

# **Oracle® Service Bus**

Tutorials

10g Release 3 (10.3)

October 2008

**ORACLE®**

Copyright © 2007, 2008, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

## 1. Introduction to the Oracle Service Bus Tutorials

Oracle Service Bus Concepts . . . . .	1-2
Proxy Services and Business Services . . . . .	1-3
Message Flows and Pipelines . . . . .	1-4
Pipeline Pairs . . . . .	1-5
Overview of the Tutorials . . . . .	1-6
How to Use the Tutorials . . . . .	1-7

## 2. Getting Started with the Oracle Service Bus Tutorials

Scenarios in the Tutorials . . . . .	2-2
Setting Up the Tutorials . . . . .	2-2
Install Oracle Service Bus . . . . .	2-3
Create an Oracle Service Bus Domain . . . . .	2-4
Start Oracle Service Bus . . . . .	2-10
Load the Client Application . . . . .	2-11
Log in to Oracle Service Bus Console . . . . .	2-13
Where to Go from Here . . . . .	2-14

## 3. Tutorial 1. Routing a Loan Application

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

Required Resources . . . . .	3-3
Tasks in This Tutorial . . . . .	3-4
Prepare the Environment . . . . .	3-4
Create a Session and Set up a Project . . . . .	3-4
Create the WSDL Resources . . . . .	3-7
Create a Proxy Service . . . . .	3-10
Create the Business Services . . . . .	3-15
Configure the LoanGateway1 Proxy Service . . . . .	3-19
Test Your Loan Application Routing Configuration Using the Test Console . . . . .	3-28
Where to go From Here . . . . .	3-32

## 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 Oracle Service Bus . . . . .	4-2
Required Resources . . . . .	4-3
Tasks in This Tutorial . . . . .	4-3
Prepare the Environment . . . . .	4-4
Create a Session . . . . .	4-4
Create the Resources . . . . .	4-4
Configure Routing for LoanGateway2 Proxy Service . . . . .	4-8
Configure Routing Request Actions for LoanGateway2 Proxy Service . . . . .	4-12
Configure Routing Response Actions for LoanGateway2 Proxy Service . . . . .	4-22
Add a Default Routing Case . . . . .	4-25
Test the Loan Application Routing Configuration Using the Test Console . . . . .	4-26
Best Practices . . . . .	4-29

## 5. Tutorial 3. Validating a Loan Application

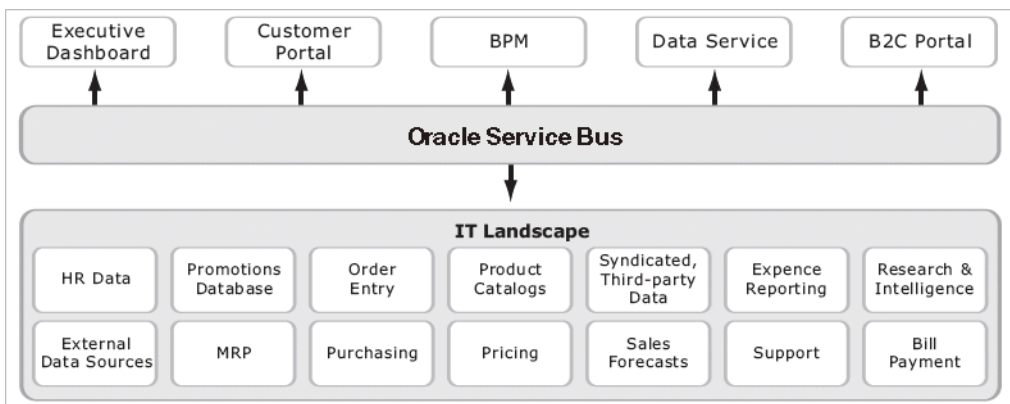
Prerequisites . . . . .	5-2
Tutorial Objectives . . . . .	5-2
Features Highlighted . . . . .	5-2
Definition of the Scenario . . . . .	5-2
Overview of the Run-Time Process in Oracle Service Bus . . . . .	5-3
Required Resources . . . . .	5-4
Tasks in This Tutorial . . . . .	5-4
Prepare Your Environment . . . . .	5-4
Create a Session . . . . .	5-5
Create the Resources . . . . .	5-5
Configure the LoanGateway3 Proxy Service . . . . .	5-6
Add a Default Routing Case . . . . .	5-17
To Test the Loan Application Routing Configuration Using Test Console . . . . .	5-19
Step 7: Using Reporting and Monitoring . . . . .	5-22



# Introduction to the Oracle Service Bus Tutorials

Oracle 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 Oracle Service Bus as a service intermediary in an enterprise IT SOA architecture.

**Figure 1-1 Oracle Service Bus Intermediary**



Oracle Service Bus Console enables you to:

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

This topic consists of the following sections:

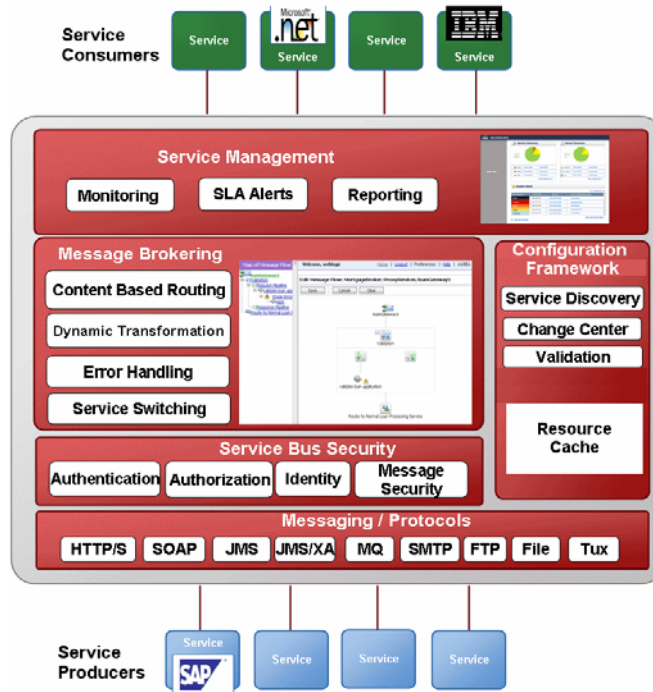
- [Oracle Service Bus Concepts](#)
- [Overview of the Tutorials](#)
- [How to Use the Tutorials](#)

## Oracle Service Bus Concepts

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



Figure 1-2 Oracle Service Bus Architecture



## Proxy Services and Business Services

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

Oracle Service Bus 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 Oracle Service Bus Console. A proxy service can route messages to multiple business services; you can choose to configure a proxy service with an interface that is independent of the business services with which the proxy service communicates. In such cases, you can configure a message

flow definition to route a message to the appropriate business service and map the message data into the format required by the business service's interface.

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

Business services are Oracle Service Bus definitions of the enterprise services that exchange messages during business processes. A business service and its interface can be defined and configured using the Oracle Service Bus 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 Oracle Service Bus Concepts and Architecture. For information on how to configure a business service using the Oracle Service Bus Console, see “Adding a Business Service” in [Business Services](#) in Using the Oracle Service Bus Console.

## Message Flows and Pipelines

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

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

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

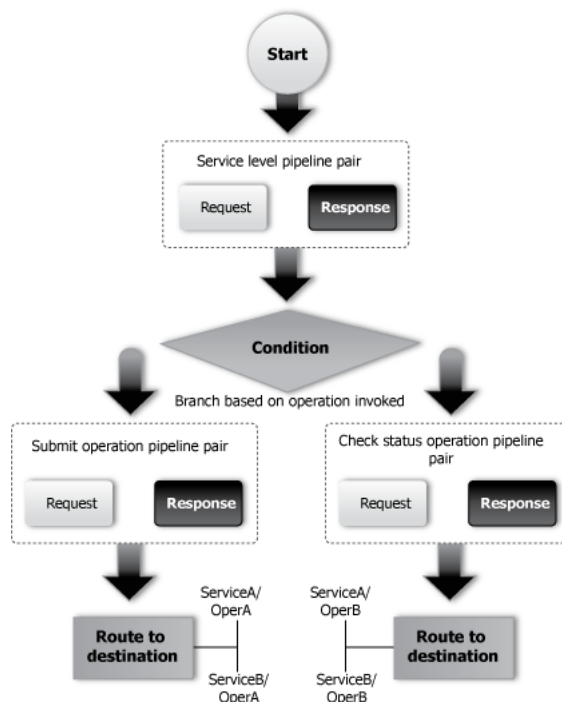
## Pipeline Pairs

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

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

Figure 1-3 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 Oracle Service Bus Concepts and Architecture.

## Overview of the Tutorials

This tutorial shows how Oracle Service Bus is used to route a loan application to appropriate business services based on the configurations set in the proxy service. Oracle Service Bus 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 Oracle Service Bus uses session management to allow configuration changes locally, and then instantiate the changes in the core environment.
- Configure resources such as WSDLs, Service Callouts, XQueries, and business services.
- Route messages according to XQuery-based policies or callouts to external Web services.
- See how Oracle Service Bus manages synchronous and asynchronous messages.
- Define a transformation on an XML message, based on XQuery, supporting a callout to a Web service to gather additional data for the transformation.
- See how message processing by Oracle Service Bus is driven by metadata specified as the *message flow definition* for a proxy service in the Oracle Service Bus Console.
- Test proxy services and business services.

# How to Use the Tutorials

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

- Provide you an understanding of the concepts underlying Oracle Service Bus.
- Show you how to navigate through the Oracle Service Bus 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 Oracle Service Bus, 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 Oracle Service Bus facilitates the routing of messages within an enterprise.
- [Tutorial 2. Transforming a Loan Application](#) describes how a proxy service is configured with a routing node, transformations, and a Web service callout (Service Callout) to allow content-based routing and message enrichment through Oracle Service Bus.
- [Tutorial 3. Validating a Loan Application](#) describes how a Oracle Service Bus proxy service can be configured to validate a message passing from a client to a business service.

# Document Conventions

The following conventions are used throughout this document:

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

## Related Documents

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

# Getting Started with the Oracle Service Bus Tutorials

The Oracle Service Bus tutorials are based on a typical Web services scenario that uses Oracle Service Bus for configuring business services and adaptive message routing. This section describes the tasks required to configure the Oracle Service Bus and Weblogic Server environment to deploy the business cases used in the tutorials. The Oracle Service Bus example suite is used as a basis for instruction in the following tutorials. (See [Oracle 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 Oracle Service Bus 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 Oracle Service Bus to route loan applications to appropriate business services. The loan applications are routed to different business services depending on qualifying criteria, such as the requested interest rate and the requested principal amount. The applicant's credit rating information is required to complete the loan application when the principal amount is greater than US \$25 million.

In the first tutorial, you will become familiar with the basic functionality of Oracle Service Bus 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 Oracle Service Bus
2. Create an Oracle Service Bus Domain
3. Change Domain Configuration Settings
4. Start Oracle Service Bus
5. Build and Deploy Business Services
6. Log into Oracle Service Bus Consoles

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

```
BEA_HOME\osb_10.3\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\osb_10.3\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).
SimpleBean.java	The jws-181 Web service definition.
/client/LoanStruct.java	The definition of the message.
build.xml	The build script that is used to build each of the services using ant.
*.wsdl	A Web service Definition Language (WSDL) file defines each of the services.
/client/Main.java	The Java client with which each of the business services interacts.

## Install Oracle Service Bus

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

## Create an Oracle Service Bus Domain

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

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

**Start > All Programs > Oracle WebLogic > WebLogic Server 10gR3 > Tools > Configuration Wizard**

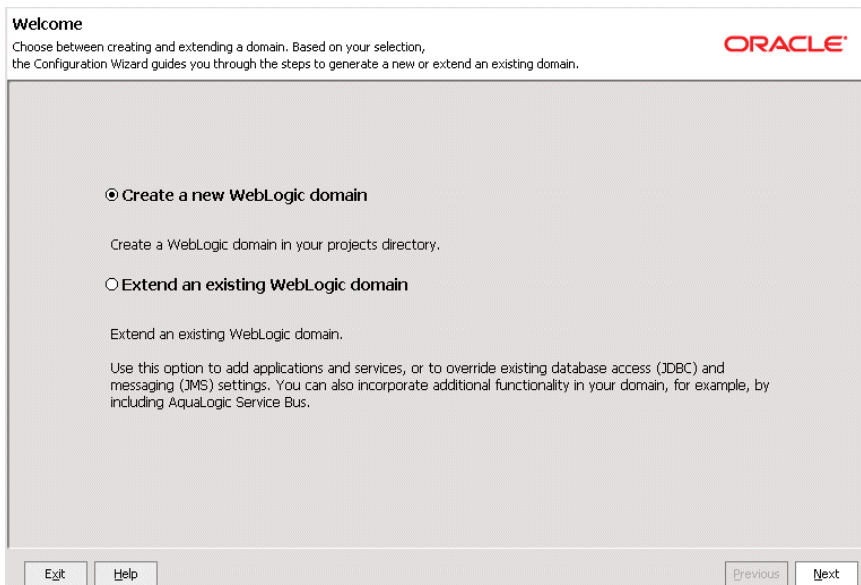
2. Using the Configuration Wizard, select the following product components:

- WebLogic Server
- Oracle 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 **Oracle 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 products:**

- ☒ WebLogic Server (Required)
- ☐ Workshop For WebLogic 10.3
- ☒ AquaLogic Service Bus

☐ **Base this domain on an existing template**

Template location:

- c. In the Configure Administrator Username and Password page, configure a new administrator user name and password. You can use the default weblogic user name, 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.

Discard Changes

\*User name: newuser

\*User password: \*\*\*\*\*

\*Confirm user password: \*\*\*\*\*

Description: This user is the default administrator.

Exit Help Previous Next

- 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**

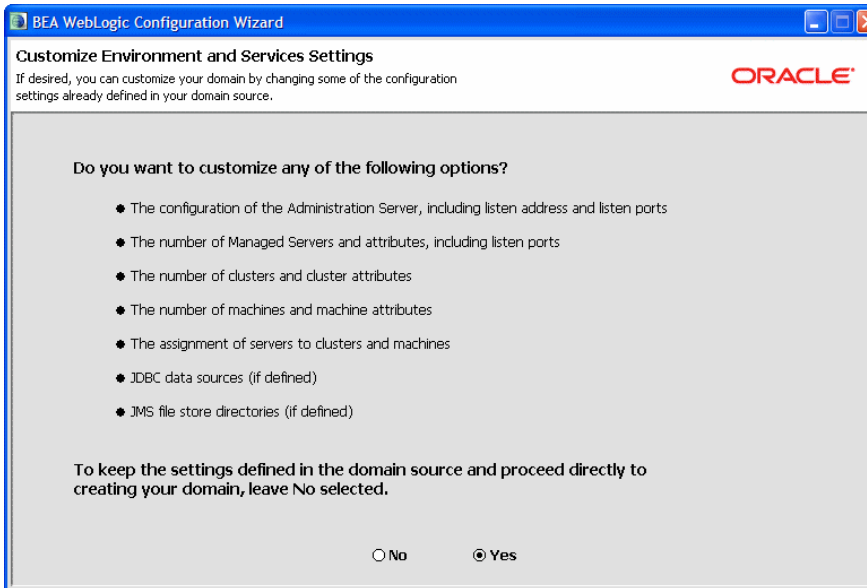
☒ **BEA Supplied JDKs**

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. Click **Next** in the Configure RDBMS Security Store Database window.
- g. 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.

ORACLE

Discard Changes

\*Name:

Listen address:

Listen port:

SSL listen port:

SSL enabled: ☐

Exit Previous Next

- h. 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.
- i. In the Create Weblogic Domain page specify the domain name as **ServiceBusTutorial**. Click **Create**.

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

Create WebLogic Domain

When you click Create the domain will be generated in the location specified.

ORACLE

Enter the name and location for the domain:

Domain name: servicebustutorial

Domain location: D:\bea\user\_projects\domains

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

## Start Oracle Service Bus

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

- On Windows systems, from the Windows Start menu, select:

**Start > All Programs > Oracle WebLogic > User Projects > *domain\_name* > Start Server for Oracle Service Bus Domain.**

For example, to start the server for the ServiceBusTutorial domain select **Start > All Programs > Oracle WebLogic > User Projects > ServiceBusTutorial > Start Server for Oracle Service Bus Domain**

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



When Oracle Service Bus 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 WebLogic Server Administration Console, which are available at

`{BEA_HOME}\osb_10.3\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. Go to the WebLogic Server Administration Console: `http://localhost:7001/console`.
2. On the Home page, click **Deployments** 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

Install




















Update

Delete

Start

Stop

Showing 1 to 10 of 79 Previous | [Next](#)

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input type="checkbox"/>	 ALDSP Transport Provider	Active	 OK	Web Application	161
<input type="checkbox"/>	 aldsp_transport-110n(3.0,3.0)	Active		Library	160
<input type="checkbox"/>	 ALSB Cluster Singleton Marker Application	Active	 OK	Enterprise Application	80
<input type="checkbox"/>	 ALSB Domain Singleton Marker Application	Active	 OK	Enterprise Application	85
<input type="checkbox"/>	 ALSB FLOW Kernel	Active	 OK	Enterprise Application	600
<input type="checkbox"/>	 ALSB Framework Starter Application	Active	 OK	Enterprise Application	90
<input type="checkbox"/>	 ALSB Logging	Active	 OK	Enterprise Application	440
<input type="checkbox"/>	 ALSB Publish	Active	 OK	Enterprise Application	430
<input type="checkbox"/>	 ALSB Resource	Active	 OK	Web Application	500
<input type="checkbox"/>	 ALSB Routing	Active	 OK	Enterprise Application	410

Install

Update

Delete

Start

Stop

Showing 1 to 10 of 79 Previous | [Next](#)

- a. Click **Install**.
- b. Navigate to  
`BEA_HOME\osb_10.3\samples\servicebus\examples\build\webservices`
- c. Select **creditLoan\_jws\_basic\_ejb**. Click **Next** to choose the targeting style.
- d. In Choose Targeting Style page accept defaults (deploy as an application). Click **Next** to go to Optional Settings page.
- e. In Optional Settings page accept defaults and click **Finish**. The application is deployed automatically.

Repeat the previous steps to deploy **largeLoan\_jws\_basic\_ejb**, **manager\_jws\_basic\_ejb**, and **normalLoan\_jws\_basic\_ejb** in the WebLogic Server console.

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

## Log in to Oracle Service Bus Console

Oracle Service Bus Console is a Web services management dashboard that 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 Oracle Service Bus environment by displaying detailed statistics about servers, services, and alerts. The Oracle Service Bus Console also enables you respond rapidly and effectively to changes in your service-oriented environment.

### To log in to Oracle Service Bus Console

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

`http://host:port/sbconsole`

Enter `http://localhost:7001/sbconsole` URL in your browser to open the Oracle Service Bus Console for the ServiceBusTutorial domain. Log in if necessary with Oracle Service Bus Console the user name and password that you specified when you created the Oracle Service Bus tutorial domain.

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

**Start > All Programs > Oracle WebLogic > User Projects > ServiceBusTutorial > Oracle Service Bus Admin Console** for Oracle Service Bus 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 Oracle Service Bus with the resources required for the loan application routing scenario. Each of the tutorials include instructions to design and configure the Oracle Service Bus resources, and procedures you can use to test the completed configurations.

# Tutorial 1. Routing a Loan Application

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

This section includes the following topics:

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

## Prerequisites

You must complete [Getting Started with the Oracle Service Bus 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 Oracle Service Bus Console. It includes the following tasks:

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

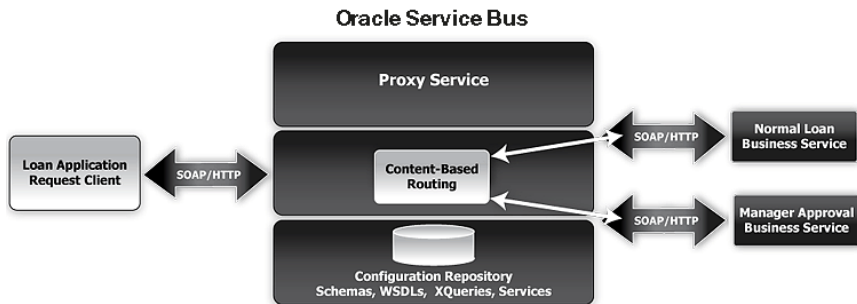
This tutorial introduces you to:

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

## Definition of the Scenario

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

[Figure 3-1](#) summarizes the logical architecture to support this scenario. It illustrates how Oracle Service Bus mediates the messaging between the clients and the business services in your enterprise.

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

## Overview of the Run-Time Process in Oracle Service Bus

A client sends a loan application to a proxy service named LoanGateway1. The default proxy service has a conditional routing stage that checks the value of the requested interest rate in the loan application document. If the interest rate is less than 5%, the loan 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 3-31](#).

**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 Oracle Service Bus.
ManagerLoanReview	This is business service used by Oracle Service Bus.
LoanGateway1	This is the Oracle Service Bus proxy service.

## Tasks in This Tutorial

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

### Prepare the Environment

Ensure that Oracle Service Bus 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 Oracle Service Bus Console, you can monitor resources and configurations in Oracle Service Bus environment. You can perform system monitoring without initiating a session. However, you must first create an session in the Oracle Service Bus 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 effective only when the current session is activated. [Figure 3-2](#) illustrates the Change Center pane available on Oracle Service Bus 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 and enter comments for those changes. 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 Oracle Service Bus system environment, system configuration entities are grouped into user-defined projects. On Oracle Service Bus Console, the Project Explorer page displays a default project and the various user defined projects under the Projects folder. This page allows you to perform all project configuration tasks. Each project is represented by a project folder on the console. You can add new project folders and navigate the project trees. You can collapse and expand folders as required by clicking them.

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

This section describes the 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

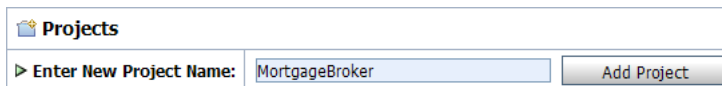
### To Start a Session and Create a Project

1. In the Oracle Service Bus Console Change Center click **Create** to create a new session.
2. In the Oracle Service Bus 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
<button>Add Project</button>	

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**, enter a description of the changes you just made, and click **Submit** to save the project directory structure.

## Create the WSDL Resources

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

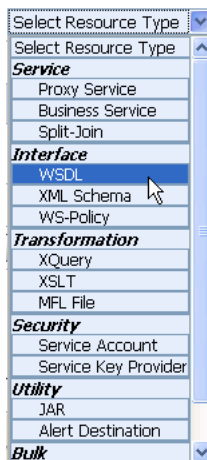
### To Create the normalLoan WSDL Resource

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 Oracle Service Bus Console Change Center click **Create** to create a new session.
2. In Oracle Service Bus 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\osb_10.3\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**

**Create a New WSDL Resource. (Path - MB)**

<b>Resource Name*</b>	normalLoan
<b>Resource Description</b>	
<b>WSDL*</b>	D:\bea3214\alsb_3.0\samples\servicebu <input type="button" value="Browse..."/>

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 the steps in the [To Import a WSDL](#) section to create a WSDL resource associated with the ManagerApproval Service. To import the ManagerApprovalService WSDL resource, use the configuration parameters listed in the following table.

**Table 3-3 ManagerApproval Service WSDL**

Configuration Parameter	Value
WSDL Resource Name	managerApproval
ManagerApprovalService WSDL	BEA_HOME\osb_10.3\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 Oracle Service Bus 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/ProxyService/)	
General Configuration	
Service Name*	<input type="text" value="LoanGateway1"/>
Description	<div></div>
Service Type*	<div><div><b>Create a New Service</b> <input checked="" type="radio"/> WSDL Web Service <input type="text"/> <input type="button" value="Browse..."/> <input type="text"/> (port or binding) <input type="radio"/> Messaging Service <input type="radio"/> Any SOAP Service <input type="text" value="SOAP 1.1"/> <input type="radio"/> Any XML Service</div><div><b>Create From Existing Service</b> <input type="radio"/> Business Service <input type="text"/> <input type="button" value="Browse..."/> <input type="radio"/> Proxy Service <input type="text"/> <input type="button" value="Browse..."/></div></div>

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

- 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 3-9 LoanGateway1 Proxy Service**

Create a Proxy Service (MortgageBroker/ProxyService/)	
<b>General Configuration</b>	
Service Name*	LoanGateway1
Description	
Service Type*	<div> <b>Create a New Service</b> <div> <input checked="" type="radio"/> WSDL Web Service                             <div>                                 MortgageBroker/WSDL/normalLoan <span>Browse...</span> </div> <div>                                 helloPort (port)                             </div> </div> <div> <input type="radio"/> Messaging Service                             <div>                                 SOAP 1.1                             </div> </div> <div> <input type="radio"/> Any SOAP Service                             <div>                                 SOAP 1.1                             </div> </div> <div> <input type="radio"/> Any XML Service                             <div>                                 SOAP 1.1                             </div> </div> </div> <div> <b>Create From Existing Service</b> <div> <input type="radio"/> Business Service <span>Browse...</span> </div> <div> <input type="radio"/> Proxy Service <span>Browse...</span> </div> </div>
<div> <span>Next &gt;&gt;</span> <span>Last &gt;&gt;</span> <span>Cancel</span> </div>	

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

9. Accept the default protocol, http.

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

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

11. Accept the default for the Get All Headers option (**No**), and click **Next**. The Create a Proxy Service-HTTP Transport Configuration page is displayed.

12. Click **Next**.

13. On the Operation Selection Configuration page, accept the default selection algorithm (SOAP Body Type). Click **Next**.

14. In the Message Content Handling page, click **Next**.

15. The Create a Proxy Service – Summary page is displayed. 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.

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

- 16. After reviewing the proxy service configuration settings, click **Save** to register the service.  
The LoanGateway1 proxy service is displayed in the Resources pane in the ProxyService folder of the MortgageBroker project.
- 17. Click the **LoanGateway1** link in the Resources pane of MortgageBroker/ProxyServices to go the View a Proxy Service page.

Figure 3-10 Proxy Services in Resource Pane

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

- 18. The View a Proxy Service page is displayed.  
The Operational Settings tab, as shown in [Figure 3-11](#), has options to enable monitoring for the proxy service.

**Figure 3-11 Monitoring Configuration of Proxy Service**

View a Proxy Service (MortgageBroker/ProxyService/LoanGateway1)

Last Modified By	weblogic	<b>Description</b> - no description -
Last Modified On	10/1/08 12:28 PM	
References	1 Ref(s)	
Referenced By	0	

Configuration Details | Operational Settings | SLA Alert Rules | Policies | Security

**General Configuration**

State ☒ Enabled

**Monitoring**

Monitoring ☐ Enable Pipeline Monitoring at Pipeline level or above

Aggregation Interval 0 hours 10 mins

SLA Alerts ☒ Enable Alerting at Normal level or above

Pipeline Alerts ☒ Enable Alerting at Normal level or above

Reports ☒ Enabled

Logs ☒ Enable Logging at Debug level or above

**Tracing**

Execution Tracing ☐ Enabled

Message Tracing ☐ Enabled
 

Detail Level Terse

Payload Tracing Limit 8 Kilobytes

Default Encoding

Back | Update | Reset

To enable monitoring:

- Click the **Operational Settings** tab.
- Select the **Enable Pipeline Monitoring** check box for the Monitoring setting.
- 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 Oracle Service Bus 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 Oracle Service Bus Dashboard and monitoring services, see: [Monitoring](#) in Using the Oracle Service Bus Console and Oracle Service Bus Operations Guide.

- Click **Update**.

- e. **Activate** the session.

This step completes the configuration of the LoanGateway1 proxy service in Oracle Service Bus Console. 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 Oracle Service Bus Console **Change Center** click **Create** to create a new session
2. In the Project Explorer, select the BusinesService 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 3-12](#).

Figure 3-12 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 Browse...  
helloPort (port)

☐ Transport Typed Service

☐ Messaging Service

☐ Any SOAP Service SOAP 1.1

☐ Any XML Service

**Create From Existing Service**

☐ Business Service Browse...

☐ Proxy Service Browse...

Next >> Last >> Cancel

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 3-13](#).

**Figure 3-13 Transport Configuration of a Business Service**

**Create a Business Service (default/normal)**

**Transport Configuration**

<b>Protocol*</b>	http				
<b>Load Balancing Algorithm</b>	none				
<b>Endpoint URI*</b>	<p>Format: http://host:port/someService</p> <p>http:// <input type="text"/> <input type="button" value="Add"/></p> <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="Up"/> <input type="button" value="Down"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> </td> </tr> </tbody> </table>	EXISTING URIS	OPTIONS	http://localhost:7001/njws_basic_ejb/NormalSimpleBean	<input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
EXISTING URIS	OPTIONS				
http://localhost:7001/njws_basic_ejb/NormalSimpleBean	<input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>				
<b>Retry Count</b>	0				
<b>Retry Iteration Interval</b>	30				
<b>Retry Application Errors</b>	<input checked="" type="radio"/> Yes <input type="radio"/> No				

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

9. 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 Oracle Service Bus server is running.

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

10. Accept the default settings for the remaining options on the page and click **Next**.
11. In the Create a Business Service-HTTP Transport Configuration page, accept the default settings on this page, then click **Next**.
12. In the Create a Business Service-SOAP Binding Configuration page, accept the default settings and click **Next**.
13. In the Message Content Handling page, click **Next**.

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

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

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

To enable monitoring:

- a. Click the **Operational Settings** tab.
- b. Select the **Enabled** check box for the Monitoring field.
- 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 Oracle Service Bus 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 Oracle Service Bus Dashboard and monitoring services, see [Monitoring](#) in Using the Oracle Service Bus Console and [Monitoring Oracle Service Bus at Runtime](#) Oracle Service Bus Operations Guide.

Click **Update**.

- d. **Activate** the session.

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

### Create the ManagerLoanReview Business Service

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

Follow the same 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	managerApproval helloPort
Load Balancing Algorithm	none
EndpointURI	http://<host:port>/mjws_basic_ejb/ManagerSimpleBean

## 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

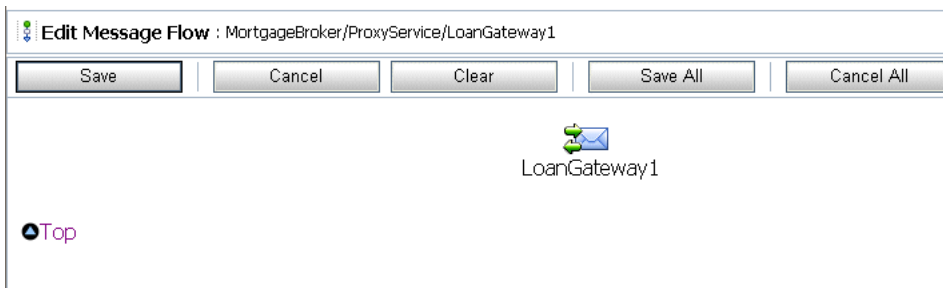
Oracle Service Bus Message Flows define the implementation of proxy services. Message flows can include zero or more pipeline pairs: request and response pipelines for the proxy service (or for the operations on the service) and error handler pipelines that can be defined for stages, pipelines, and proxy services. Pipelines can include one or more stages, which in turn include actions.

### Configure the Routing Behavior of the Message Flow

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

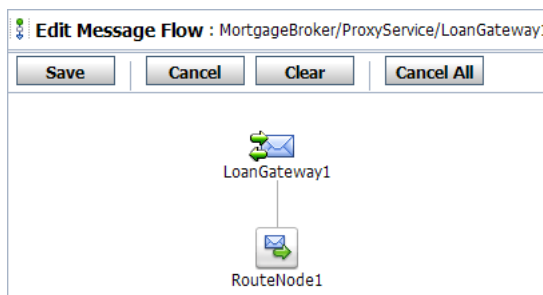
3. In the Actions column associated with the LoanGateway1 proxy service, click the **Edit Message Flow** icon. The Edit Message Flow page for the proxy service LoanGateway1 is displayed as shown in [Figure 3-14](#).

**Figure 3-14 Edit Message Flow for LoanGateway1 Proxy Service**



4. Click **LoanGateway1**, and select **Add Route** from the short-cut menu options.  
RouteNode1 is added in the configuration page as shown in [Figure 3-15](#).

**Figure 3-15 Add a Route Node**

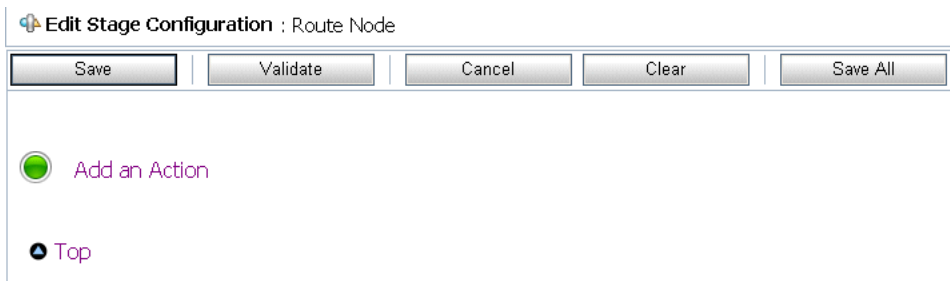


5. Click **RouteNode1** and select **Edit > Route** from the short-cut menu options.

The Edit Stage Configuration: Route Node page is displayed as shown in [Figure 3-16](#). This page contains a single Add an Action link.

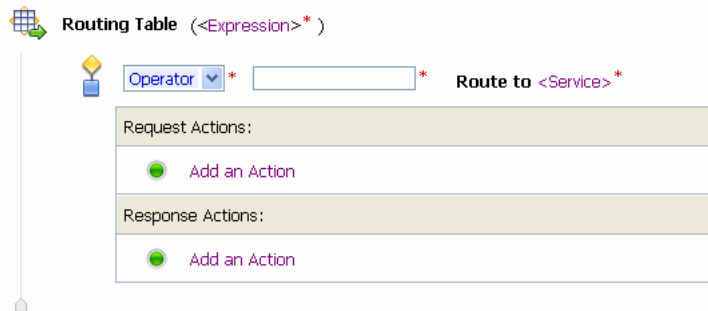
**Note:** A stage is an element of a pipeline and also a container for actions defined in a pipeline. Actions are the elements of a pipeline stage that define the handling of messages as they flow through a proxy service at run time.



**Figure 3-16 Add an Action Link**

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

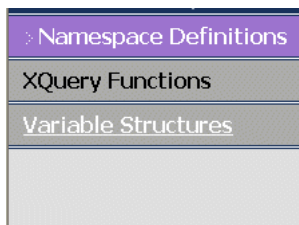
The Edit Stage Configuration page changes to display routing table configuration page.

**Figure 3-17 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.

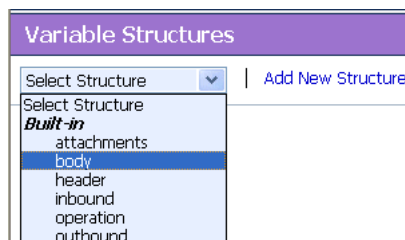
7. In the Routing table, click **<Expression>**. The XQuery Expression Editor is displayed.
8. In the left pane, click **Variable Structures**.

**Figure 3-18 Namespace Definitions Pane**



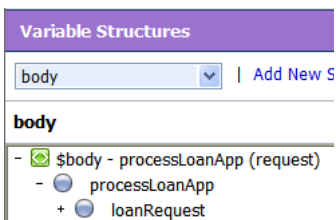
9. The Variable Structures pane is displayed. Select **body** from the Select Structure drop-down list in the Variable Structures pane.

**Figure 3-19 Variable Structures Pane**

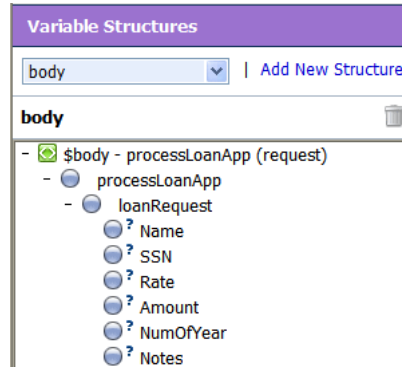


10. A structural representation of the body element is displayed in the Variable Structures pane.
11. Click + to expand the processLoanApp element, within the \$body - processLoanApp (request) element.

**Figure 3-20 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.

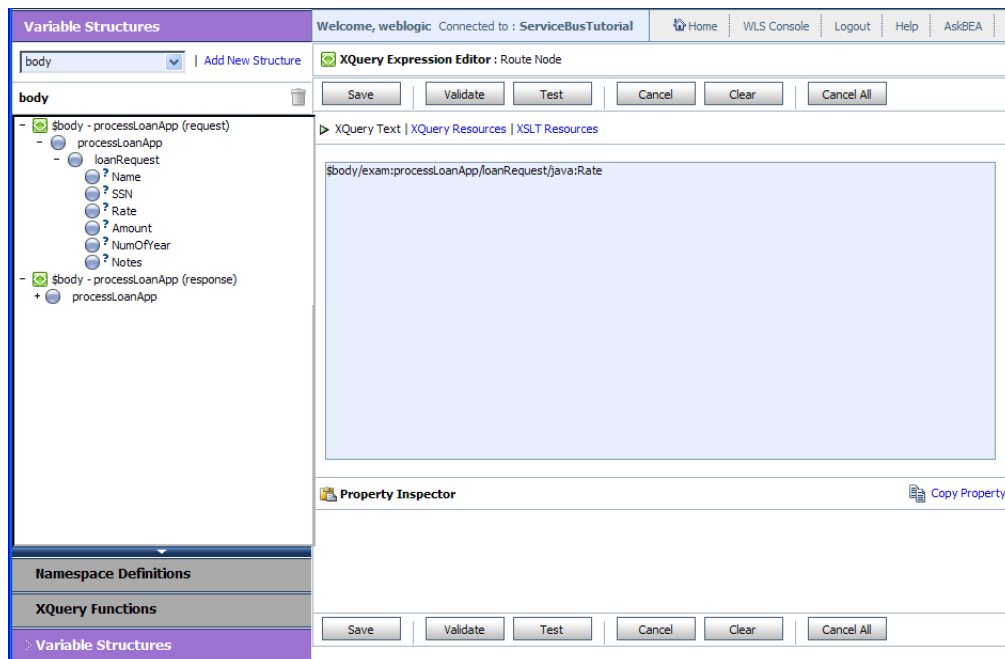
**Figure 3-21** loanRequest Element

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
```

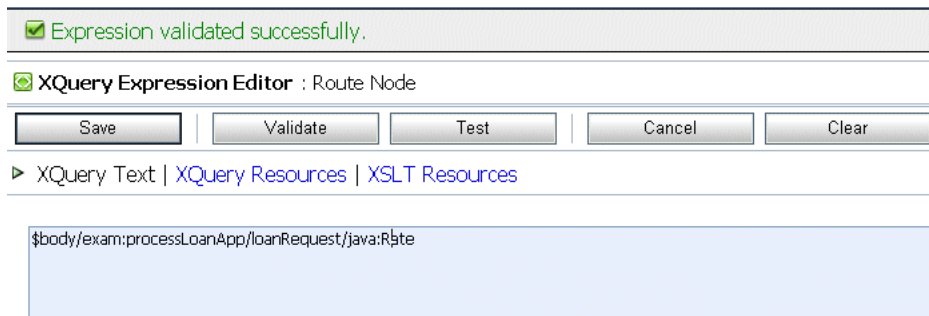
**Note:** The drag-and-drop functionality works only in Internet Explorer (IE) browsers. If you are using a browser other than IE, select the Rate element in the Variable Structures pane. The expression is displayed in the Property Inspector palette. Copy the expression in the palette and paste it in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

**Figure 3-22 XQuery Expression Editor**



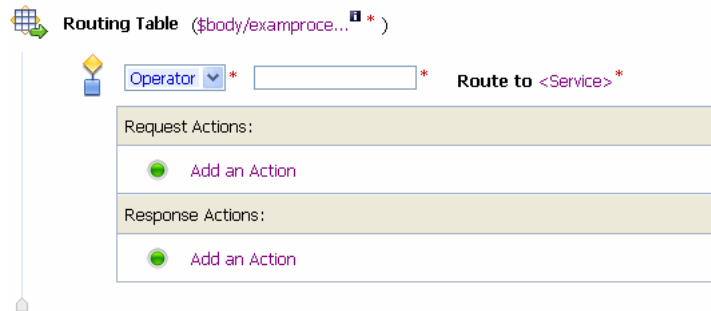
14. Click **Validate** to validate the XQuery. It is a good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button.

**Figure 3-23 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.

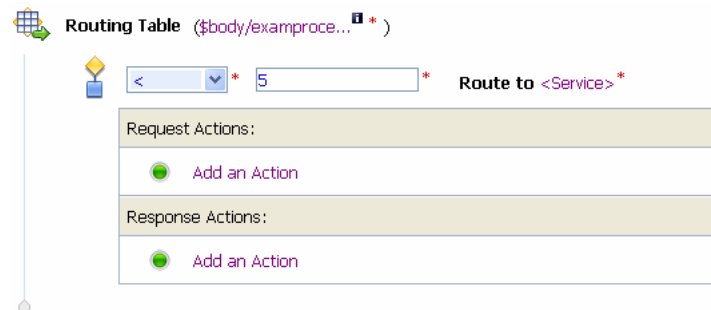
**Figure 3-24 Routing Table View - Expression**



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 3-25 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.
- Select the **ManagerLoanReview** business service.
  - 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 (below the Routing Table label), then select Insert Default Case. The else condition (default case) is added to the routing table.

**Figure 3-26 Routing Table Default Condition**

The screenshot shows the 'Routing Table' configuration page. At the top, there's a title bar with a grid icon and the text 'Routing Table (\$body/exam:proc...\*)'. Below this, there's a condition editor showing '<' followed by a dropdown arrow, then '5', and an asterisk. To the right of this is the text 'Route to ManagerLoanReview\*' and a dropdown menu showing 'processLoanApp'. Below the condition editor, there's a checkbox labeled 'Use inbound operation for outbound' which is currently unchecked. There are two main sections for actions: 'Request Actions:' and 'Response Actions:'. Each section has a green circular icon and the text 'Add an Action'. Below these sections, there's a 'Default' section with a blue folder icon and the text 'Default: Route to <Service>\*'. This section also has 'Request Actions:' and 'Response Actions:' sub-sections, each with a green circular icon and the text 'Add an Action'.

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 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 3-27](#).

**Figure 3-27 Routing Behavior of the Message Flow**

**Routing Table** (`{body/exam:proc...}`)

☐ Use inbound operation for outbound

Request Actions:

Response Actions:

☐ Use inbound operation for outbound

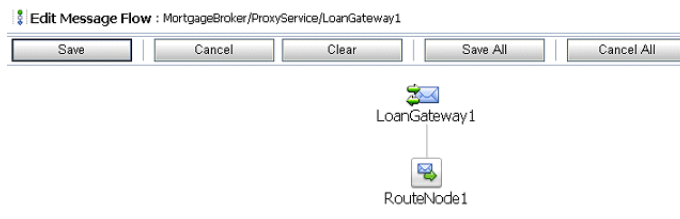
Request Actions:

Response Actions:

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

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

**Figure 3-28 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 Oracle Service Bus 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 WebLogic Server in ServiceBusTutorial Domain.
2. In Oracle Service Bus Console, make sure the current session is activated.
3. Click on **Resource Browser > Proxy Services** in the left panel.
4. Click on **Launch Test Console** icon in Actions column for LoanGateway1 to launch the test console (see [Figure 3-29](#)).



Figure 3-29 Test Console for LoanGateway1

Proxy Service Testing - loanGateway1 [Help](#)

Available Operations: processLoanApp

Execute Reset Close

**Test Configuration**

Direct Call: ☒

Include Tracing: ☒

**Request Document**

Form XML

**SOAP Header:**

```
<soap:Header xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
</soap:Header>
```

**loanRequest:**

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

**Transport**

Execute Reset Close

- 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 3-30](#) is obtained.

**Figure 3-30 Response from the ManagerLoanApplicationReview Service**



The Invocation Trace section of the test console indicates that the proxy service routed the request to ManagerLoanReview business service because the interest rate requested is 4.1 (less than 5).

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

1. Start WebLogic Server in ServiceBusTutorial Domain
2. In Oracle Service Bus Console, make sure the session is activated.
3. Click on **Resource Browser > Proxy Services** in the left panel.
4. Click on **Launch Test Console** icon in Actions column for LoanGateway1 to launch the test console.
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 3-31](#) is obtained.

**Figure 3-31 Response From NormalLoanProcessor**



The Invocation Trace section of the test console indicates that the proxy service routed the request to NormalLoan 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. Oracle Service Bus can be configured to route and transforms messages when necessary, based on specific proxy service configurations.

This section includes the following topics:

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

## Prerequisites

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 Oracle Service Bus Console. Using the Oracle Service Bus 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.

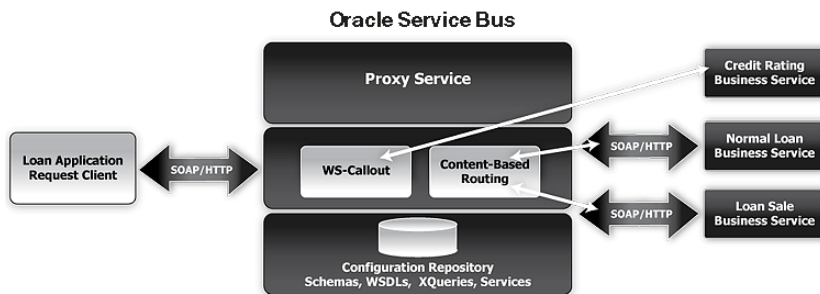
## Definition of the Scenario

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

The credit rating information is added to the loan application. The application is then forwarded to the secondary mortgage company Web service to be processed. Loan applications with a principal request equal to or less than US \$25 million are routed to a different business service for processing. The target business services respond indicating whether the loan application is approved or rejected.

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

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



## Overview of the Run-time Process in Oracle Service Bus

A primary mortgage company receives a loan application. It is routed through the Oracle Service Bus proxy service, LoanGateway2, to determine the target business service to process the application. If the loan amount is greater than US \$25 million, the application is routed to the LoanSaleProcessor business service. If the amount is less than or equal to US \$25 million, the application is routed to the NormalLoan business service.

When the loan amount is more than US \$25 million, the request pipeline makes a service callout to the CreditRating business service and receives the credit rating of the applicant using the \$creditRating variable. To fulfill the interface requirements of the secondary loan company service, the message body is transformed by adding the credit rating details. The transformed

message (\$body) is routed to a business service that handles applications for large loan amounts. The service returns a response similar to the following:

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

## Required Resources

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

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

Resource Name	Description
creditRating, loanSale	These are the WSDL resources.
NormalLoan, LoanSaleProcessor, CreditRating	These are the external business services used by Oracle Service Bus.
LoanGateway2	This is the Oracle Service Bus 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 Oracle Service Bus 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 Oracle Service Bus Console Change Center click **Create** to create a new session.
2. In the Oracle Service Bus 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:
  - BusinessService
  - ProxyService
  - 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\osb_10.3\samples\servicebus\examples\src\examples\webservices\jws_basic\large\LargeLoanPurchasingService.wsdl
creditRating	BEA_HOME\osb_10.3\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**

Resources				
▶ Create Resource: <span>Select Resource Type</span> ▼				
			Items 1-4 of 4	
<input type="checkbox"/> Name ▲	Resource Type	Actions	Options	
<input type="checkbox"/> creditRating	WSDL			
<input type="checkbox"/> loanSale	WSDL			
<input type="checkbox"/> managerApproval	WSDL			
<input type="checkbox"/> normalLoan	WSDL			
			Items 1-4 of 4	

## 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, **LoanGateway2**, 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**

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

After you complete this task, a summary of the configuration settings for the proxy service is displayed.

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 4-3](#).

**Figure 4-3 Proxy Service Resources**

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

## Create a Business Service

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

- **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** – 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** – 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**

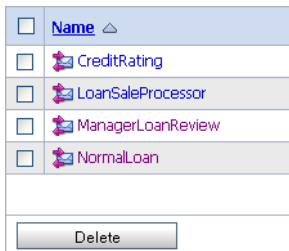
Business Service Name	LoanSaleProcessor
Service Type	Select the <b>WSDL Web Service</b> as the Service Type option. Select the <b>loanSale</b> WSDL in the Select a WSDL page. Select the <b>helloPort</b> port in the Select a WSDL Definition page.
Load Balancing Algorithm	none
Endpoint URI	Ensure that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

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

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

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 4-4](#).

**Figure 4-4 Business Service Resources**



### 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 Oracle Service Bus 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

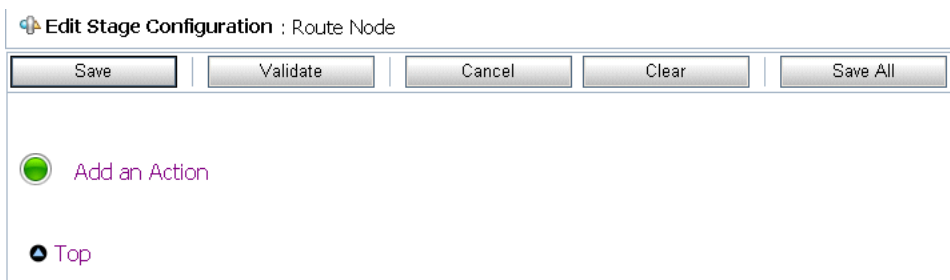
1. In the Oracle Service Bus Console navigation pane, select **Resource Browser**.  
The Resource Browser pane is opened in the navigation pane and the Summary of Proxy Services page is displayed in the console.
2. In the Actions column associated with the LoanGateway2 proxy service, click the **Edit Message Flow** icon.
3. Click **LoanGateway2**, then select **Add Route** from the short-cut menu. In the configuration dialog, RouteNode1 is added.

**Figure 4-5 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.

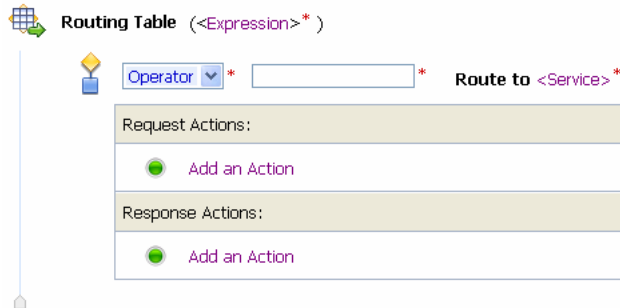
**Figure 4-6 Add an Action Link**



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

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

**Figure 4-7 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.
2. In the Namespace Definitions pane, click **Variable Structures**.
3. Select **body** from the Select Structure drop-down list in the Variable Structures pane.
4. 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.

5. 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.
6. 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. Copy the expression in the palette and paste it in the XQuery Expression text box.

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

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

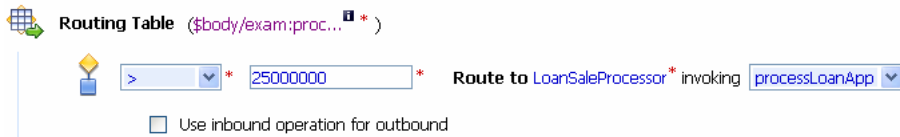
The routing table now contains an expression that determines the routing behavior. If the value in the Amount field is greater than US \$25 million, then messages are routed according to the routing table configuration.

10. 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**.
11. 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 4-8](#).

**Figure 4-8 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**.

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

2. Click **<XPath>**. The XPath Expression editor is displayed.

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

```
./exam:processLoanApp/loanRequest/@xsi:type
```

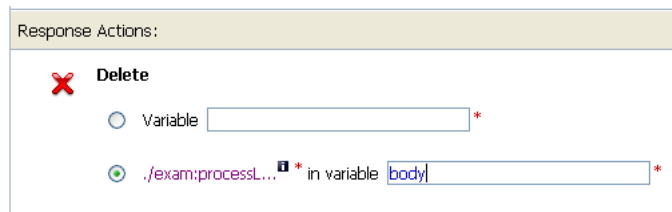
- b. Click **Validate**, then **Save**. You will return to the Edit Stage Configuration page.

3. Enter **body** in the variable text field of the **<XPath>**.

The configuration for the Delete action is therefore:

Delete `./exam:processLoanApp/loanRequest/@xsi:type` in variable `body` as shown in [Figure 4-9](#).

**Figure 4-9 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. In the Request Actions pane of the Routing Table, click the **Delete** icon, then select **Add an Action > Message Processing > Assign**.

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

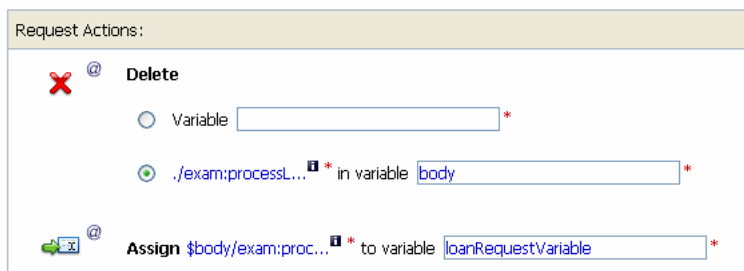
2. Click the **<Expression>** link. The XQuery Expression editor is displayed.

3. Navigate to the **Variables Structures** pane. Select **body** from the Variables Structures drop-down list.

4. Expand the request 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
```
5. 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.
6. Enter **loanRequestVariable** in the variable text box as shown in [Figure 4-10](#).

**Figure 4-10** 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. In the Request Actions pane of the Routing Table, click the **Assign** icon, then **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 **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.

6. 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 4-11](#).

**Figure 4-11 LoanGateway2 Routing Request Rename Action**

The screenshot shows the configuration for a 'Rename' action in the Oracle Service Bus. It includes three main sections: 'Delete', 'Assign', and 'Rename'.

- Delete:** A radio button is selected for 'namespace'. The XPath expression is `./exam:processL...` in variable `body`.
- Assign:** A radio button is selected for 'Assign'. The expression is `$body/exam:proc...` assigned to the variable `loanRequestVariable`.
- Rename:** A radio button is selected for 'Rename'. The XPath expression is `./java:*` in variable `loanRequestVariable`. The 'namespace' option is selected, and the new namespace is `java:credit.client`.

## 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, then **Add an Action > Communication > Service Callout**.

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

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

The service callout action is displayed.

3. From the Operation drop-down list, select **processLoanApp**. The fields that allow you to configure the request and response parameters for the service callout are displayed as shown in [Figure 4-12](#).

**Figure 4-12 Service Callout Action - Configured Operation**

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

☐ Configure Soap Body
 ☒ Configure Payload Parameters

Request Parameters:

loanRequest \*

Response Parameters:

return \*

SOAP Request Header:

SOAP Response Header:

Request Actions:

☒ Add an Action

Response Actions:

☒ Add an Action

- a. In the service callout Request Parameters section, set the **loanRequest** parameter to **loanRequestVariable** (the message context variable you defined in the preceding section).
- b. In the service callout Response Parameters section, set the **return** parameter name to **creditRating** as shown in [Figure 4-13](#).

**Figure 4-13 Service Callout Action - Configured Parameters**

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

☐ Configure Soap Body
 ☒ Configure Payload Parameters

Request Parameters:

loanRequestVariable \*

Response Parameters:

creditRating \*

Request Actions:

☒ Add an Action

Response Actions:

☒ Add an Action

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 an 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 4-14](#).

**Figure 4-14 Message Namespace Rename Action**

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

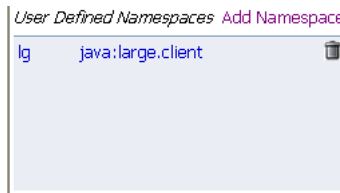
### Insert a New Element in the Message

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 4-15](#).

**Figure 4-15 User Defined Namespace**

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

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

where:

- **lg:** is 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**.
6. In the Insert action, select **after** from the drop-down list.
7. Click the **<XPath>** link. The XPath Expression Editor 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 request element list and drag the `Notes` element 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. 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
```

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 4-16](#).

**Figure 4-16 Insert New Element into Message**



11. Click **Save** in the Edit Stage Configuration page.

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 4-17 Configuration of Outbound Message**

The screenshot shows the 'Routing Table' configuration for a message with body '\$body/exam:proc...'. The configuration includes a 'Route to loanSubProcessor' action, followed by a 'Delete' action, an 'Assign' action to set 'loanrequestVariable', a 'Rename' action for the namespace, a 'Service Callout to CreditRatingService' with request and response parameters, and another 'Rename' action for the final body. An 'Insert' action is also present at the bottom.

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

Route to `loanSubProcessor` invoking `processLoanApp`

☐ Use inbound operation for outbound

**Request Actions:**

- Delete**
  - Variable: `body`
- Assign** `$body/exam:proc...` to variable `loanrequestVariable`
- Rename** `./java:` in variable `loanrequestVariable` to
  - localname: `loanrequestVariable`
  - namespace: `java:credit.client`
- Service Callout to CreditRatingService** invoking `processLoanApp`
  - ☐ Configure Soap Body ☒ Configure Payload Parameters
  - Request Parameters:**
    - `loanrequest`: `loanrequestVariable`
  - Response Parameters:**
    - `return`: `creditRating`
  - SOAP Request Header:**
  - SOAP Response Header:**
  - Request Actions:**
    - Add an Action
  - Response Actions:**
    - Add an Action
- Rename** `./java:` in variable `body` to
  - localname: `body`
  - namespace: `java:target.client`
- Insert** `<lg:CreditRat...` `after` `./exam:process...` in variable `body`

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 Oracle Service Bus Console. (Click **RouteNode1 > Edit Route** in the proxy service's Edit Message Flow page).
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>** option. 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**.
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 4-18](#).

**Figure 4-18 Response Delete Action**

Response Actions:

**Delete**

☐ Variable  \*

☒ ./exam:processL... \* in variable  body \*

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

### Add a Rename Action

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

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

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 Oracle Service Bus replaces all namespaces with the prefix lg: in the response message with java:normal.client as shown in [Figure 4-19](#).

**Figure 4-19 Add Rename Action**

**Rename**  **In variable**  **\* to**

☐ 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 4-20](#).

**Figure 4-20 Configuration of Response Actions of Proxy Service**

**Response Actions:**

☒ **Delete**

☐ Variable  \*

☒  **In variable**  \*

**Rename**  **In variable**  **\* to**

☐ 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

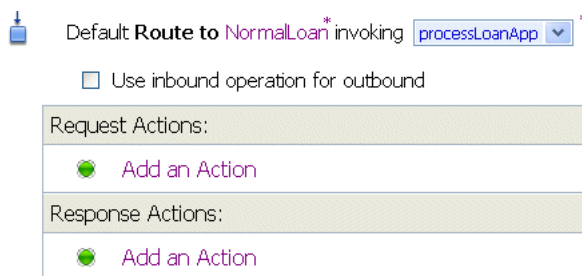
This section describes how you can configure a default routing case for the routing table of the LoanGateway2 proxy service. At run time, if none of the conditions configured in the Routing Table are met, Oracle Service Bus routes messages according to the configuration of this default case.

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.
  - 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 4-21](#).


**Figure 4-21 Add Default Routing Case**




Default **Route to NormalLoan\*** invoking **processLoanApp** \*

☐ Use inbound operation for outbound

Request Actions:

 Add an Action

Response Actions:

 Add an Action

4. Click **Save** to save the route node configuration and return to the Edit Message Flow page for the LoanGateway2 proxy service.
5. Click **Save**.
6. Click **Activate**, and then **Submit** to save the session configuration information.

The message flow map of the proxy service is displayed.

## 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 WebLogic Server console. For more information see, [Load the Client Application](#).

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

1. Start WebLogic Server in ServiceBusTutorial Domain.
2. In Oracle Service Bus Console, make sure the session is activated.
3. Click on **Resource Browser > Proxy Services** in the left panel.
4. Click the **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see [Figure 3-29](#)).
5. Replace the default payload in the 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 4-22](#).

**Figure 4-22 Response From LoanSaleProcessor Business Service**



The Invocation Trace indicates that the proxy service LoanGateway2 routes the application to LoanSaleProcessor because the amount of the loan requested is greater than US \$25 million.

## Test the NormalLoan Business Service

1. Start WebLogic Server in ServiceBusTutorial Domain.
2. In Oracle Service Bus Console activate the current session.
3. Click on **Resource Browser > Proxy Services** in the left panel.
4. Click on **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see [Figure 3-29](#)).
5. Replace the default payload in loanRequest field with the following code:

```

<loanRequest xmlns:java="java:normal.client">

    <!--Optional:-->

```

## Tutorial 2. Transforming a Loan Application

```
<java:Name>Smith</java:Name>
<!--Optional:-->
<java:SSN>1234567</java:SSN>
<!--Optional:-->
<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 4-23](#).

**Figure 4-23 Response from NormalLoanProcessor**



The Invocation Trace indicates that the proxy service LoanGateway2 routes the application to NormalLoanApproval business service 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

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

You can configure message validation logic in the following ways:

- By validating elements selected by an XPath expression against a top level XML schema element or WSDL resource. For more information about the Validate action, see “Adding an Action” in [Proxy Services: Actions](#) in Using the Oracle 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 Oracle 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 Oracle Service Bus resources that implement message validation and error handling, in Oracle Service Bus Console. This tutorial includes the following:

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

## Features Highlighted

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

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

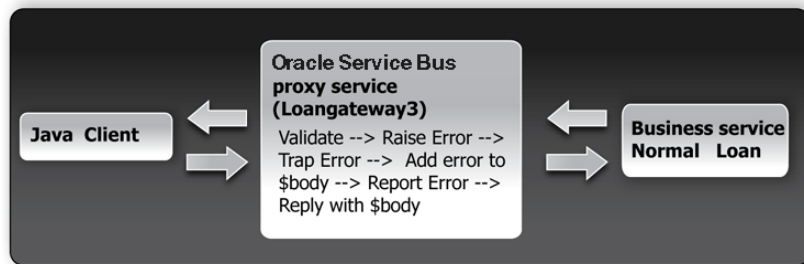
## Definition of the Scenario

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

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

Figure 5-1 summarizes the logical architecture to support this scenario and illustrates how Oracle Service Bus is positioned in the enterprise system environment to mediate the messaging between the client and the business services.

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



## Overview of the Run-Time Process in Oracle Service Bus

In this scenario, you can configure Oracle Service Bus to validate an incoming message in the message flow against a schema definition in a WSDL resource. Every field is checked for valid content. All elements must be populated for the application to be valid. When the value of the loan duration field is not an integer, the validate action raises an error. The context variable, \$fault, is populated with the error details.

The error is trapped by a stage error handler which replaces the body context variable <Notes> field value with the text of the \$fault <reason> element. A Report action then saves the error code and the message body, for later viewing and searching in the console. The Reply action returns the body context variable to the client.

In this scenario, the message routing is as follows:

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

# Required Resources

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

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

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

# Tasks in This Tutorial

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

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

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

# Prepare Your Environment

Ensure that Oracle Service Bus 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 Oracle Service Bus 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:
  - BusinessService
  - ProxyService
  - 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 **LoanGateway3** 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 <b>normalLoan</b> WSDL, <b>helloPort</b>
Endpoint URI	<b>/loan/LoanGateway3</b>

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, review the summary of configuration settings before registering the proxy service, and click **Save**.

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

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

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

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

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

## Create a Pipeline Pair

1. In the Oracle Service Bus 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.

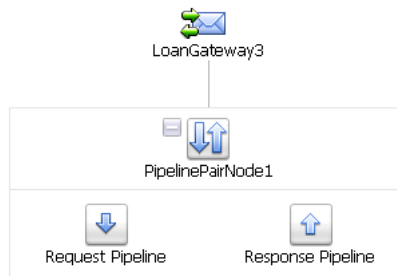


The Edit Message Flow page for the proxy service LoanGateway3 is displayed.

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

The PipelinePairNode1 is created and placed in the message flow. Request and response pipelines are displayed for this node as shown in [Figure 5-2](#).

**Figure 5-2 Add Pipeline Pair**



## Configure the Request Actions for the LoanGateway3 Proxy Service

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

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

## Configure an Action to Perform Message Validation

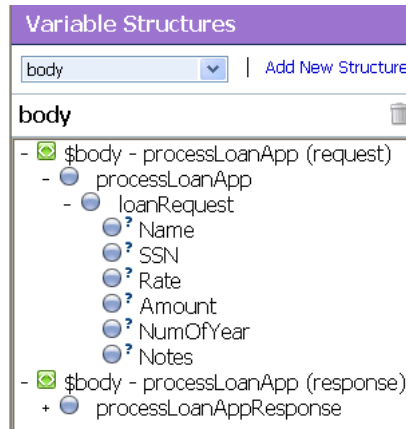
1. To edit the stage, click **validate loan application**, then select **Edit Stage**. The Edit Stage Configuration page is displayed.
2. Select **Add an Action**, then select **Message Processing > Validate** from the drop-down list. The Validate action is added to the stage.
3. In the Validate statement, click the **<XPath>** link to edit the XPath expression. The XPath Expression Editor is displayed.
4. Click **Variable Structures**. The Variable Structures pane is displayed.
5. In the Variables Structure pane, from the Select Structure drop-down list, select **body**.

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

- a. Expand the processLoanApp request element to expand the element.
- b. Click + to expand the loanRequest element.

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

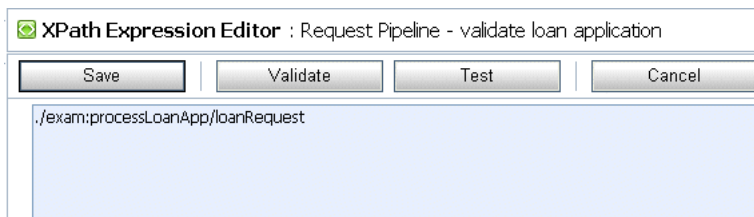
**Figure 5-3 Variable Structures Pane**



- 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 5-4 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. Copy the expression in the palette and paste it in the XQuery Expression text box.

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

- The Edit Stage Configuration page displays <XPath> which is replaced by the XQuery expression.
7. In the **variable** text field, enter **body**.
  8. 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.
  9. 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**.
    - b. Click **Submit** to complete the selection.
  10. Select the **Raise Error on validation failure** option. The Validate action will be displayed as shown in [Figure 5-5](#).

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



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

**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

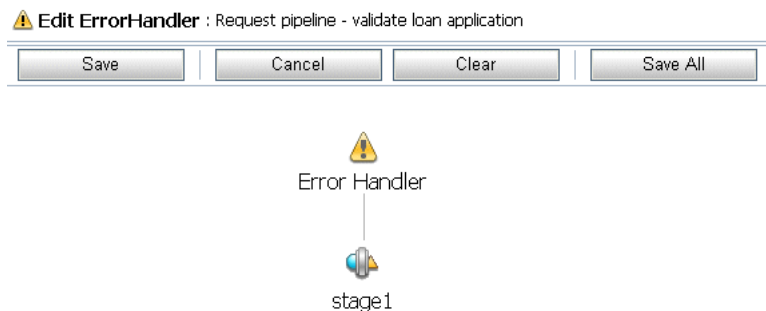
You must perform the following tasks to configure the Stage Error Handler:

- Add Stage Error Handler Element
- Add an If...Then... action to the Stage Error Handler
- Add a Replace Action to the Stage Error Handler
- Add a Report Action to Report the Error
- Add an Else Action to the Stage Error Handler
- Add a Replace Action Within the Else Action
- Add a Log Action Within the Else Action
- Add a Reply Action to Reply with a Failure Code

### Add Stage Error Handler Element

1. Click the **validate loan application** stage, 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 5-6 Add Stage Error Handler**



3. Click **stage1** and select **Edit Name and Comments**.
4. Change the stage name to **Reply**, then click **Save**.  
The Edit Stage Configuration page is displayed.
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 > Flow Control > If... Then...**  
 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.
  - 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 5-7](#):

**Figure 5-7 Conditions in the Expression Builder**

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

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

**Figure 5-8 Expressions Textbox**

Expressions	
↑ ↓ ✎ 🗑️	\$fault/ctx:errorCode = "BEA-382505"

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

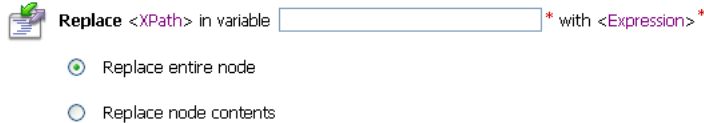
## Add a Replace Action to the Stage Error Handler

1. To add a Replace, click **Add an Action > Message Processing > Replace** 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.

The replace statement is displayed as shown in [Figure 5-9](#).

**Figure 5-9 Replace Statement**



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

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

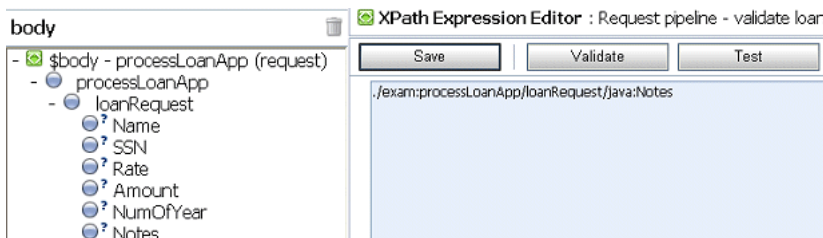
A structural representation of the body element is displayed in the Variable Structures pane.

- a. Expand the processLoanApp request elements.
- b. Select the 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
```

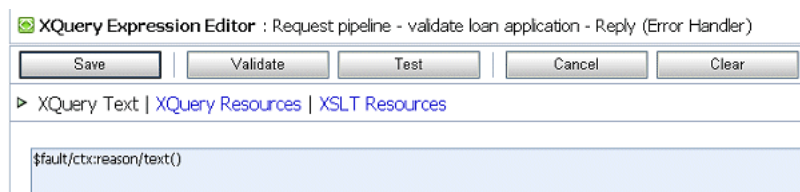
In browsers other than Internet Explorer, copy and paste the expression from the Property Inspector pane.

**Figure 5-10 Replace Statement XPath Expression**



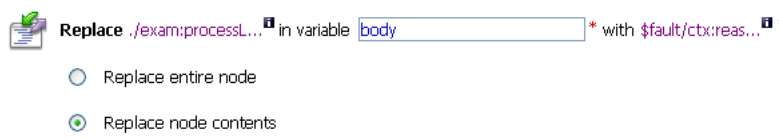
4. Click **Validate**, then **Save**.
5. In the **variable** text field, enter **body**.
6. Click the **<Expression>** link to display the XQuery Expression Editor page.
7. In the XQuery Text pane, enter: `$fault/ctx:reason/text()`

**Figure 5-11 Replace Statement XQuery Expression**



8. Click **Validate**, then **Save**.
9. Select **Replace node contents** option. You need not replace the entire node.  
The Replace action is displayed as shown in [Figure 5-12](#).

**Figure 5-12 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.
2. In the Report statement, click the **<Expression>** link. The XQuery Expression Editor page is displayed.
3. In the Variables Structures pane, select **body** from the Select Structure drop-down list.
  - a. Select the icon associated with the **\$body - ProcessLoanApp (request)** element and drag and drop it on the XQuery Expression text box.

The \$body XPath expression is entered 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. Copy the expression from the Property Inspector pane and paste it in the XPath Expression text box.



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

The Edit Stage Configuration page is displayed.

4. In the key-value pair table, click the **Add a Key** link.
  - a. Enter a Key Name of **errorCode**.

**Figure 5-13 Report Action Expression Key**

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


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






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

**Figure 5-14 Fault Variable Structure**

**Variable Structures**

| [Add New Structure](#)

**fault** 

-  \$fault
  -  errorCode
  -  reason
  -  details
  -  location

- a. Drag and drop the icon associated with the errorCode in the XPath Expression text box.
- Type `./ctx:errorCode` expression in the text box.

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


6. Click **Validate**, then **Save**.

7. In the **variable** text field, enter **fault**.

The **Report** action must be displayed as shown in [Figure 5-15](#).


**Figure 5-15 Configured Report Action**

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


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

The Edit Stage Configuration page is displayed as shown in [Figure 5-16](#).

**Figure 5-16 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

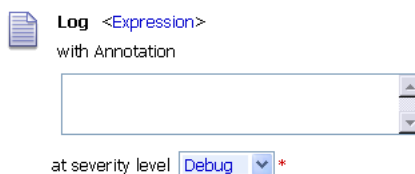
1. Select **Add an Action > Message Processing > Replace**.
2. Execute the steps in the [Add a Replace Action to the Stage Error Handler](#) section.

## Add a Log Action Within the Else Action

1. Click the **Replace** icon within the Else action.
2. Select **Add an Action > Reporting > Log**.

The Log action is displayed.

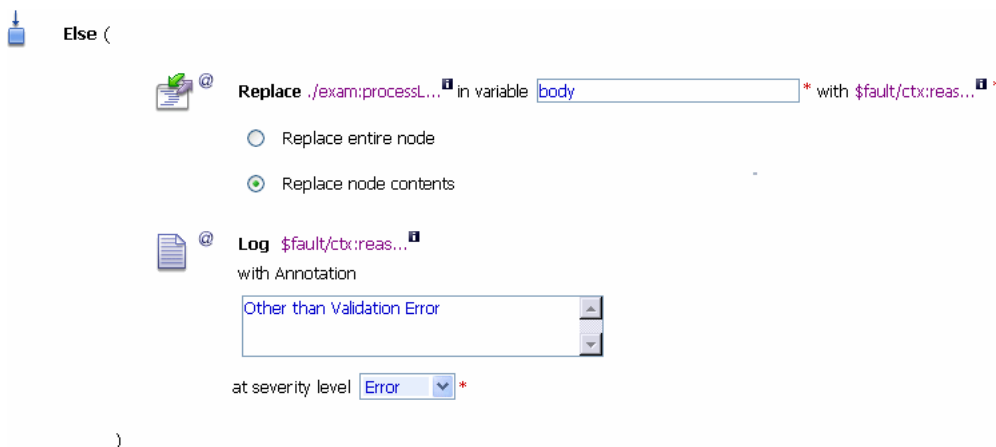
**Figure 5-17 Reporting Log Action**



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 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 5-18](#).

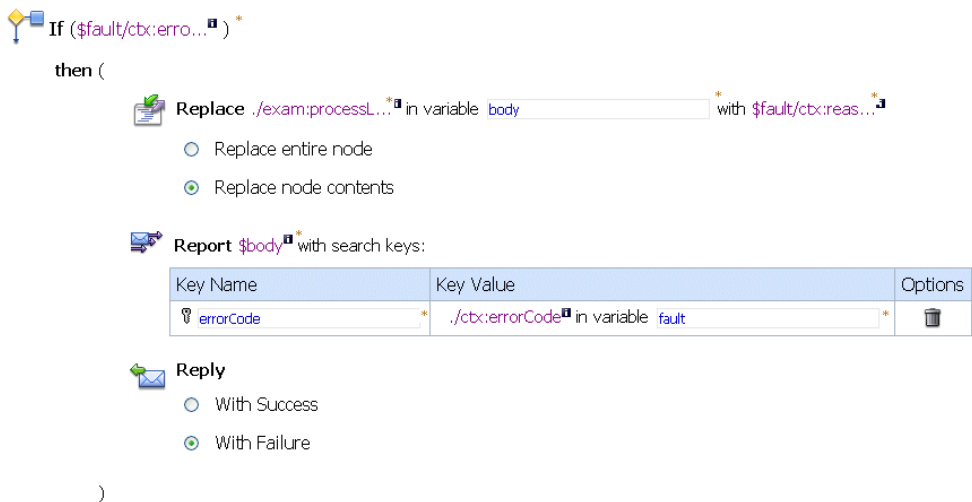
**Figure 5-18 LoanGateway3 Proxy Service Request Else Actions**



## 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 5-19](#).

**Figure 5-19 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**.  
The Edit Message Flow page is displayed.
2. Click **RouteNode1**, then select **Edit Route**. The Edit Stage Configuration page is displayed.
3. Click **Add an Action**, then select **Communication > Routing**.

The Route to action is displayed in the Edit Stage Configuration page as shown [Figure 5-20](#).


**Figure 5-20 Route to Statement**

**Edit Stage Configuration : Route Node**


Save Validate Cancel Clear

 **Route to <Service> \***

Request Actions:

 Add an Action

Response Actions:

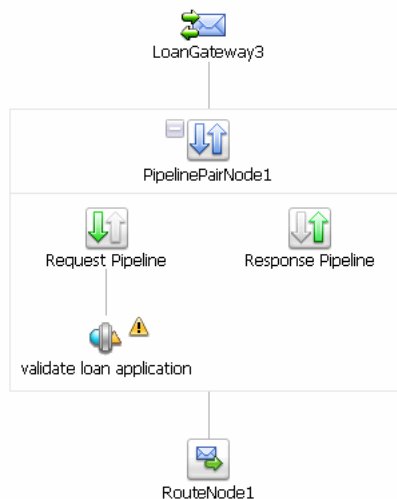
 Add an Action

4. In the route statement, click the **<Service>** link to define the service to which you want to route messages. The Select Service page is displayed.
  - a. Select the **NormalLoan** business service, then **Submit**.

5. In the Operation drop-down list, select **processLoanApp**.
6. 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 5-21](#).

**Figure 5-21 LoanGateway3 Proxy Service Message Flow**



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

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

Now that you have configured Oracle Service Bus to work with the client and the target business services, you must test the configuration. To test the routing configuration, you must change the value of the Loan duration (in years) field in the build.xml file. Then you must test the routing logic by verifying whether the proxy service behavior changes for different loan values.

If the loan value entered is a non integer, 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 WebLogic Server console. For more information see, [Load the Client Application](#).

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

### Scenario One

1. Start WebLogic Server in ServiceBusTutorial Domain.
2. In Oracle Service Bus Console, make sure the session is activated.
3. Click on **Resource Browser > Proxy Services** in the left panel.
4. Click the **Launch Test Console** icon in Actions column for LoanGateway3 to launch the test console.
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</java:Rate>
  <!--Optional:-->
  <java:Amount>9000000000</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 5-22](#).

Figure 5-22 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 WebLogic Server in ServiceBusTutorial Domain
2. In Oracle Service Bus Console, make sure the current session is activated.
3. Click on **Resource Browser > Proxy Services** in the left panel
4. Click on **Launch Test Console** icon in Actions column for LoanGateway1 to launch the test console.
5. Replace the default payload in loanRequest field with the following code:

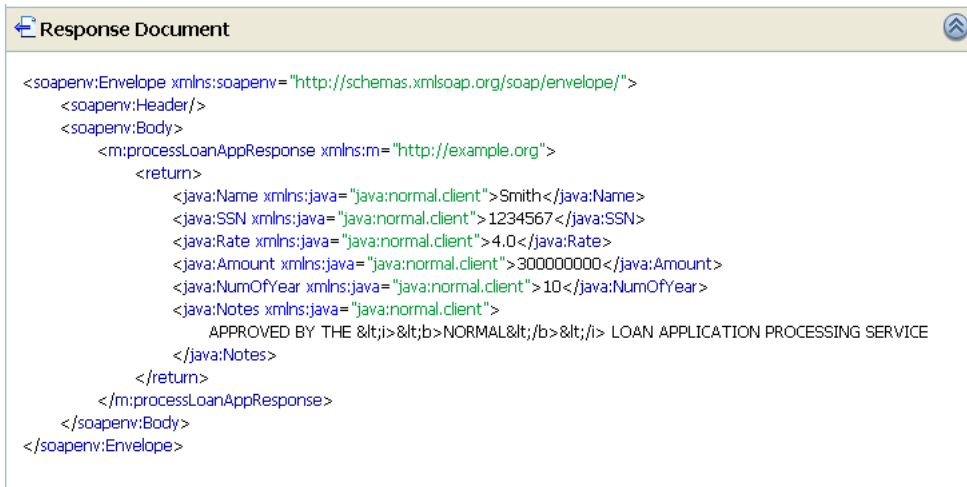
```
<loanRequest xmlns:java="java:normal.client">
  <!--Optional:-->
  <java:Name>Smith</java:Name>
```

## Tutorial 3. Validating a Loan Application

```
<!--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 5-23](#).

**Figure 5-23 Response From Normal Loan Processor**



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 Oracle Service Bus 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

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

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

For more information, see:

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

## Monitoring

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

For more information, see:

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