAP 1.1 (WSDL: MortgageBroker/WSDL/creditRatingService, port="helloPort")
Content Streaming	Disabled
Transport Configuration	
Protocol	http
Endpoint URI	/loan/LoanGateway3
Get All Headers	No
Headers	
HTTP Transport Configuration	
HTTPS required	No
Authentication	None
Operation Selection Configuration	
Enforce WS-I Compliance	No
Selection Algorithm	SOAP Body Type

<< Prev.      Save      Cancel

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

- LoanGateway1
- LoanGateway2
- LoanGateway3

## Configure the LoanGateway3 Proxy Service

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

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

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

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

- [Create a Pipeline Pair](#)
- [Configure the Request Actions for the LoanGateway3 Proxy Service](#)

## Create a Pipeline Pair

1. In the ALSB Console navigation pane, select the Project Explorer.

The Project Explorer pane is opened in the navigation pane.

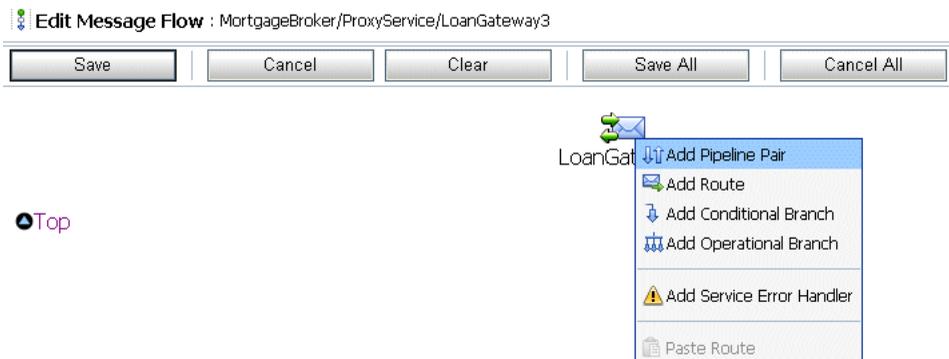
**Note:** You must be in a session to edit resources. If you have not already done so, begin a session (click **Create** in the **Change Center**) so that you can configure the proxy service and edit the message flow.

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

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

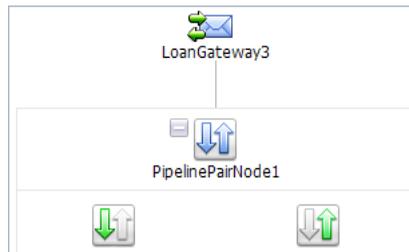
3. Click the LoanGateway3 node, then select **Add Pipeline Pair**.

**Figure 5-4 Add Pipeline Pair Node**



The PipelinePairNode1 is created and placed in the message flow. Request and response pipelines are displayed for this node as shown in [Figure 5-5](#).

Figure 5-5 Add Pipeline Pair



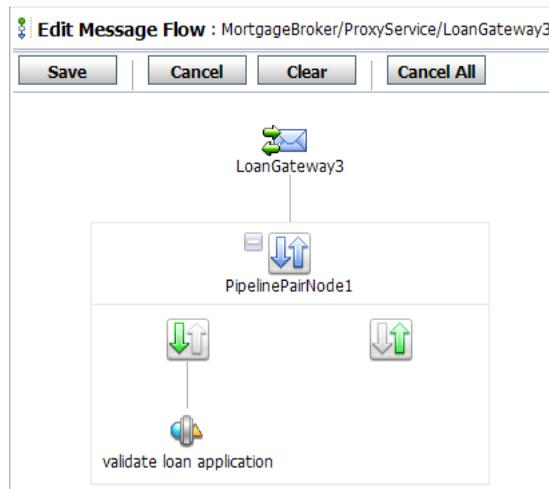
## Configure the Request Actions for the LoanGateway3 Proxy Service

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

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

The page is displayed as shown in [Figure 5-6](#).

Figure 5-6 Add Stage to Request Pipeline

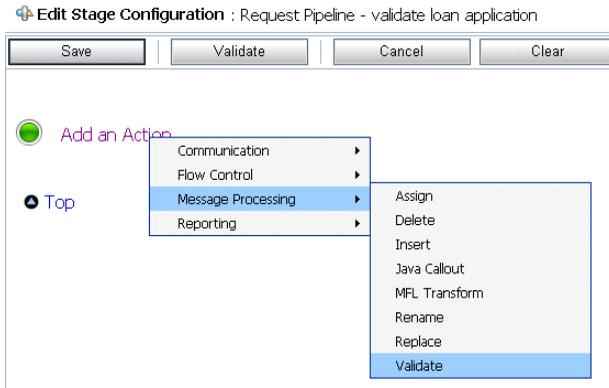


## Configure an Action to Perform Message Validation

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

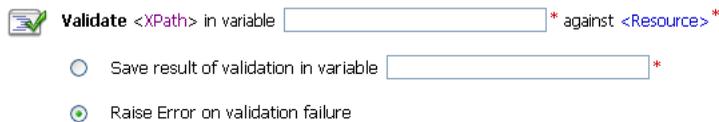
2. Select **Add an Action**, then select **Message Processing > Validate** from the drop-down list. The Validate action is added to the stage.

**Figure 5-7 Add Validate Action**



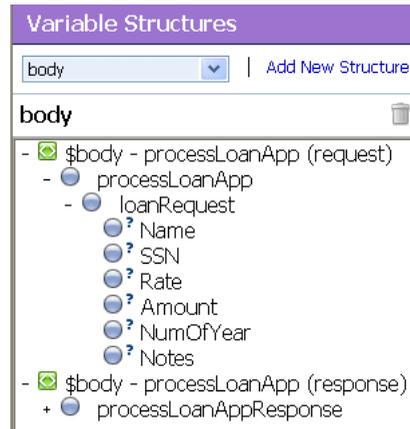
3. In the **Validate** statement, click the **<XPath>** link to edit the XPath expression. The XPath Expression Editor is displayed.

**Figure 5-8 XPath Expression Editor**



4. Click **Variable Structures**. The **Variable Structures** pane is displayed.
5. In the variables structure pane, from the Select Structure drop-down list, select body.  
A structural representation of the body element is displayed in the Variables Structure pane.
  - a. Click + beside the processLoanApp element to expand the element.  
The processLoanApp element is within the \$body - processLoanApp (request) element.
  - b. Click + to expand the loanRequest element.  
A graphical representation of the structure of the loan application document is displayed.

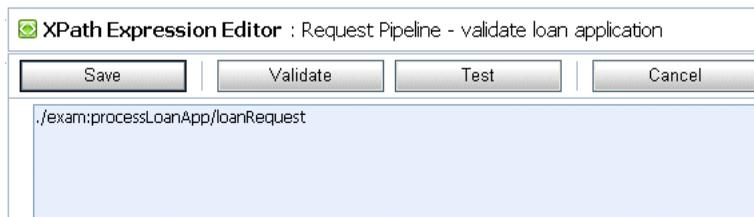
Figure 5-9 Variable Structures Pane



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

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

Figure 5-10 XQuery Expression Editor



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

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

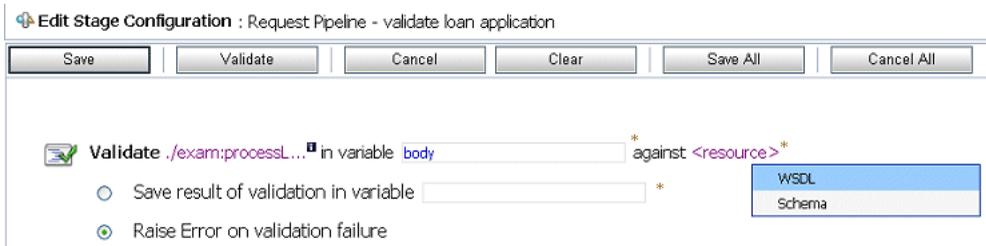
It is good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button. In this case, the expression is valid.

7. Click **Save**.

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

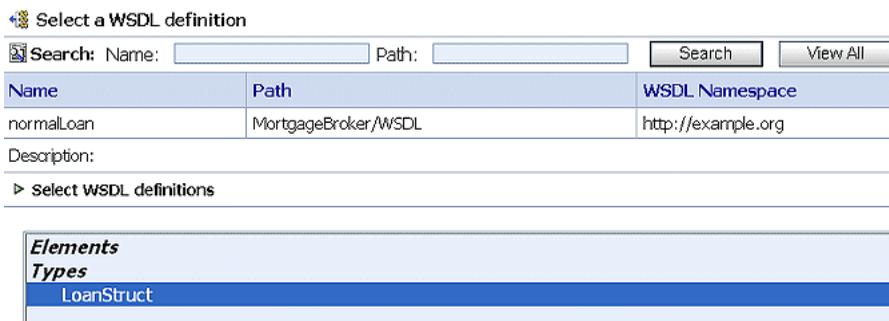
8. In the **Variable** text field, enter `body`.
9. Click **<resource>** and select WSDL from the list. The **Select a WSDL** page is displayed.  
You must validate the message against the resource since the proxy service is based on the WSDL resource that you originally created.

**Figure 5-11 WSDL Resource Selection**



10. In the **Select a WSDL** page, select the normalLoan WSDL resource.  
The Select WSDL Definitions pane is populated with the content categories of the WSDL.
  - a. In the Select WSDL Definitions pane, from the Types category, select the WSDL type for this WSDL as loanStruct.

**Figure 5-12 WSDL Definitions Pane**



- b. Click **Submit** to complete the selection.
11. Select the Raise Error on validation failure option. The Validate action will be displayed as shown in [Figure 5-13](#).

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

**Figure 5-13 Configured Message Validate Action**



**Note:** Message validation must be done as early as possible in a message flow. This prevents unnecessary processing of invalid message content. By raising an exception, further message processing is stopped and an error response is immediately returned to the client.

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

## To Configure the Stage Error Handler

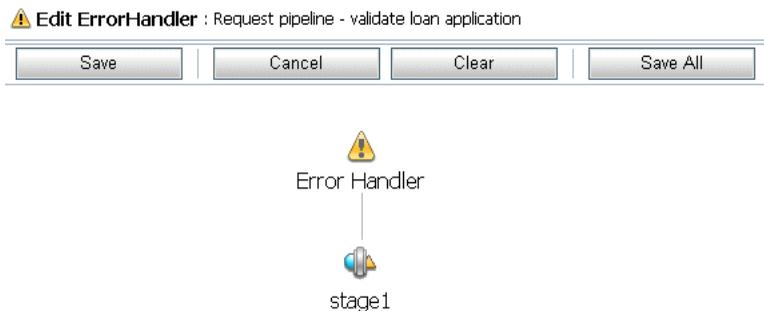
You must perform the following tasks to configure the Stage Error Handler:

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

## Add Stage Error Handler Element

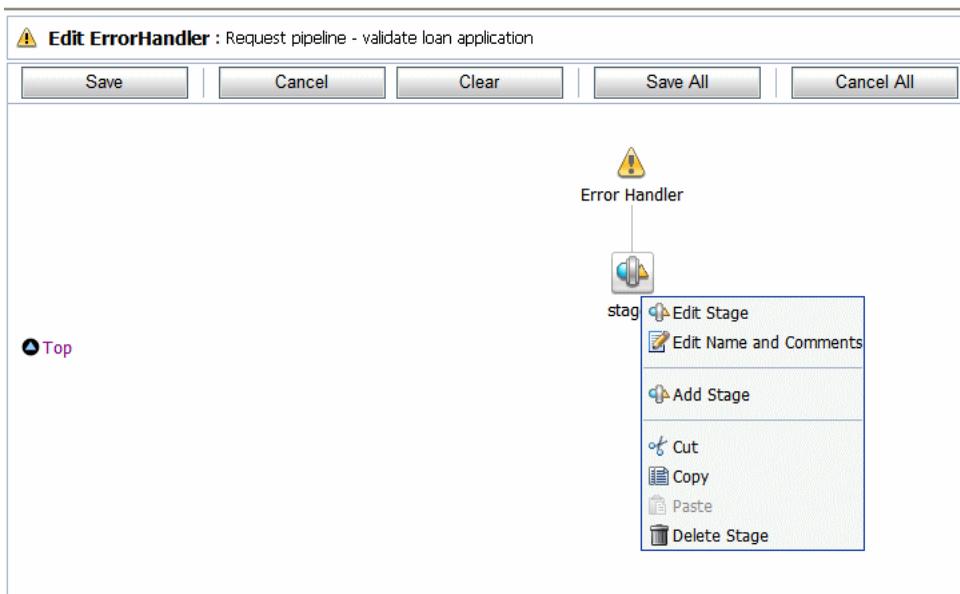
1. Click validate loan application, then select **Add Stage Error Handler**. The **Edit Error Handler** page is displayed.
2. Click **Error Handler**, then select **Add Stage**. The stage1 node is displayed.

Figure 5-14 Add Stage Error Handler



3. Click stage1 and select **Edit Name and Comments** as shown in [Figure 5-15](#).

Figure 5-15 Edit Stage Error Handler



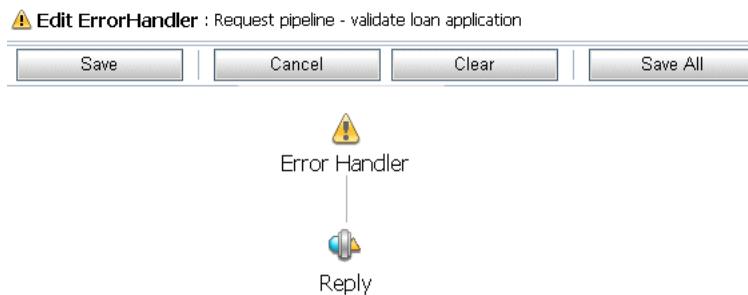
4. Change the stage name to Reply, then click **Save**.

**Figure 5-16 Edit Name and Comments of Stage**



The **Edit Stage Configuration** page is displayed as shown in [Figure 5-17](#).

**Figure 5-17 Edit Error Handler**

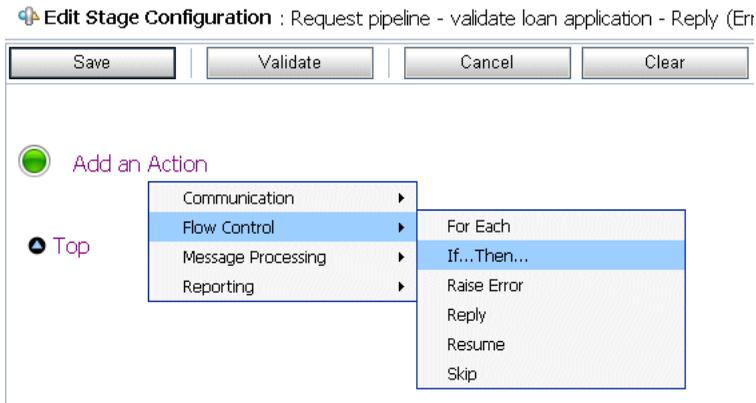


5. To edit the stage, click **Reply**, then select **Edit Stage**.

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

1. Select **Add an Action**, then select **Flow Control > If...Then...**

**Figure 5-18 Add Flow Control Action**



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

2. Click the **If...Then...** action **<Condition>** link to specify the expression that needs to be validated. The **Xquery Condition Editor** page is displayed.
  - a. Select the **Builder** pane.
  - b. In the **Comparison Expression** pane within the **Expression Builder** pane, enter `$fault/ctx:errorCode` in the Operand textbox.

**Figure 5-19 Expression Builder Pane**



- c. Select the = Operator from the drop-down list.
- d. Enter "BEA-382505" (including double quotes) in the Value pane.
- e. Select the radio button for the And conjunction.

The **Expression Builder** pane is displayed as in [Figure 5-20](#):

**Figure 5-20 Conditions in the Expression Builder**

Expression Builder				
▶ Comparison Expression   Unary Expression				
Operand	Operator	Value	Conjunction	
\$fault/ctx:errorCode	=	"BEA-382505"	<input checked="" type="radio"/> <b>And</b> <input type="radio"/> <b>Or</b>	<input type="button" value="Add"/> <input type="button" value="Clear"/>

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

The following expression is displayed in the textbox:

**Figure 5-21 Expressions Textbox**

Expressions	
<input type="button" value="↑"/> <input type="button" value="↓"/> <input type="button" value="✎"/> <input type="button" value="🗑"/>	\$fault/ctx:errorCode = "BEA-382505"

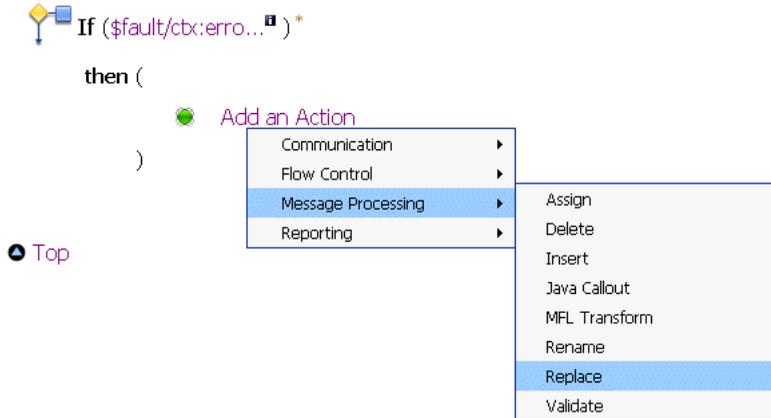
3. Click **Save** to save the expression and return to the **Edit Stage Configuration** page.

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

1. To add a **Replace** select **Add an Action** within the **Then...** portion of the **If...Then** action.
 

**Note:** When adding multiple actions on a page, click the action preceding the placement of the new action and select the action to be added from there. actions are added in sequence.
2. Click **to view available options** icon, select **Message Processing > Replace**.

Figure 5-22 Add a Replace Action



The replace statement will be displayed as shown in Figure 5-23.

Figure 5-23 Replace Statement



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

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

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

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

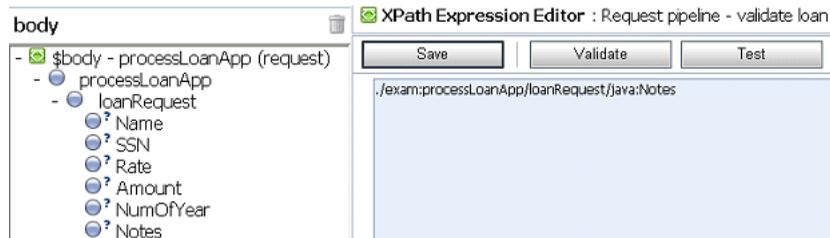
The processLoanApp element is within the **\$body - processLoanApp (request)** element.

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

c. Select the Node icon associated with the Notes element of \$body, drag-and-drop it on the XPath expression text box. The following XPath expression is written in the text box.

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

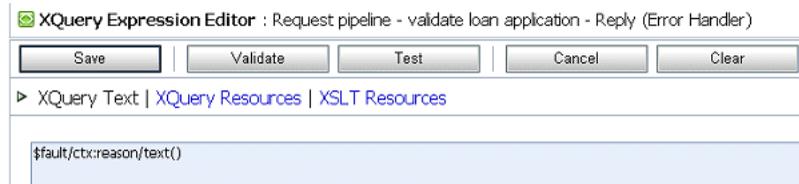
Figure 5-24 Replace Statement XPath Expression



5. Click **Validate** to validate the expression, then click **Save**.
6. In the **Variable** text field, enter `body`.
7. Click the **<Expression>** link to display the **XQuery Expression Editor** page.
8. In the XQuery Text pane, enter:
 

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

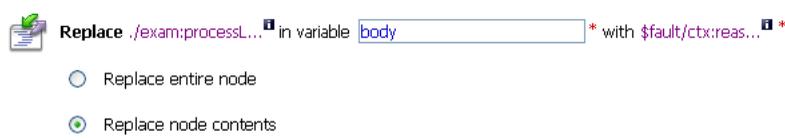
Figure 5-25 Replace Statement XQuery Expression



9. Click **Validate**, then **Save**.
10. Select **Replace node contents** option. You need not replace the entire node.

The Replace action is displayed as shown in [Figure 5-26](#).

Figure 5-26 Add Error Handler Using Replace Action

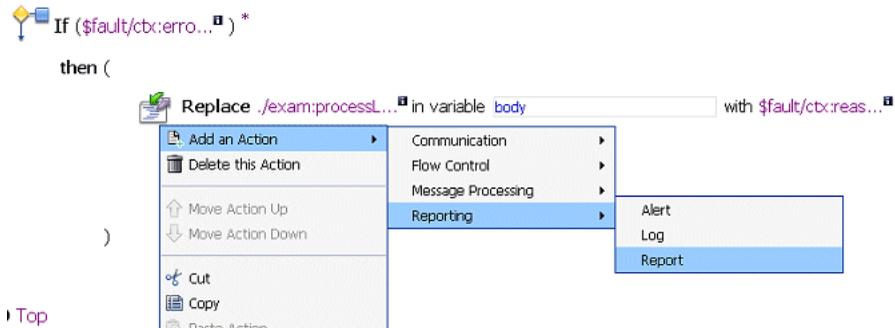


## Add a Report Action to Report the Error

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

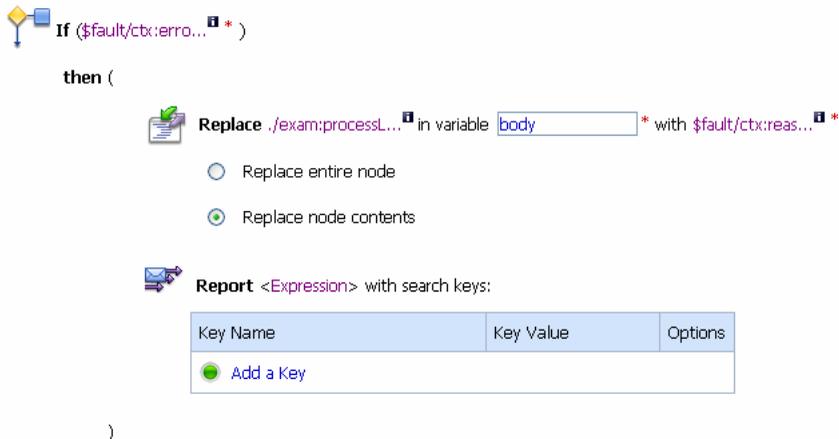
1. Click the **Replace icon** and select **Add an Action > Reporting > Report**. The report action is added to the page.

Figure 5-27 Adding a Report Action



The Stage Error Handler is displayed as shown in [Figure 5-28](#).

Figure 5-28 Stage Error Handler with Report Action



2. In the **Report** statement, click the **<Expression>** link. The **XQuery Expression Editor** page is displayed.
3. Accept the default selection of **XQuery Text** in the page.
4. In the Variables Structures pane, select `body - processLoanApp` from the Select Structure drop-down list.
  - 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 typed 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 in the XQuery Expression text box.

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

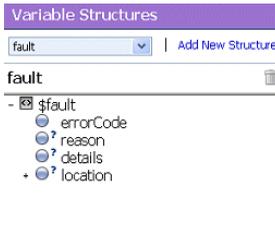
The **Edit Stage Configuration** page is displayed.
5. In the key-value pair table, select the Add a key link.
  - a. Enter a name for the key. In this case, enter `errorCode`.

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



- b. In the key value field, click **<XPath>** link. The **XPath Expression Editor** is displayed.
6. In the **Variable Structures** pane, select `fault` from the **Select Structure** drop-down list.

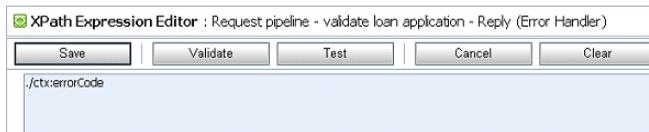
**Figure 5-30 Fault Variable Structure**



- a. Drag-and-drop the Node icon associated with the errorCode in the XPath Expression text box.

Type `./ctx:errorCode` expression in the text box.

**Figure 5-31 Report Action ErrorCode Expression**



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

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

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

The **Report** action must be displayed as shown in [Figure 5-32](#).

**Figure 5-32 Configured Report Action**



The **Edit Stage Configuration** page is displayed as shown in [Figure 5-33](#).

**Figure 5-33 Request Actions for LoanGateway3 Proxy Service**

🔑 If (`$fault/ctx:erro... *`)  
 then (  
     📄 **Replace** `./exam:processL... *` in variable `body` with `$fault/ctx:reas... *`  
          Replace entire node  
          Replace node contents  
     📧 **Report** `$body` with search keys:  

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

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

1. Click the **If...Then...** icon.
2. Select **Add else Condition** to add an **Else** action to the stage error handler.

### Add a Replace Action Within the Else Action

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

The **Edit Stage Configuration** page is displayed as shown in [Figure 5-34](#).

**Figure 5-34 Edit Stage Configuration Page - Else Replace Action**

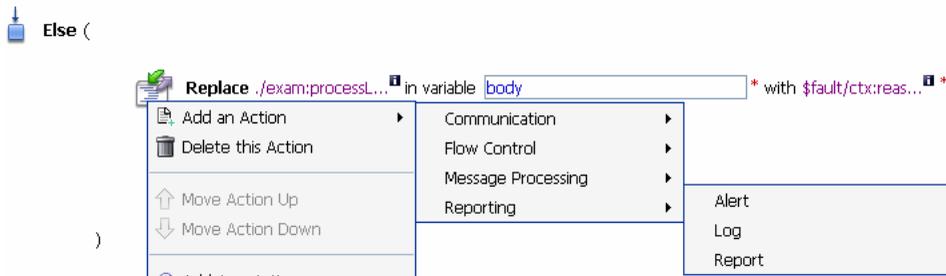
⬇ Else (  
     📄 **Replace** `./exam:processL... *` in variable `body` with `$fault/ctx:reas... *`  
          Replace entire node  
          Replace node contents

### Add a Log Action Within the Else Action

1. Click the **Replace** icon within the **Else** action.

2. Select **Add an Action > Reporting > Log**.

**Figure 5-35 Reporting Log Action**



The reply error handler **Else** action is displayed as shown in [Figure 5-35](#). Add the Log action

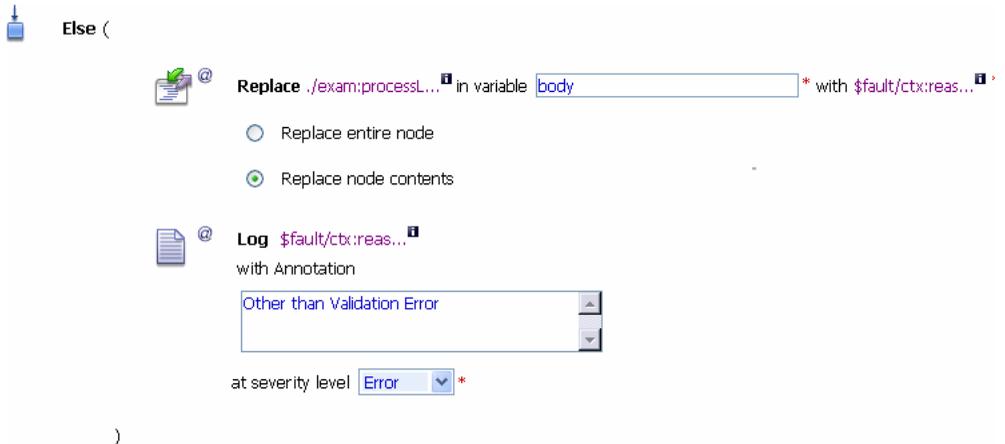
**Figure 5-36 Reporting Log Action**



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

The completed **Else** condition must be displayed as shown in [Figure 5-37](#).

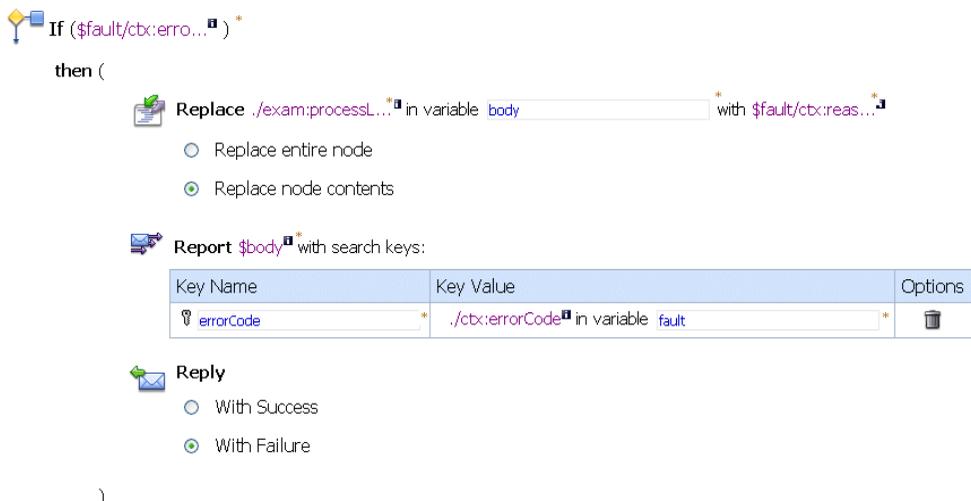
**Figure 5-37 LoanGateway3 Proxy Service Request Else Actions**



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

1. Click the **Report** icon under the **If...Then** action.
2. Select **Add an Action > Flow Control > Reply**. The reply action is added to the page.
3. Select **With Failure** option. The **Report Reply** action is displayed as shown in [Figure 5-38](#).

**Figure 5-38 Report Reply Action with Failure Code**



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

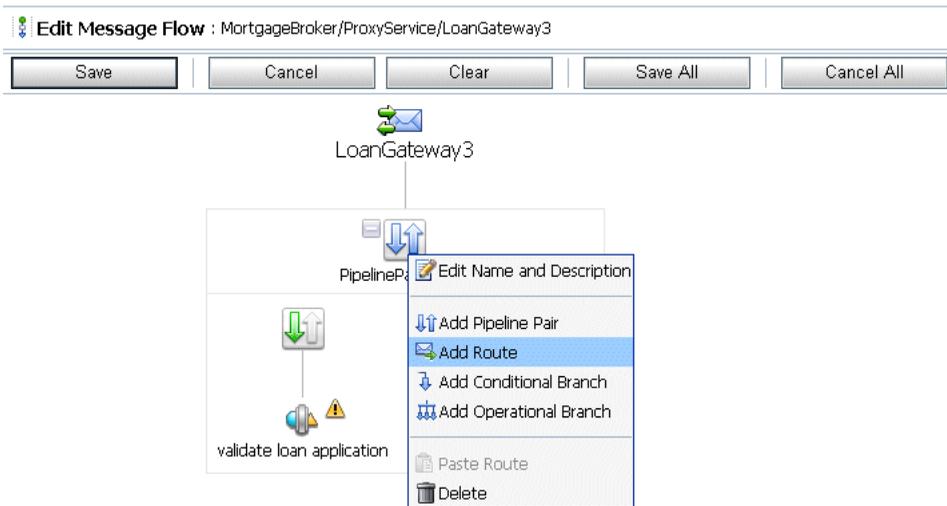
## Add a Default Routing Case

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

### Configure the Route Node

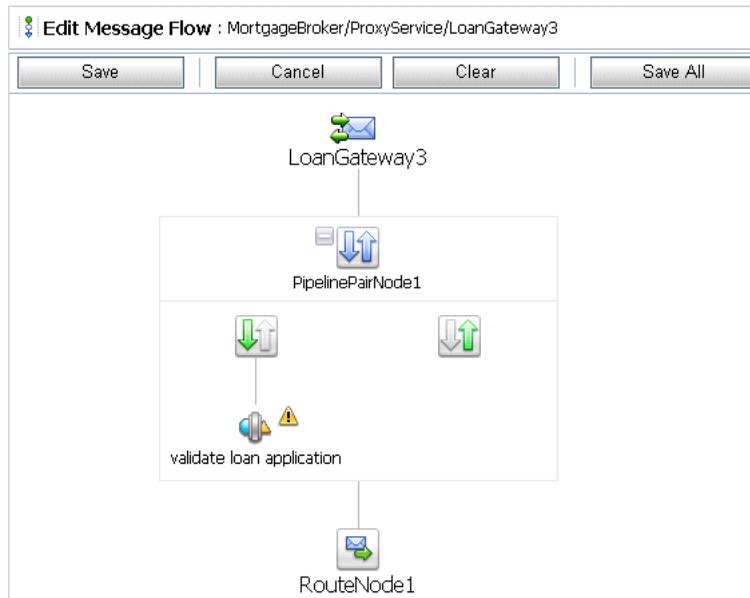
1. Click **PipelinePairNode1**, then select **Add Route**.

Figure 5-39 Pipeline Route Node



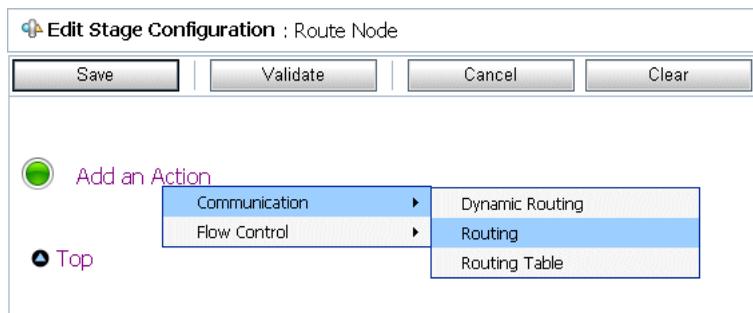
The **Edit Message Flow** page is displayed as shown in [Figure 5-40](#).

**Figure 5-40 Edit Message Flow Page - Pipeline Route Node**



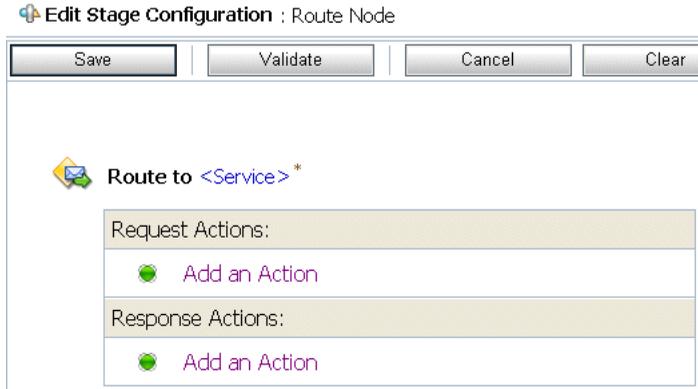
2. Click **RouteNode1**, then select **Edit Route**. The **Edit Stage Configuration** page is displayed.
3. Click **Add an Action**, then select **Communication > Routing**.

**Figure 5-41 RouteNode1 Communication Routing Action**



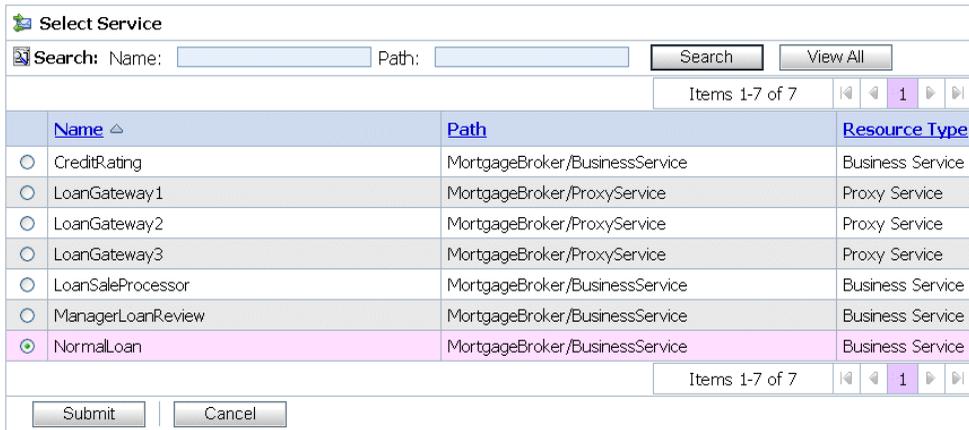
The Route to action is displayed in the **Edit Stage Configuration** page as shown [Figure 5-42](#).

**Figure 5-42 Route to Statement**



4. In the route statement, click the **<Service>** link to define the service to which you want to route messages. The **Select Service** page is displayed as shown in [Figure 5-43](#).

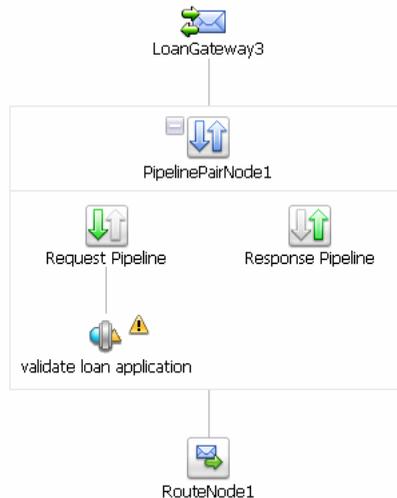
**Figure 5-43 Route to Statement Service**



- a. Select the NormalLoan business service.
  - b. Click **Submit** and return to the **Edit Stage Configuration** page.
5. In the Operation drop-down list, select **processLoanApp**. Click **Save**.

This is the operation on the NormalLoan business service that is invoked at run time. You have now defined the case for routing the loan application to the NormalLoan business service. The **Edit Message Flow** page is displayed as shown in [Figure 5-44](#).

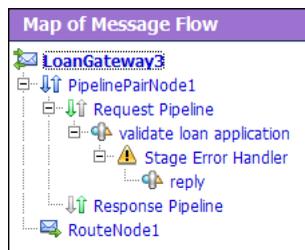
Figure 5-44 LoanGateway3 Proxy Service Message Flow



6. Click **Save** to save the configuration and return to the **MortgageBroker/ProxyService** page.
7. Click **Activate**, then Submit.

You can view the map of the message flow of the proxy service as shown in [Figure 5-45](#).

Figure 5-45 Map of Message Flow for LoanGateway3 Proxy Service



## To Test the Loan Application Routing Configuration Using Test Console

Now that you have configured ALSB to work with the client and the target business services, you must test the configuration. To test the routing configuration, you must change the value of the Loan duration (in years) field in the build.xml file. Then you must test the routing logic by

verifying whether the proxy service behavior changes for different loan values. If the loan value entered is a non integer, an exception must be triggered and an error message displayed in the Notes field of the message. If the loan value entered is an integer, the message must be routed to the NormalLoan business service. The message returned in the command window indicates whether a validation exception occurred or the message was routed to the appropriate business service.

To test the loan application using the test console, you must first deploy the client jars in the WLS console. For more information see, [Load the Client Application](#).

### To test the Validation of the Loan Application by LoanGateway3

#### Scenario One

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page. Click on **Launch Test Console** icon in **Actions** column for LoanGateway3 to launch the test console (see [Figure 3-40](#)).
4. Replace the default payload in loanRequest field with the following code:

```
<loanRequest xmlns:java="java:normal.client">
  <!--Optional:-->
  <java:Name>Smith</java:Name>
  <!--Optional:-->
  <java:SSN>1234567</java:SSN>
  <!--Optional:-->
  <java:Rate>5</java:Rate>
  <!--Optional:-->
  <java:Amount>900000000</java:Amount>
  <!--Optional:-->
  <java:NumOfYear>10.1</java:NumOfYear>
  <!--Optional:-->
  <java:Notes>Large Loan Processing</java:Notes>
</loanRequest>
```

Accept other default settings and click **Execute**. The response as in [Figure 5-46](#).

**Figure 5-46 Response from Error Handler**



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

## Scenario Two

1. Start WLS Server in ServiceBusTutorial Domain
2. In ALSB Console activate the current session.
3. Click on **Resource Browser** in the left panel to navigate to **Summary of Proxy Services** page. Click on **Launch Test Console** icon in **Actions** column for LoanGateway1 to launch the test console (see [Figure 3-40](#)).
4. Replace the default payload in loanRequest field with the following code:

```

<loanRequest xmlns:java="java:normal.client">
  <!--Optional:-->

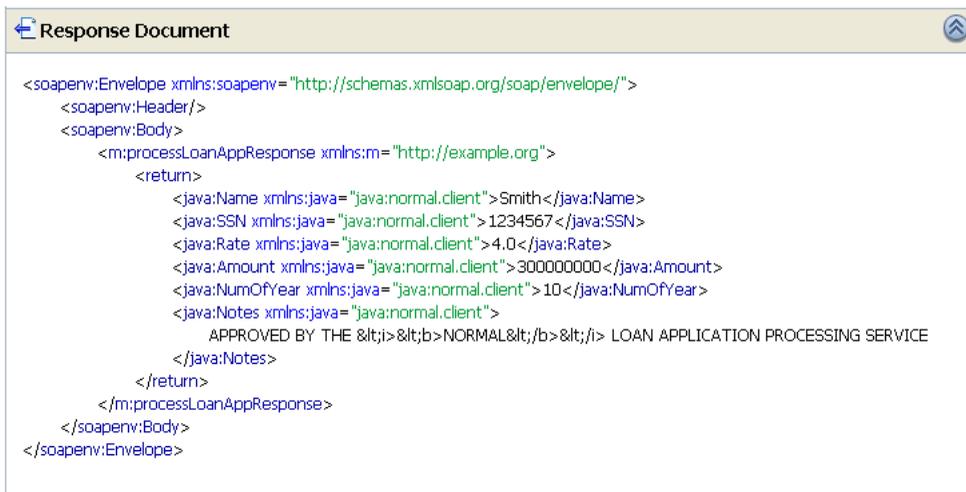
```

## Tutorial 3. Validating a Loan Application

```
<java:Name>Smith</java:Name>
<!--Optional:-->
<java:SSN>1234567</java:SSN>
<!--Optional:-->
<java:Rate>4</java:Rate>
<!--Optional:-->
<java:Amount>300000000</java:Amount>
<!--Optional:-->
<java:NumOfYear>10</java:NumOfYear>
<!--Optional:-->
<java:Notes>Normal Loan</java:Notes>
</loanRequest>
```

Accept other default settings and click **Execute**. The response as in

**Figure 5-47 Response From Normal Loan Processor**



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <m:processLoanAppResponse xmlns:m="http://example.org">
      <return>
        <java:Name xmlns:java="java:normal.client">Smith</java:Name>
        <java:SSN xmlns:java="java:normal.client">1234567</java:SSN>
        <java:Rate xmlns:java="java:normal.client">4.0</java:Rate>
        <java:Amount xmlns:java="java:normal.client">300000000</java:Amount>
        <java:NumOfYear xmlns:java="java:normal.client">10</java:NumOfYear>
        <java:Notes xmlns:java="java:normal.client">
          APPROVED BY THE &lt;i>&lt;b>NORMAL&lt;/b>&lt;/i> LOAN APPLICATION PROCESSING SERVICE
        </java:Notes>
      </return>
    </m:processLoanAppResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

The response message indicates that the NormalLoan business service processed the loan application because the loan duration value was an integer.

## Step 7: Using Reporting and Monitoring

Now that you have tested the proxy service, you can explore the reporting and monitoring features of the ALSB Console.

**Note:** You must have run the tests in [To Test the Loan Application Routing Configuration Using Test Console](#) multiple times (at least five times) to generate data to enable you to explore the reporting and monitoring features.

### Reporting

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

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

For more information, see:

- [Reporting](#) in AquaLogic Service Bus Operations Guide
- [Reporting](#) in Using the AquaLogic Service Bus Console

### Monitoring

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

For more information, see:

- [Monitoring](#) in AquaLogic Service Bus Operations Guide
- [Monitoring](#) in Using the AquaLogic Service Bus Console