ORACLE

Oracle Service Bus Examples and Tutorials

April 2010

Contents

1 Oracle Service Bus Examples

This chapter describes the Oracle Service Bus examples that you can install with the product.

The installed examples provide the following:

A preconfigured domain that lets you run the examples and modify input parameters in a Web application.
A set of example source files for reference and for using to reconstruct the examples in the tutorials.

Note: Oracle Service Bus version 11.1.1.3 does not include the example source files. To complete the tutorials, you must download the required files. For download information, see “Before You Begin” in 2 Introduction to the Oracle Service Bus Tutorials.

This chapter includes the following sections:

What are Oracle Service Bus Examples?

The Oracle Service Bus examples provide you with a quick and easy way to experience the run-time capability of using Oracle Service Bus services in your design environment. The examples are preconfigured, user-driven scenarios that use Oracle Service Bus to communicate with business services. The examples are based on typical business scenarios that benefit from using proxy services to communicate between clients and business processes. You can run the examples to see how Oracle Service Bus operates in the run-time environment, or you can use the tutorials build the examples in the design environment to get more in-depth knowledge of how to configure the services.

Examples Description

Oracle Service Bus provides the following examples:

Routing a Loan Application

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

All other loan applications are routed to another business service for processing. The target business service responds, indicating whether the loan application is approved or rejected.

Transforming a Loan Application

A primary mortgage company uses Oracle Service Bus to identify and re-route loan applications that are easy to sell to secondary mortgage companies. A loan application with a principal request of $25,000,000.00 can be sold to a secondary mortgage company. For such a loan application, a Web service lookup is performed to retrieve the customer's credit rating.

The credit rating information is added to the loan application and the application is then forwarded to the secondary mortgage company's Web service to be processed. All other loan applications are routed to another business service for processing. The target business service responds indicating whether the loan application is approved or rejected.

Validating a Loan Application

A primary mortgage company uses Oracle Service Bus to route loan applications to appropriate business services and to validate the loan applications. When an application is invalid (due to missing or incorrect data), an error message is returned to the client and the error is reported in the Oracle Service Bus Console.

A complete application is routed to a selected business service for review. If approved, the business service returns a message indicating whether the loan is accepted or rejected.

Prerequisites to Running the Examples

When installing Oracle WebLogic Server, you must install the Examples, which sets up a sample database.

When installing Oracle Service Bus, you must select the custom installation option and select the Oracle Service Bus Examples to install.

The example and tutorials are based on using an Admin-only domain with Oracle Service Bus running on an Admin server.

Running the Examples

You can run the examples from the Windows Start menu, or by running the domain start script in *MW_HOME*/user_projects/samples/domains/servicebus/.

When the domain starts, the examples page automatically appears in a browser. Click **Load the Examples** to deploy the examples. Click **Reload the Examples** only if you have changed your domain.

What the Examples Illustrate

The Oracle Service Bus examples are based on a loan application request scenario, in which Oracle Service Bus is used to manage message routing in the enterprise environment.

Running the examples introduces the routing, transformation, and validation capabilities of Oracle Service Bus. You can see how the proxy service manages the interaction between the client and the business services to route messages, based on the information from the loan application.

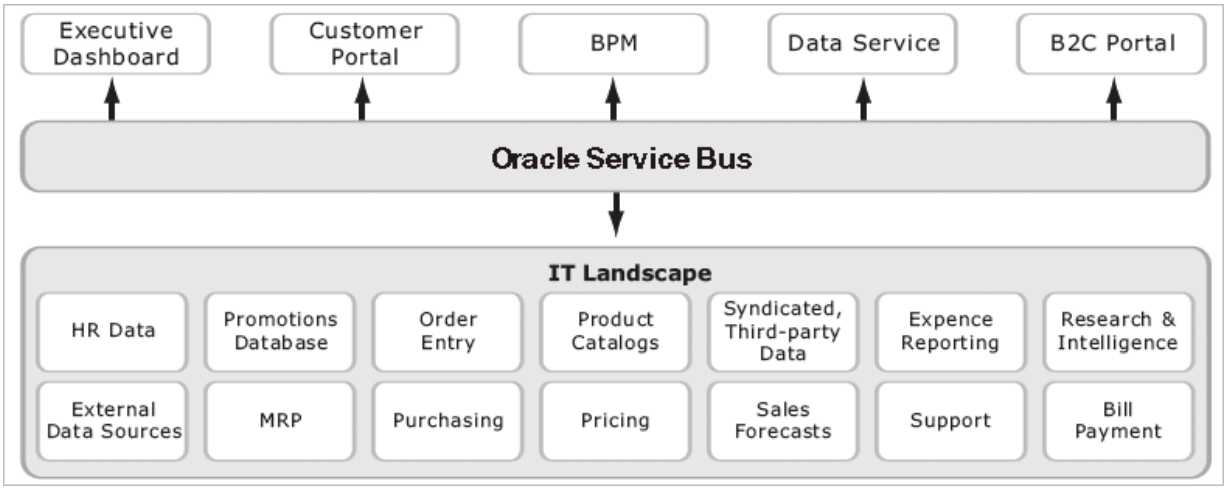
Running an example invokes an application that uses the proxy service to communicate with the client and the business services. When you run an example, you are prompted to change the parameter setting, which in turn, changes the behavior of the proxy services defined in Oracle Service Bus. The fields that you can change for a given example are highlighted on the examples page. Each page also provides basic instructions for executing the examples successfully.

2 Introduction to the Oracle Service Bus Tutorials

Before You Begin – The Oracle Service Bus 11.1.1.3 examples do not include the sample source files needed to complete the tutorial. To complete the tutorials, download the "OSB Tutorial WSDLs" at http://java.net/projects/oraclesoasuite11g/downloads/download/OSB/osb11113_sample_WSDLs.zip. After you download the ZIP file, extract the WSDLs to a temporary location on your computer for use in the tutorials.

Oracle Service Bus is a market-leading enterprise service bus built from the ground up for Service Oriented Architecture (SOA) life cycle 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 2-1 illustrates the role of Oracle Service Bus as a service intermediary in an enterprise SOA architecture.

Figure 2-1 Oracle Service Bus Intermediary



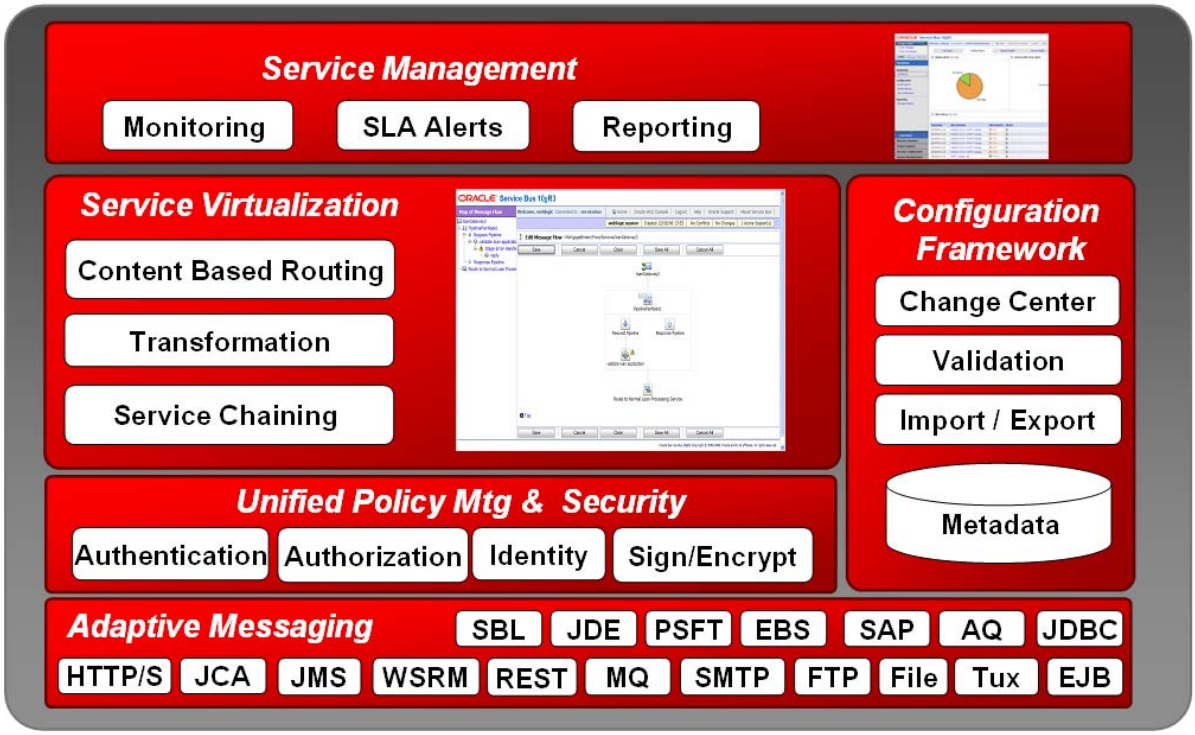
The Oracle Service Bus design-time tooling lets you configure a proxy service to suit your needs and to manage Web services by controlling the service and policy configurations and by monitoring system and operations tasks. Oracle Service Bus relies on Oracle WebLogic Server run-time features.

This topic includes the following sections:

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 the Oracle Service Bus design-time environment. The underlying concepts of Oracle Service Bus are briefly described in this section. Oracle Service Bus manages the routing and transformation of messages in an enterprise system to promote seamless application integration. The following diagram illustrates key functional components of Oracle Service Bus. Oracle Service Bus provides additional features and protocol support not shown in the diagram.

Figure 2-2 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 development and run-time tooling. Proxy services are Oracle Service Bus definitions of intermediary Web services that Oracle Service Bus implements locally on Oracle 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 lets you implement proxy services independently and configure them dynamically, as driven by your business needs, without requiring costly infrastructure development and re-deployment efforts. The configuration functions are separated from the management functions in Oracle Service Bus.

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 proxy service 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 interface.

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 design-time tooling. To configure a business service, you must specify its interface, the 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 pipelines (a message flow).

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.

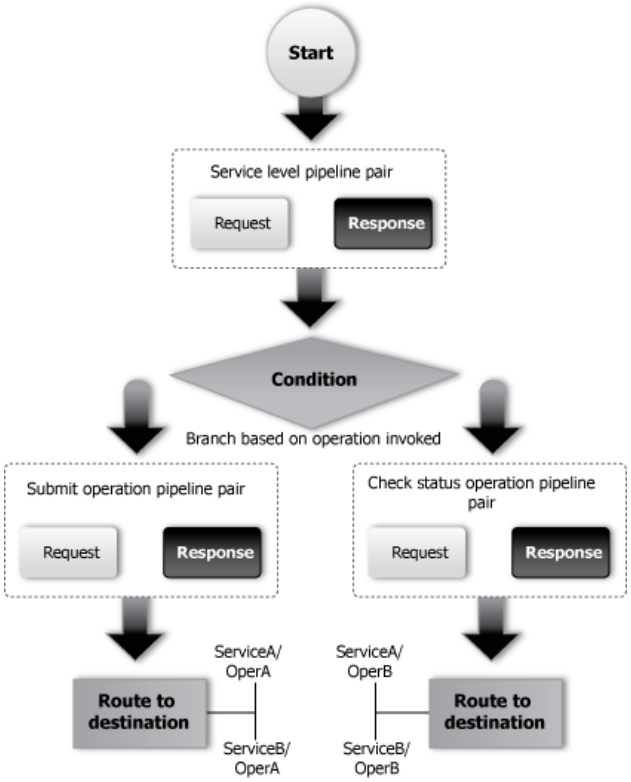
Pipeline Pairs

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.

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

Figure 2-3 Example of Operation Pipelines in a Proxy Service



Overview of the Tutorials

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

You will use the Oracle Service Bus Console design-time tooling in the tutorials, though you can perform similar tasks in the Oracle Service Bus IDE.

In the tutorials, you will use the console to create the proxy services that interact with the service clients and business services. As you work through the tutorials, 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 Description Language resources (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.
Gain a better understanding of the concepts underlying Oracle Service Bus.
Learn how to navigate through the Oracle Service Bus Console and demonstrate the capabilities of the console.
Perform the basic tasks that will help you to configure specific scenarios.

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.

After you complete the Getting Started steps for the tutorials, 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 tutorials include:

3 Getting Started with the Oracle Service Bus Tutorials describes the task to complete before starting the tutorials.
4 Tutorial 1. Routing a Loan Application is based on scenarios that demonstrate how Oracle Service Bus facilitates the routing of messages within an enterprise.
5 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.
6 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.

3 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 Oracle 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.

This topic consists of the following sections:

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. Create an Oracle Service Bus Domain
2. Start Oracle Service Bus
3. Load the Client Application

4. Log in to Oracle Service Bus Console

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

OSB_ORACLE_HOME\samples\servicebus\examples

Note: Oracle Service Bus version 11.1.1.3 does not include the example source files. To complete the tutorials, you must download the required files. For download information, see "Before You Begin" in 2 Introduction to the Oracle Service Bus Tutorials.

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 3-1 and Table 3-2. You can explore the other directories as an exercise.

Table 3-1 Sample Files Provided in Support of the Tutorials

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

Table 3-2 Sample Files Provided for Business Services the Tutorials

Filename/Location	Description
OSB_ORACLE_HOME\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.

Filename/Location	Description
*.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.

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 Oracle Fusion Middleware Configuration Wizard. Create an “Oracle Service Bus Extension – Single Server Domain Topology” domain called **ServiceBusTutorial** with a **Listen Port** of **7001**.

> **Note:**
> The example and tutorials are based on using an Admin-only domain. If the tutorial domain you create includes managed servers, you may need to change the port numbers in the services you create to the correct managed server ports.

Start Oracle Service Bus

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

On Windows systems, from the Windows Start menu in the **Oracle WebLogic** > **User Projects** group, or running startWebLogic.cmd in the new domain.
On UNIX systems, run startWebLogic.sh from the root of the new domain.

Load the Client Application

To test the loan application using the test console, you must first deploy the client jars in the Oracle WebLogic Server Administration Console, which are available at *OSB_ORACLE_HOME*\samples\servicebus\examples\build\webservices. The following client jars are available:

creditLoan_jws_basic_ejb
largeLoan_jws_basic_ejb

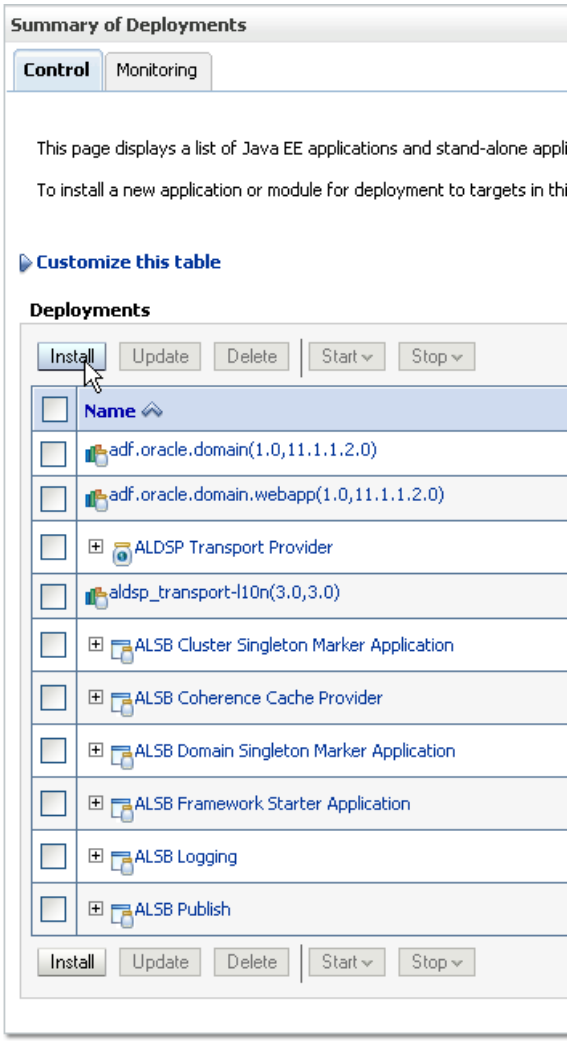
manager_jws_basic_ejb
normalLoan_jws_basic_ejb

To deploy creditLoan_jws_basic_ejb client jar:

1. Go to the Oracle 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 3-1).

Figure 3-1 Summary of Deployments



3. Click **Install**.

4. Navigate to *OSB_ORACLE_HOME*\samples\servicebus\examples\build\webservices.

5. Select **creditLoan_jws_basic_ejb**. Click **Next** to choose the targeting style.

6. Accept the remaining defaults by clicking **Next**, then 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 Oracle WebLogic Server console.

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

Open a browser window and enter the following URL to open the Oracle Service Bus Console for the ServiceBusTutorial domain:

http://localhost:7001/sbconsole

Log in if necessary with Oracle Service Bus Console the user name and password that you specified when you created the domain.

Where to Go from Here

After you complete the tasks required to set up the tutorials, you can proceed to 4 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.

4 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

You must complete 3 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:

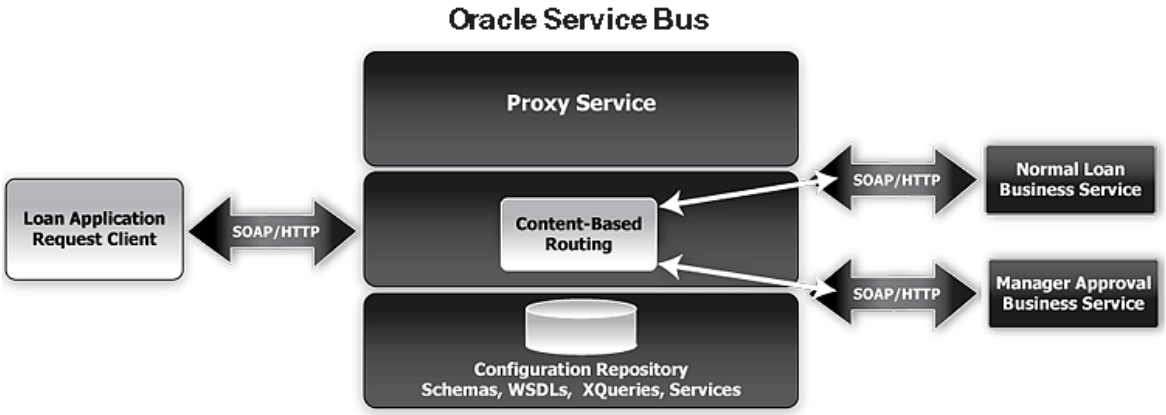
- Using Oracle Service Bus Console
- Implementation details of proxy services: message flows and actions
- Editing XQuery expressions using the XQuery Expression Editor
- Working with context variables to manipulate the content of the message

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 4-1 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 4-1 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 4-31.

> **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.

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

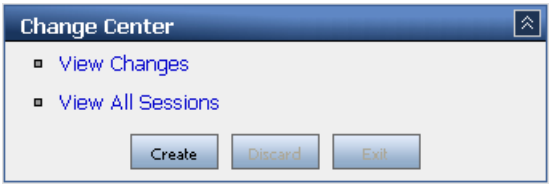
Ensure that Oracle Service Bus is running in the domain that you have created for the tutorial and that you have completed the tasks described in Setting Up the Tutorials.

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 4-2 illustrates the Change Center pane available on Oracle Service Bus Console.

Figure 4-2 Manage Sessions Using Change Center



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

Table 4-1 Change Center Options

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

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

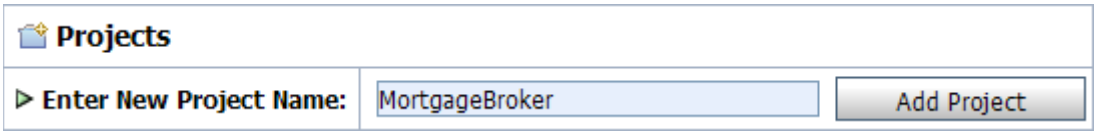
1. In the Oracle Service Bus Console Change Center click **Create** to create a new session.

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 4-3.

   Figure 4-3 Create a Project



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 4-4.

   Figure 4-4 Create a Folders

3. Click Add Folder.

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

4. Repeat the previous steps 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

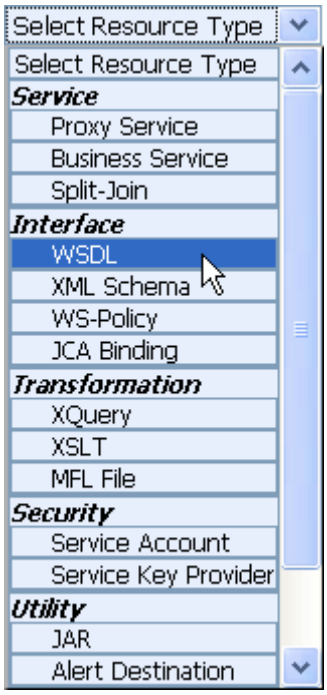
1. In the Oracle Service Bus Console Change Center click **Create** to create a new session.

2. In Oracle Service Bus Console navigation pane, select the **Project Explorer**.

3. In the project explorer, click the **WSDL** folder you created.

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

   Figure 4-5 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 4-6):
    a. Enter normalLoan as the resource name.
    b. Click **Browse**. Select the WSDL associated with the normal loan process:

       OSB_ORACLE_HOME\samples\servicebus\examples\src\examples\web services\jws_basic\normal\NormalLoanApprovalService.wsdl

       Note: In Oracle Service Bus 11.1.1.3, get the WSDL from the location where you extracted the downloaded resources as described in "Before You Begin" in 2 Introduction to the Oracle Service Bus Tutorials.

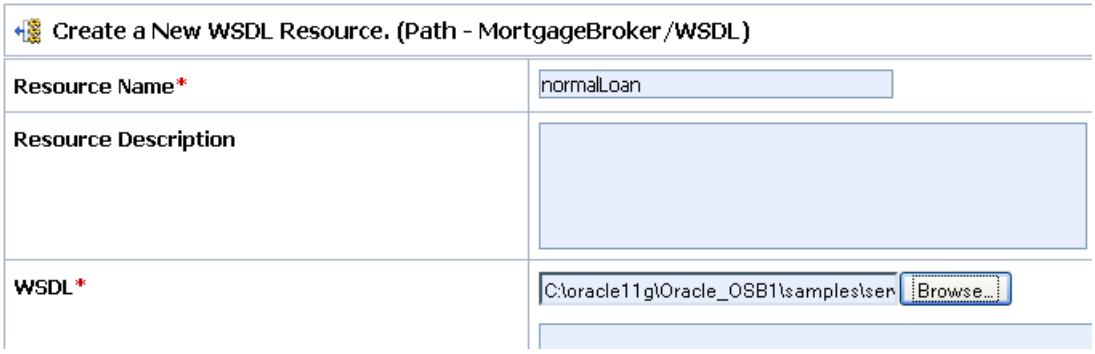
    c. Optionally, enter information in the Resource Description field to describe the resource.

       The resource is now displayed in the resources pane, as shown in Figure 4-6.

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

    e. Click **Activate** in the Change Center to activate the session.

Figure 4-6 Create a WSDL Resource



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

Create the managerApproval WSDL Resource

Complete the steps in the Create the managerApproval WSDL Resource 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 to create the resource in the project's /WSDL folder.

Configuration Parameter	**Value**
WSDL Resource Name	managerApproval
ManagerApprovalService WSDL	*OSB_ORACLE_HOME*\samples\servicebus\examples\src\examples\webservices\jws_basic\manager\ManagerApprovalService.wsdl **Note**: In Oracle Service Bus 11.1.1.3, get the WSDL from the location where you extracted the downloaded resources as described in "Before You Begin" in 2 Introduction to the Oracle Service Bus Tutorials.

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.

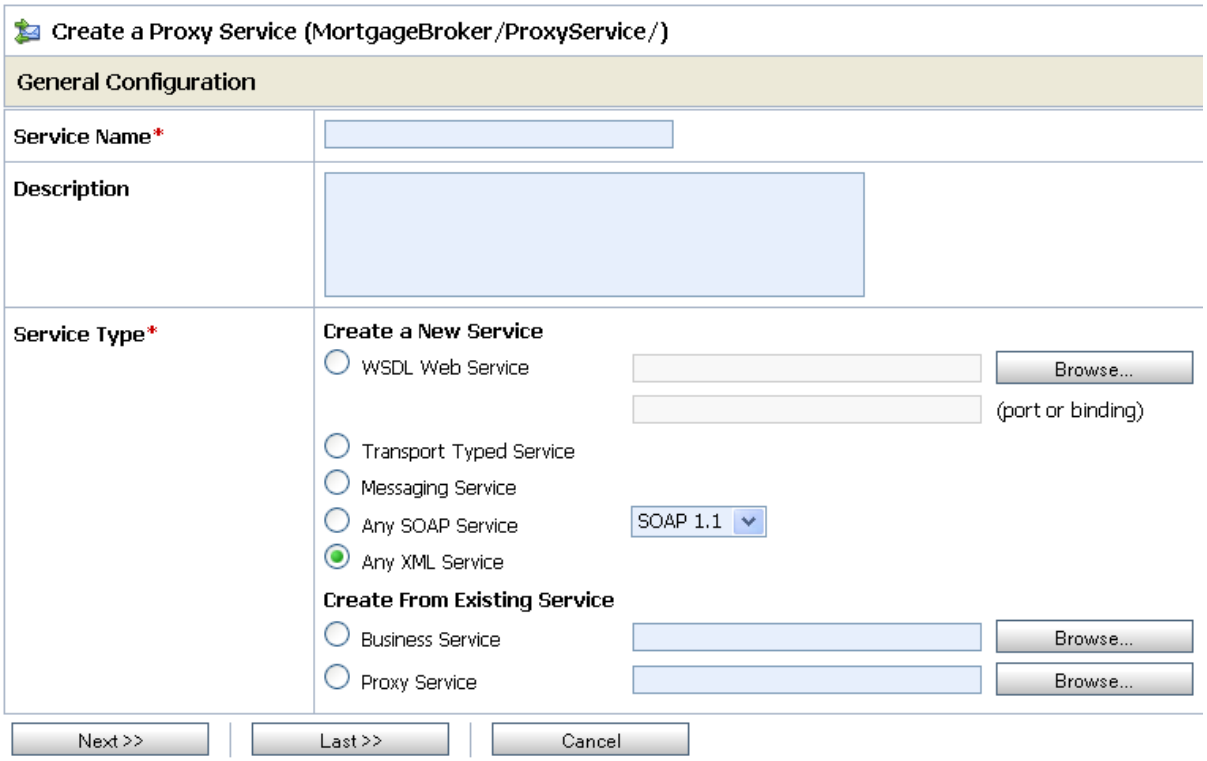
2. In the Project Explorer, select the **MortgageBroker** project. The project folder is expanded to show the directory structure of the project.

3. Select the **ProxyService** folder.

4. In the Select Resource Type list, select **Proxy Service**.

   The Create a Proxy Service - General Configuration page is displayed as shown in Figure 4-7.

Figure 4-7 Create a Proxy Service



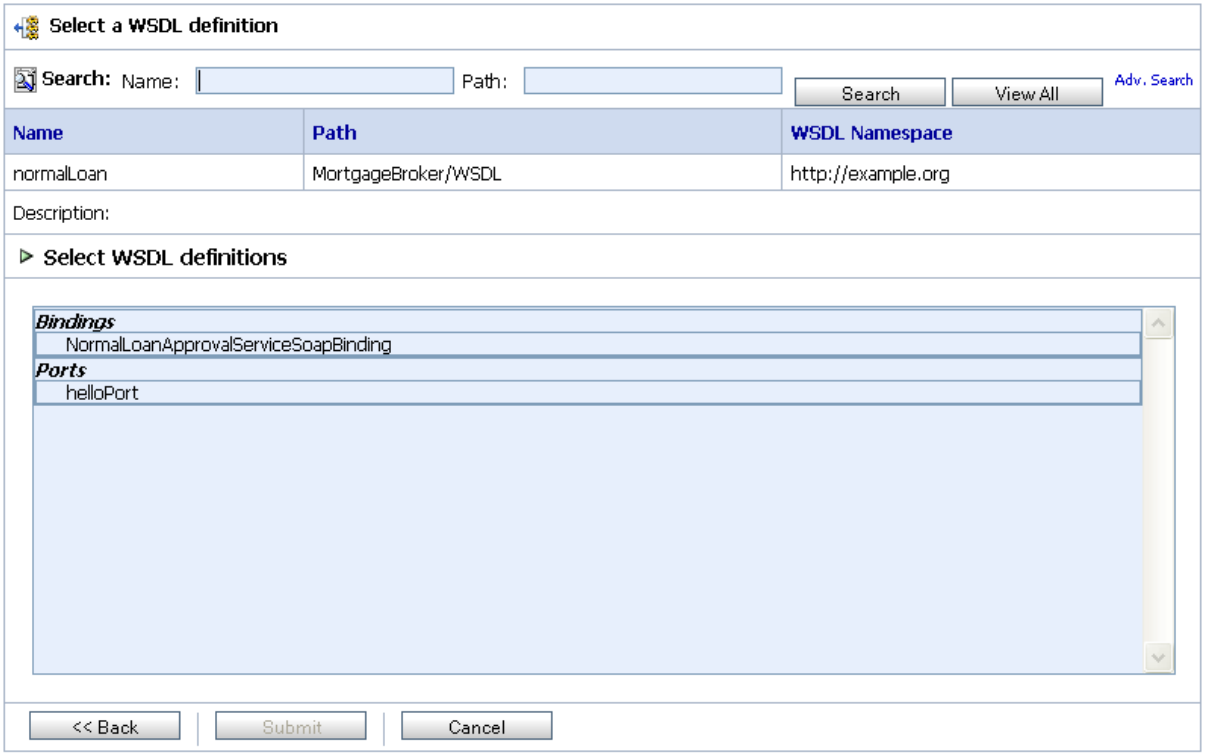
5. Give the proxy service a Service Name of **LoanGateway1**.

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.

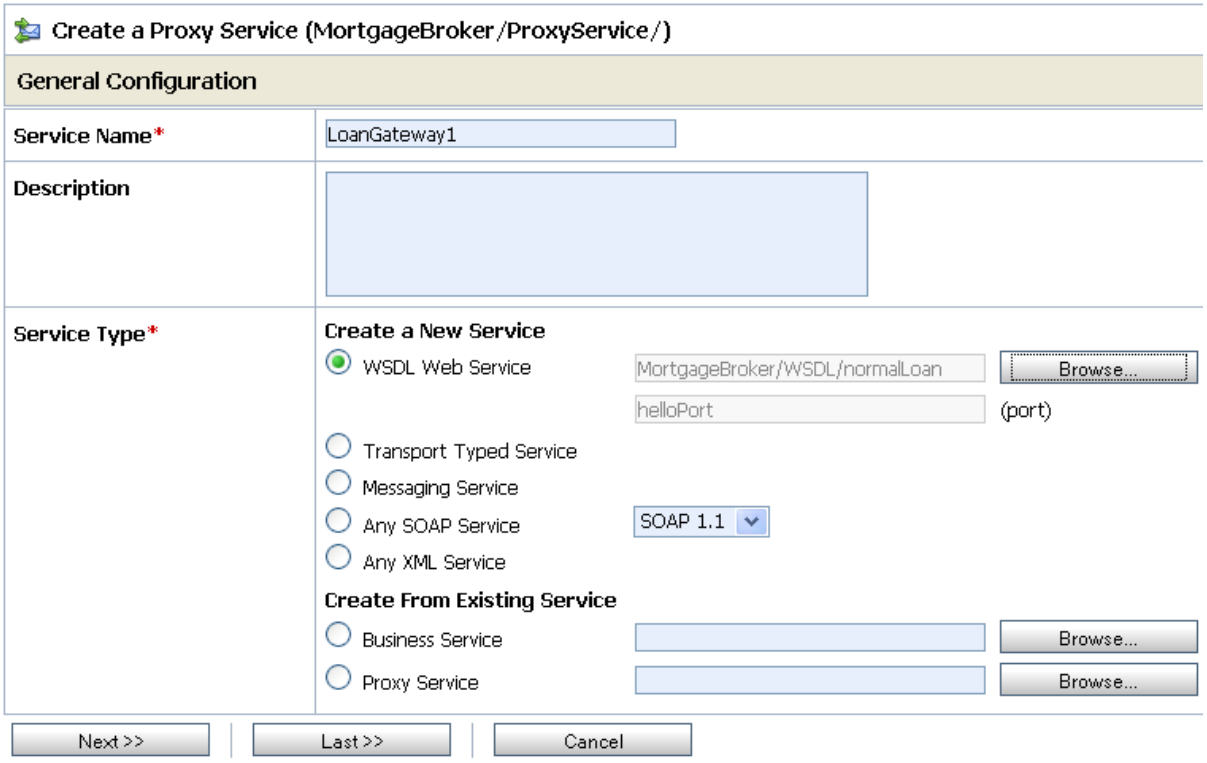
Figure 4-8 Select WSDL Definition



8. In Select WSDL Definitions pane, from the Ports category, Click **helloPort**, which is the WSDL port for the normalLoan WSDL (helloPort).

9. Click **Submit**. The port name is displayed in the WSDL port text field as shown in Figure 4-9.

Figure 4-9 LoanGateway1 Proxy Service



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

11. 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.

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

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

14. Click **Next**.

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

16. On the Message Handling page, click **Next**.

17. 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.

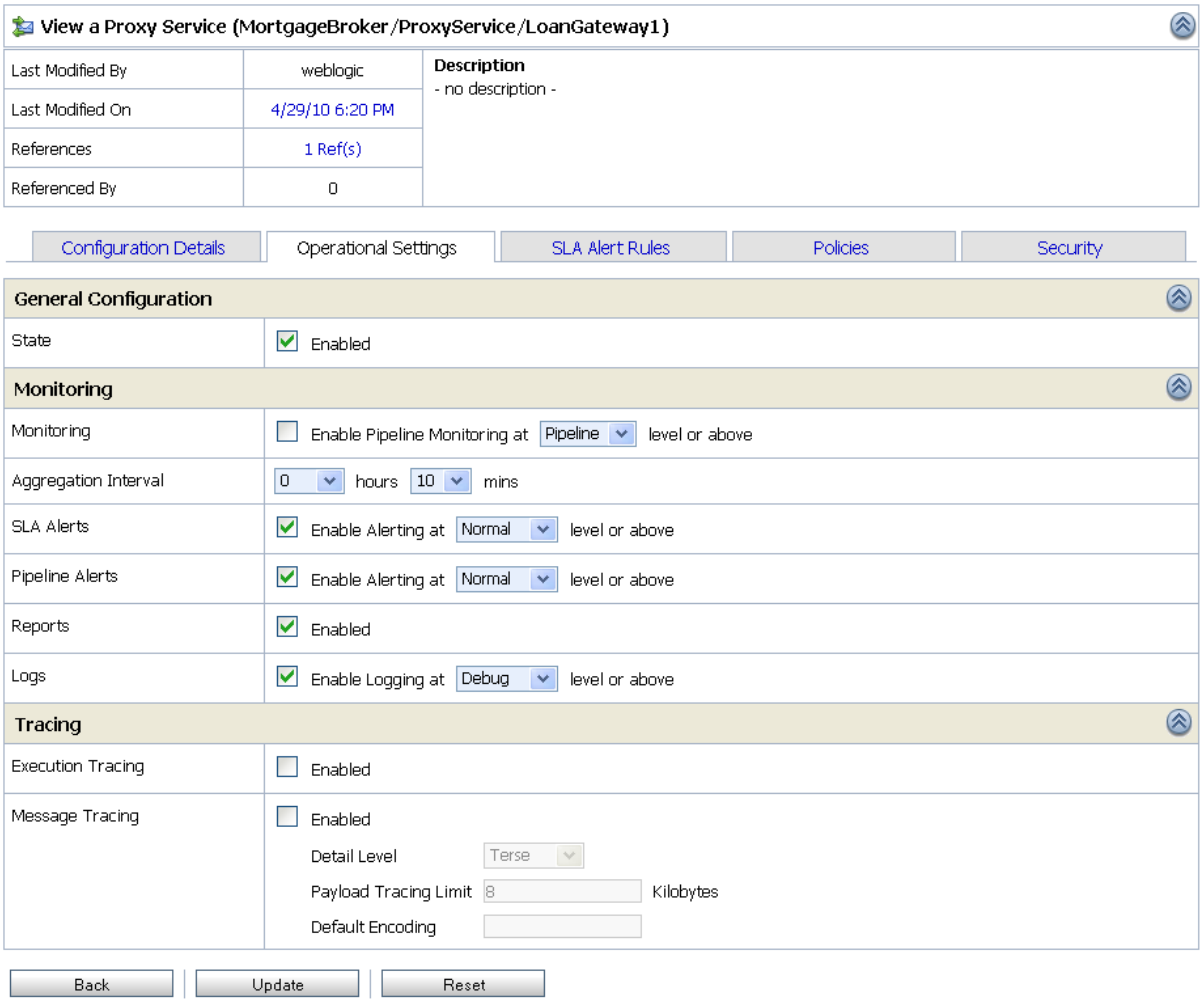
    Before registering the proxy service, you can review the configuration settings and change them if necessary, by clicking Edit icon.

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

19. Click the **LoanGateway1** link in the Resources pane of MortgageBroker/ProxyServices to go the View a Proxy Service page. The View a Proxy Service page is displayed.

    The Operational Settings tab, as shown in Figure 4-11, has options to enable monitoring for the proxy service.

Figure 4-11 Monitoring Configuration of Proxy Service



To enable monitoring:

1. Click the **Operational Settings** tab.

2. Select the **Enable Pipeline Monitoring** check box for the Monitoring setting.

3. 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.

4. Click **Update**.

5. **Activate** the session.

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

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.

Create the NormalLoan Business Service

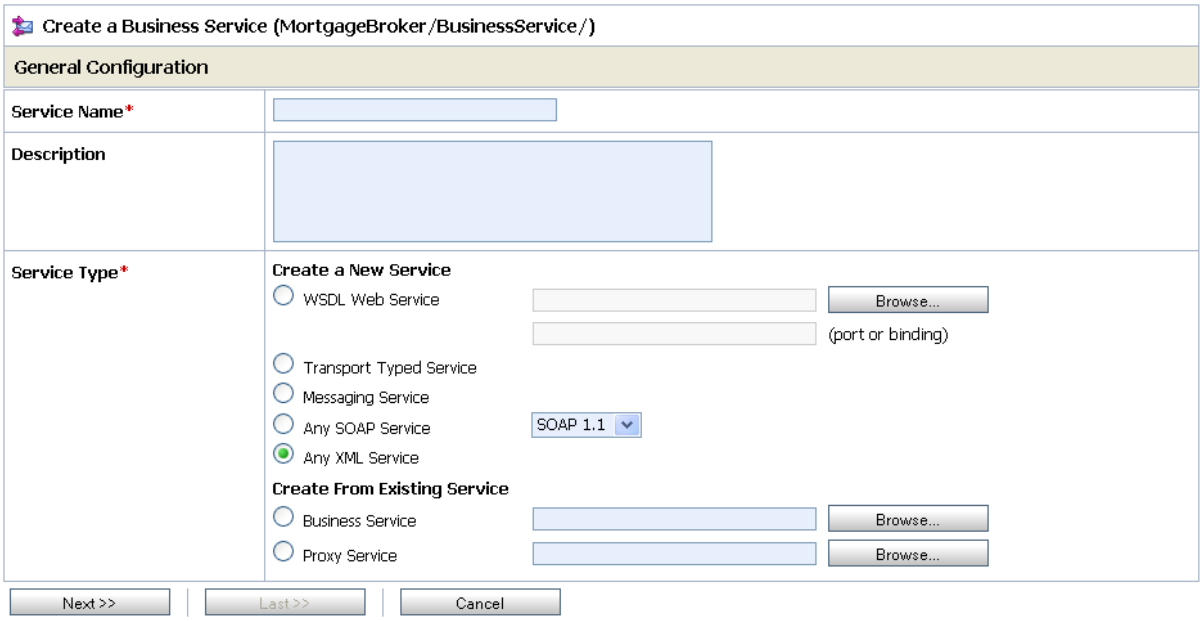
1. In the Oracle Service Bus Console **Change Center** click **Create** to create a new session.

2. In the Project Explorer, select the **BusinessService** folder.

3. In the Select Resource Type list, select **Business Service**.

   The Create a Business Service - General Configuration page is displayed as shown in Figure 4-12.

Figure 4-12 Create Business Service



4. Enter a name for the service. In this case, enter **NormalLoan**.

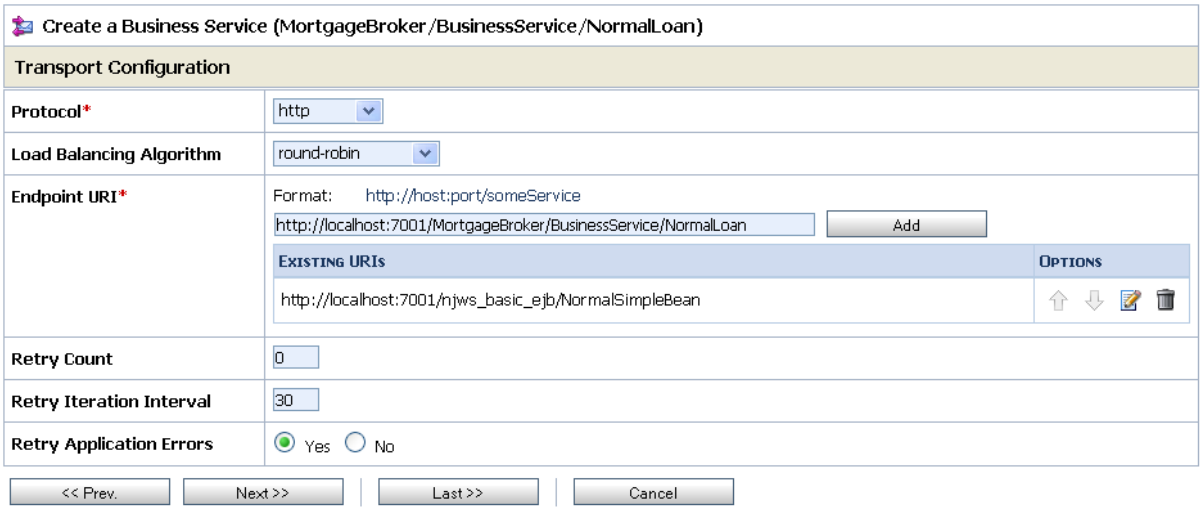
5. For Service Type, select **WSDL Web Service**, then click **Browse**. Select the WSDL and port on which to base the business service:
   a. In the WSDL Browser, select the normalLoan WSDL
   b. Select the port (`helloPort`) for the WSDL.
   c. Click **Submit**.

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

6. Click **Next**.

   The Create a Business Service - Transport Configuration page is displayed as shown in Figure 4-13.

Figure 4-13 Transport Configuration of a Business Service



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 affect 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 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.
    d. Click **Update**.
    e. **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 previous section on creating the ManagerLoanReview business service, but use the configuration parameters in the following table.

Configuration Parameter	Value
Business Service Name	ManagerLoanReview
ManagerApprovalService WSDL	managerApproval helloPort
Load Balancing Algorithm	none
Endpoint URI	http://<host:port>/mjws_basic_ejb/ManagerSimpleBean (this should already be populated in the “Existing URIs” field.)

Summary

You have created two business services (NormalLoan and ManagerLoanReview), a proxy service (LoanGateway1) and two WSDL resources (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 in the proxy service message flow.

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.

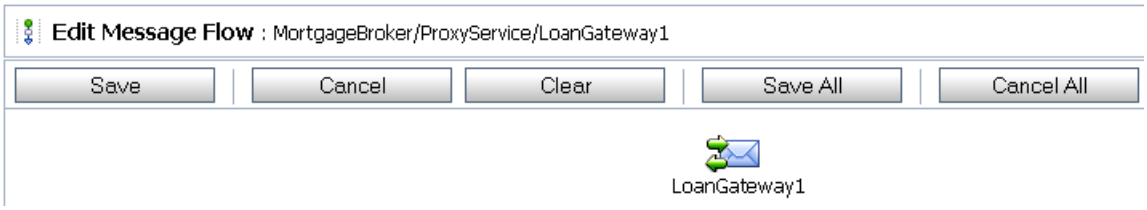
Configure the Routing Behavior of the Message Flow

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 4-14.

Figure 4-14 Edit Message Flow for LoanGateway1 Proxy Service



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

   RouteNode1 is added in the configuration page as shown in Figure 4-15.

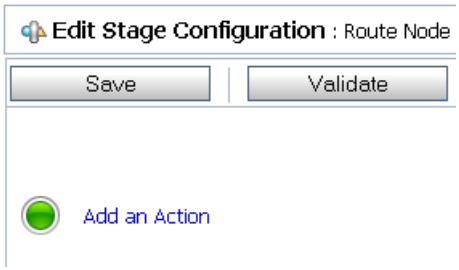
Figure 4-15 Add a Route Node

5. Click **RouteNode1** and select **Edit Route** from the menu options.

   The Edit Stage Configuration: Route Node page is displayed as shown in Figure 4-16. 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.

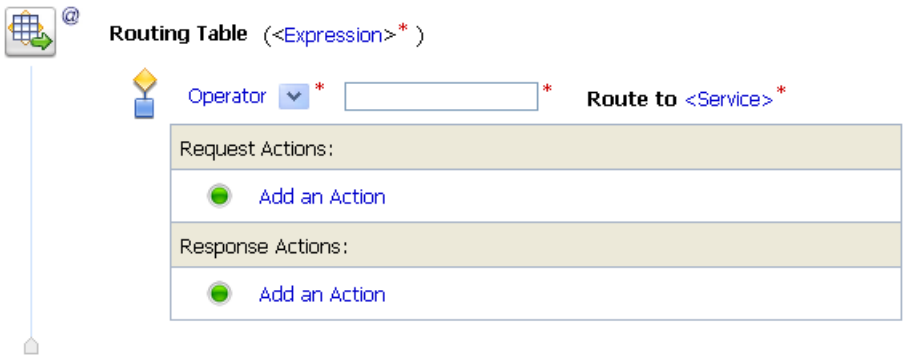
Figure 4-16 Add an Action Link



6. Click the **Add an Action** link, and select **Communication** > **Routing Table**.

   The Edit Stage Configuration page changes to display routing table configuration page, as shown in the following figure.

Figure 4-17 Routing Table Configuration Page



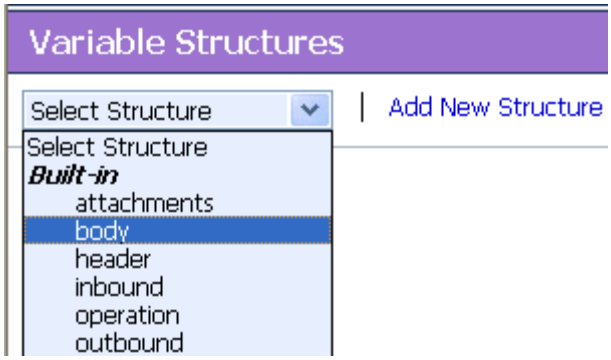
   You must configure the routing table to route messages to business services based on the value of the interest rate element in the incoming message. You can configure content-based routing by creating an XQuery expression, using the XQuery Expression Editor.

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

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

9. The Variable Structures pane is displayed. Select **body** from the Select Structure list.

Figure 4-19 Variable Structures Pane

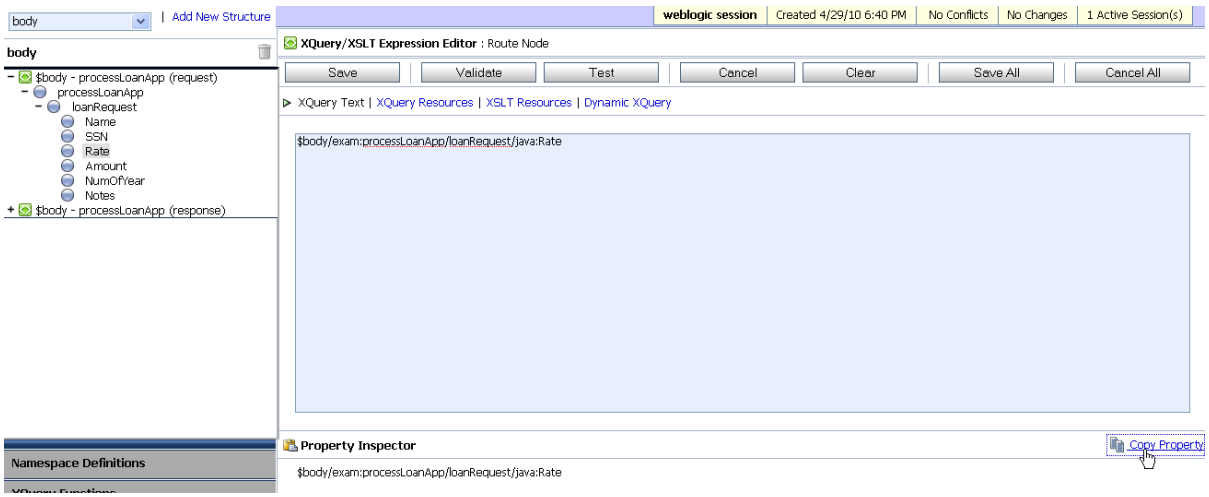


A structural representation of the body element is displayed in the Variable Structures pane.

10. Click **+** to expand $body - processLoanApp (request) > processLoanApp > loanRequest.

    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 4-21 loanRequest Element



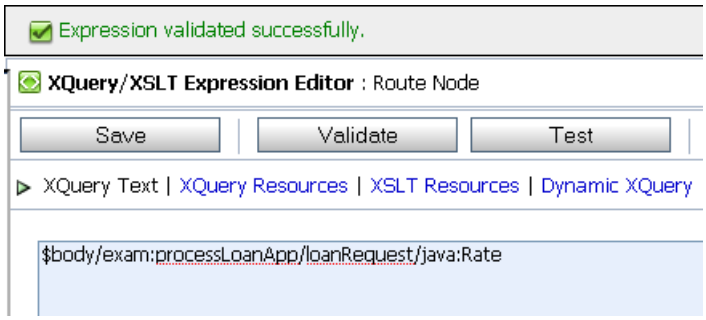
11. Click the **Rate** element. The property appears below the XQuery editor in the Property Inspector field. Click in the XQuery text box and click **Copy Property** in the lower right of the XQuery Expression text box. The following XQuery expression is written in the text box:

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

Note:
In Internet Explorer (IE) browsers, you can simply drag the icon next to the Rate element into the XQuery text editor.

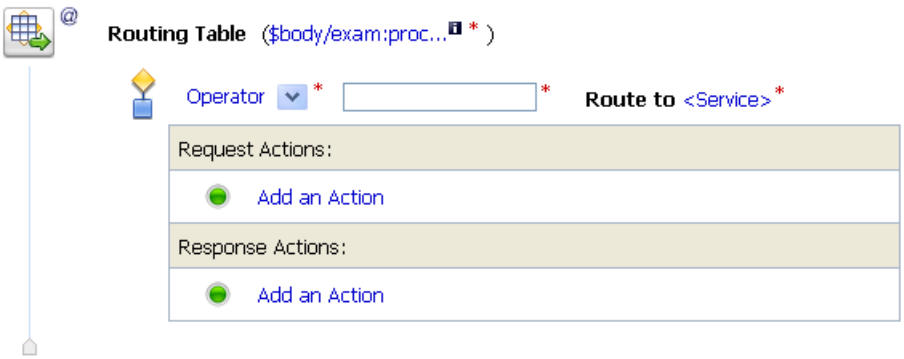
12. Click **Validate** to validate the XQuery. It is a good practice to do this before you submit the expression. The expression is validated for syntax. If there are errors in the expression, they are displayed directly above the Validate button.

Figure 4-23 XQuery Validation



13. 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 4-24 Routing Table View - Expression



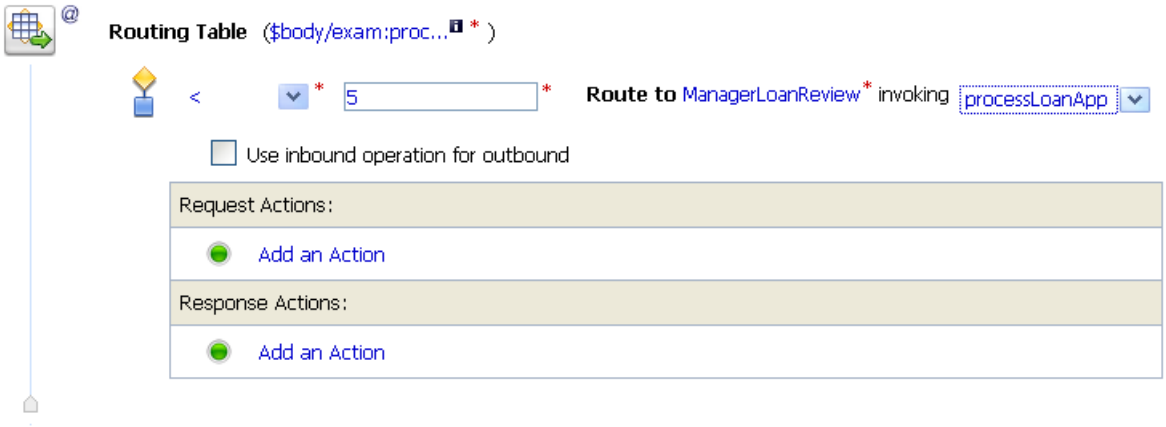
14. In the Operator list, select <. In the associated text field enter the number **5**.

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.

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

16. In the Routing Table Operation 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.

Figure 4-25 Routing Table View - Routing Condition



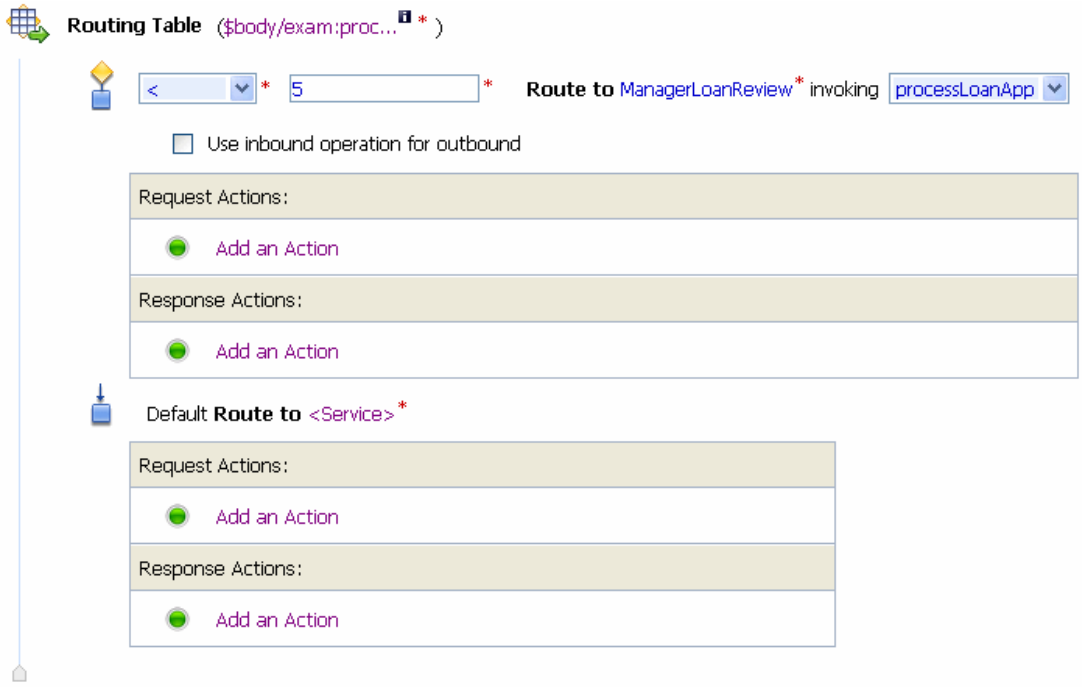
You have now defined the case to route the loan application to the ManagerLoanReview business service. If the rate specified in the loan application is equal to or greater than five percent, then the message is routed to the NormalLoan business service. The next section describes how you can add a condition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

Add a Default (Else) Condition to the Routing Table

Follow these steps:

1. 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 4-26 Routing Table Default Condition



2. In the Default (else) condition, click the **Service** link. The Select Service page is displayed.
    a. Select the **NormalLoan** business service.
    b. Click **Submit**.

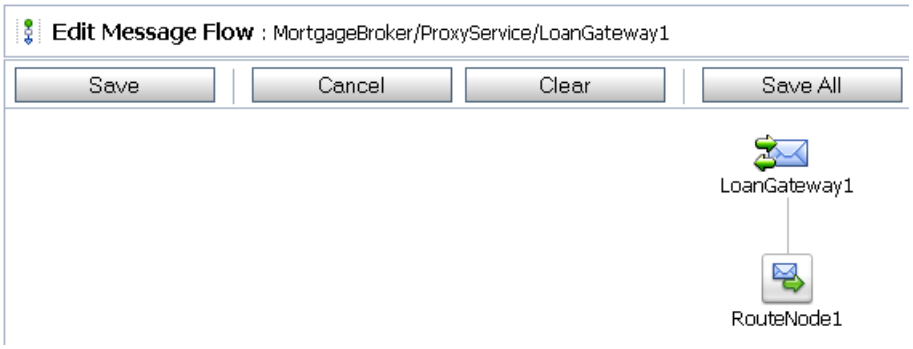
3. In the Operation list, select the **processLoanApp** operation for the service.

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

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

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

Figure 4-28 LoanGateway1 Proxy Service Message Flow Map



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

7. In the Change Center, **Activate** the session.

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.

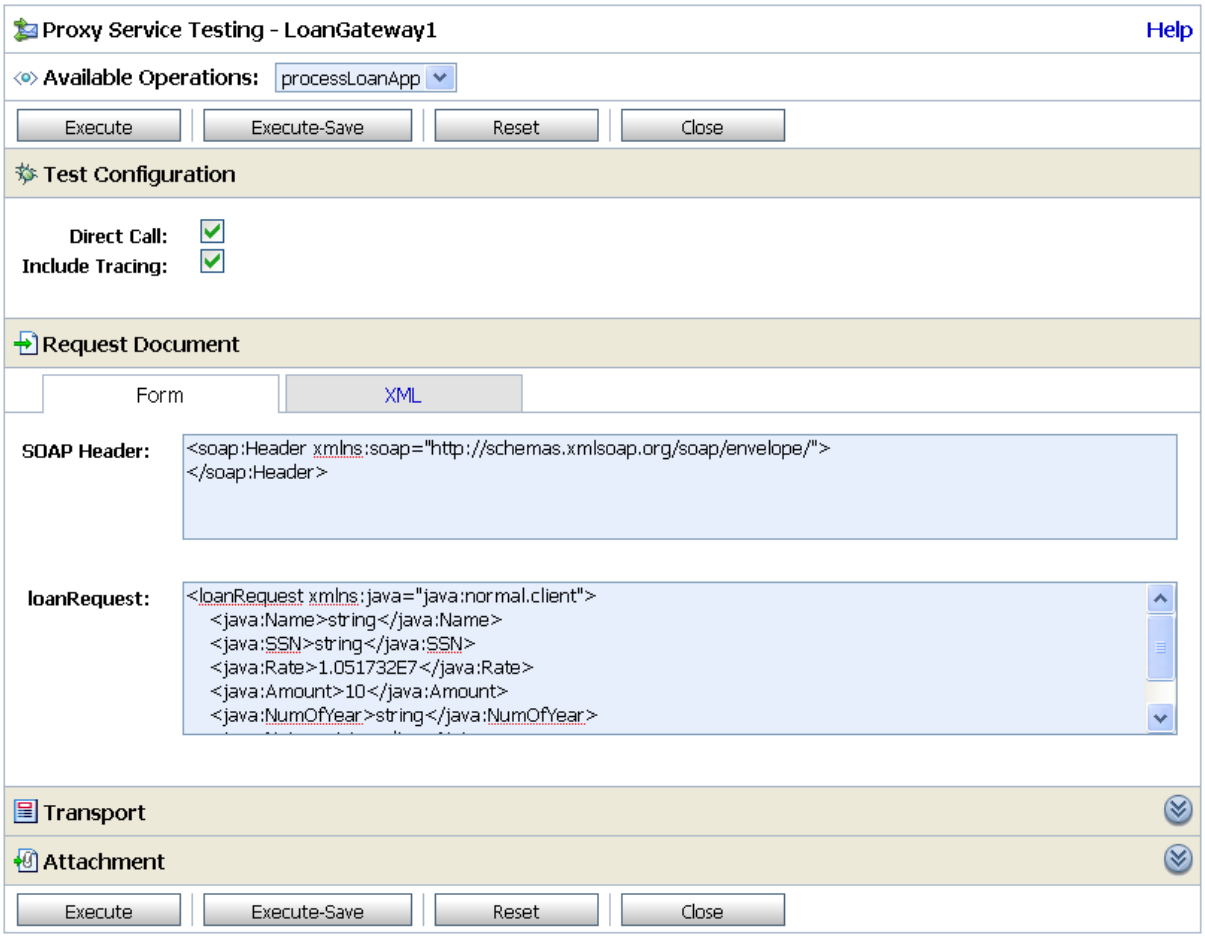
To test the Routing of the Loan Application ManagerLoanReviewService

1. In Oracle Service Bus Console, make sure the current session is activated.

2. Select **Resource Browser** > **Proxy Services** in the left panel.

3. Click on **Launch Test Console** icon in Actions column for LoanGateway1 to launch the test console (see Figure 4-29).

Figure 4-29 Test Console for LoanGateway1



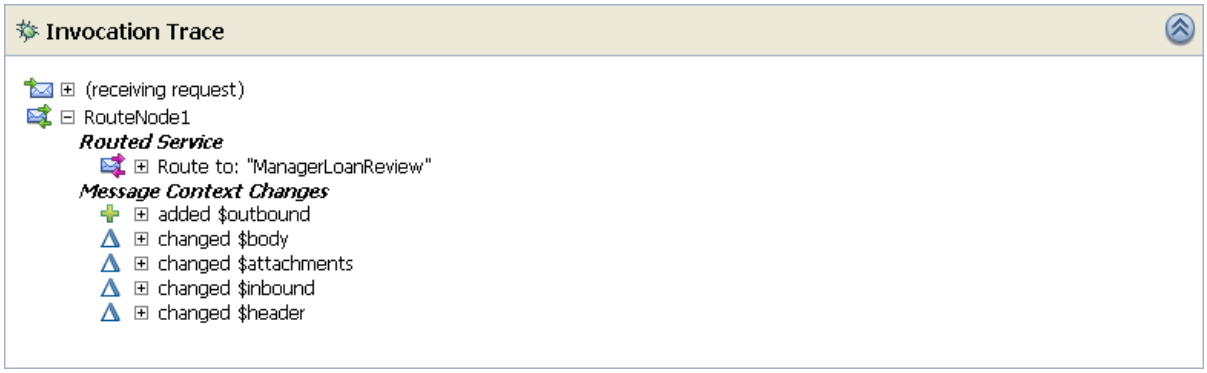
4. Replace the default payload in loanRequest field with the following code:

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

5. Accept other default settings and click **Execute**. The response as in Figure 4-30 is obtained.

Figure 4-30 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. In Oracle Service Bus Console, make sure the session is activated.

2. Click on **Resource Browser** > **Proxy Services** in the left panel.

3. Click on **Launch Test Console** icon in Actions column for LoanGateway1 to launch the test console.

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

```
<loanRequest xmlns:java="java:normal.client">
    <!--Optional:-->
    <java:Name>Smith</java:Name>
    <!--Optional:-->
    <java:SSN>1234567</java:SSN>
    <!--Optional:-->
    <java:Rate>5.3</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 4-31 is obtained.

Figure 4-31 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 5 Tutorial 2. Transforming a Loan Application.

5 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

You must complete 4 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 4 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

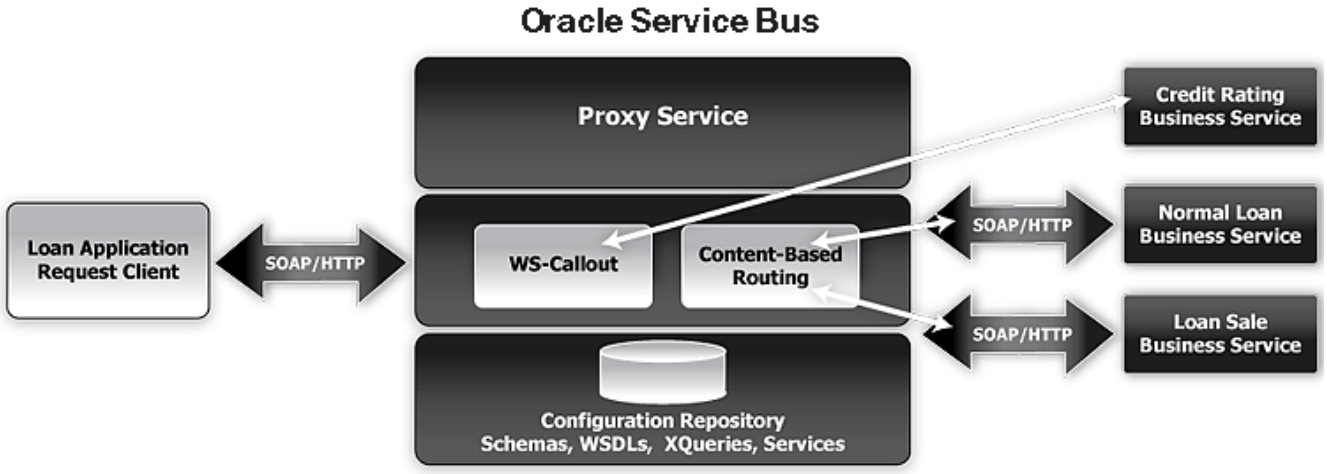
A primary mortgage company uses Oracle Service Bus to identify and re-route loan applications that can be sold to secondary loan companies. Loan applications with a principal request of greater than US $25 million are candidates for sale to a secondary loan company. When Oracle Service Bus receives a 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 5-1 illustrates where Oracle Service Bus fits in your enterprise to mediate the messaging between the enterprise services and the business services.

Figure 5-1 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.

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 4 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 4 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
4. Complete the following steps to create the necessary resources. Be sure to **Activate** your session after creating the resources.

Create the WSDL Resources

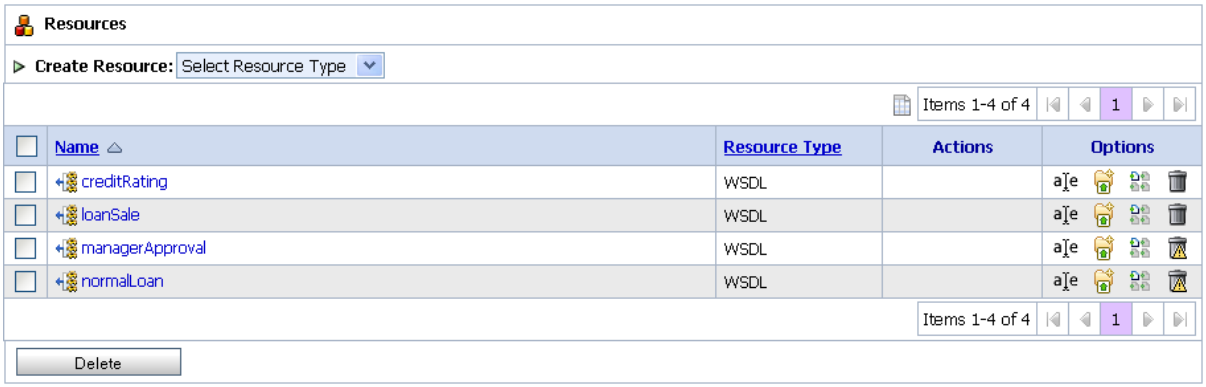
You must create the WSDL resources (loanSale and creditRating) before creating the other resources required for this scenario. WSDL resources are the building blocks for creating the business and proxy services. To import the appropriate WSDLs and create the WSDL resources, follow the tasks described in To Create the normalLoan WSDL Resource section in 4 Tutorial 1. Routing a Loan Application. However, for this instance, name your resources and base them on the WSDLs listed in the following table.

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

Note: In Oracle Service Bus 11.1.1.3, get the WSDLs from the location where you extracted the downloaded resources as described in “Before You Begin” in 2 Introduction to the Oracle Service Bus Tutorials.

When you complete this task, the MortgageBroker/WSDL folder contains the WSDL resources created in this tutorial and in 4 Tutorial 1. Routing a Loan Application as shown in Figure 5-2.

Figure 5-2 WSDL Resources



Create a Proxy Service

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

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

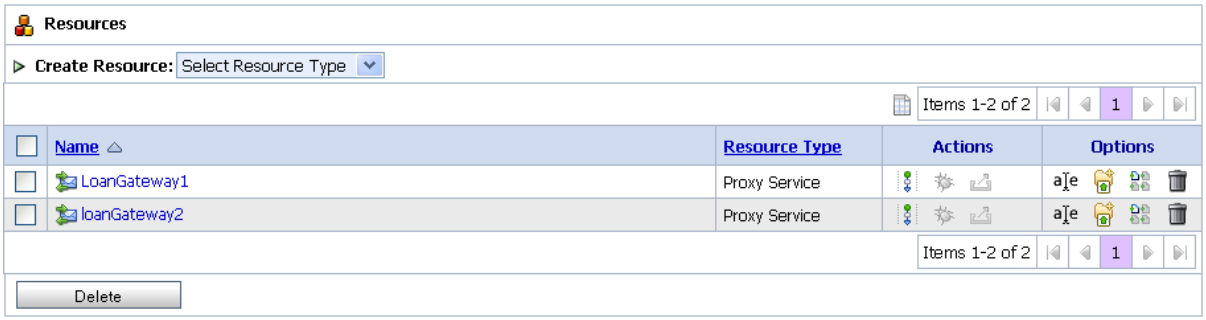
To create a new proxy service, **LoanGateway2**, follow the tasks described in Create a Proxy Service For this instance, use the proxy service name and the endpoint URI listed in the following table.

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

After you complete this 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 4 Tutorial 1. Routing a Loan Application as shown in Figure 5-3.

Figure 5-3 Proxy Service Resources



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:

> NormalLoan (you have already created this) – 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.
> 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).
> 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 4 Tutorial 1. Routing a Loan Application.

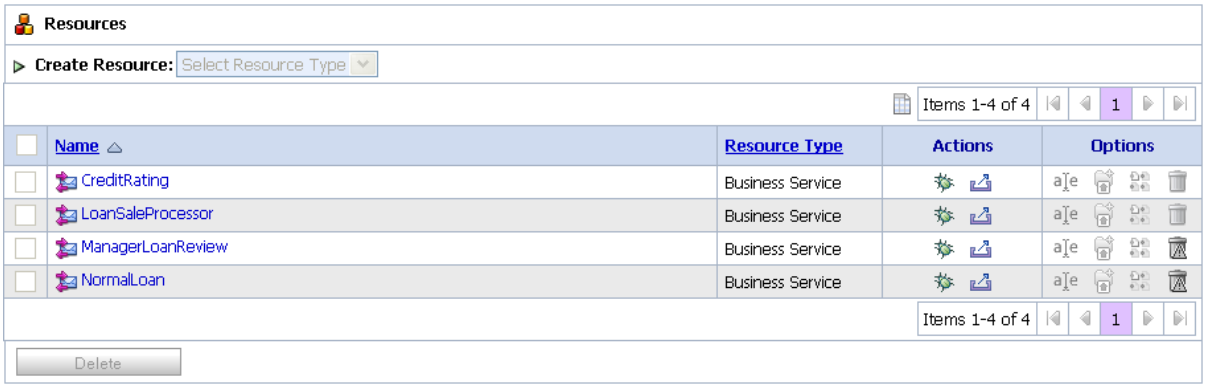
To create the **LoanSaleProcessor** and **CreditRatingService** business services for this scenario, follow the tasks described in Create the Business Services. You must configure your services using the names, service types, and endpoint URIs listed in the following two tables.

Business Service Name	**LoanSaleProcessor**
Service Type	Select the **WSDL Web Service** as the Service Type option. Select the **loanSale** WSDL in the Select a WSDL page. **Select the helloPort** 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 **http://<host:port>/ljws_basic_ejb/LargeSimpleBean** where <host:port> represents the machine and port on which the Oracle Service Bus server is running.

Business Service Name	**CreditRating**
Service Type	Select the **WSDL Web Service** as the Service Type option. Select the **creditRating** WSDL in the Select a WSDL page. Select the **helloPort** 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 **http://<host:port>/crejws_basic_ejb/CreditSimpleBean**. 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 4 Tutorial 1. Routing a Loan Application as shown in Figure 5-4.

Figure 5-4 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. Complete the following tasks.

Create a Routing Table

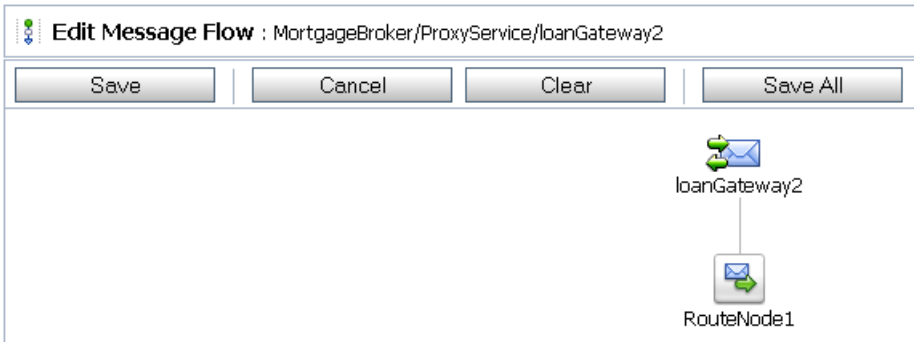
1. In the Oracle Service Bus Console navigation pane, select **Resource Browser**.

   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.

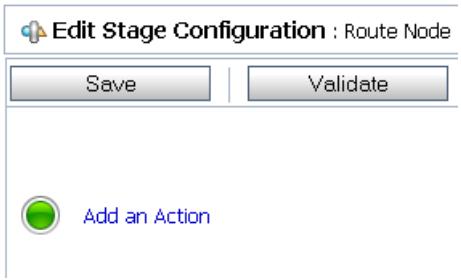
3. Click **LoanGateway2**, then select **Add Route** from the menu. In the configuration dialog, RouteNode1 is added.

Figure 5-5 Edit Message Flow - LoanGateway2 RouteNode1



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

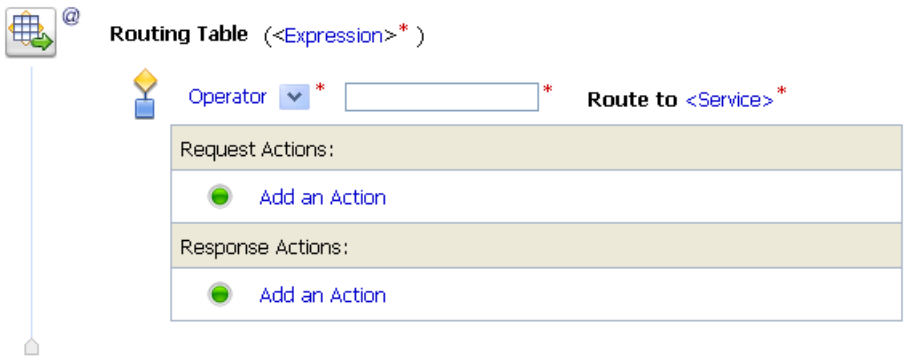
Figure 5-6 Add an Action Link



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

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

Figure 5-7 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 list in the Variable Structures pane.

4. A structural representation of the body element is displayed in the Variable Structures pane. Click **+** to expand **$body – processLoanApp (Request)** > **processLoanApp** > **loanRequest**.

   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.

5. Click the **Amount** element. The property appears below the XQuery editor in the Property Inspector field. Click in the XQuery text box and click **Copy Property** in the lower right of the XQuery Expression text box. The following XQuery expression is written in the text box:

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

   Note:
   In Internet Explorer (IE) browsers, you can simply drag the icon next to the Rate element into the XQuery text editor.

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

7. 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.

8. In the **Operator** 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.

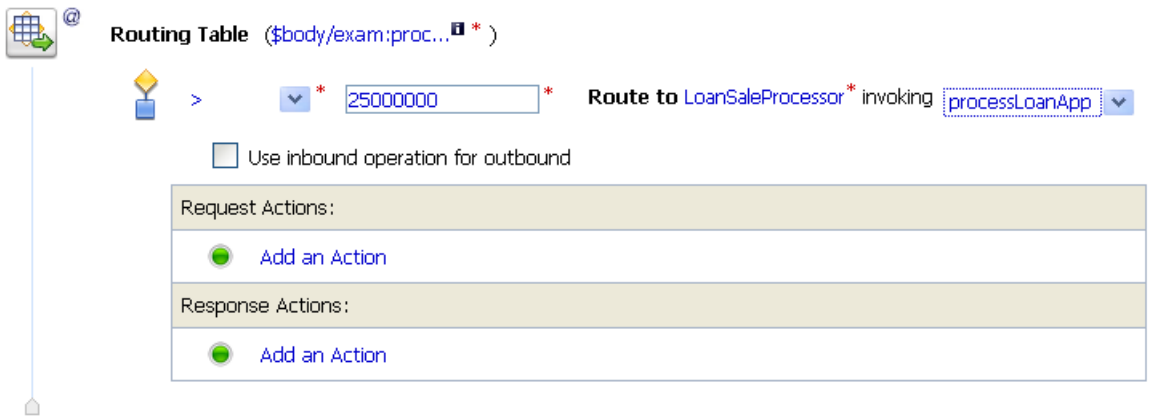
9. Click the **Service** link to define the service to which you want to route when the amount is greater than US $25. Select **LoanSaleProcessor**, and click **Submit**.

10. In the **Operation** 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 5-8.

Figure 5-8 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
      ```

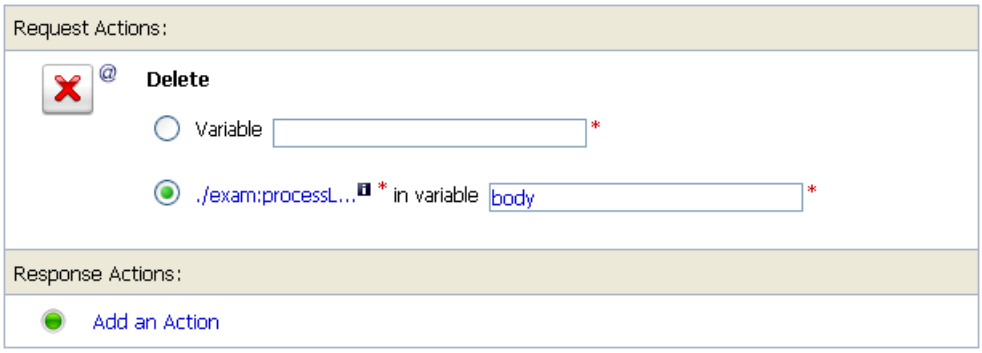
   b. Click **Validate**, then **Save**. You will return to the Edit Stage Configuration page.

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 5-9.

Figure 5-9 LoanGateway2 Routing Request Delete Action



Assign a Service Callout Input Parameter for the Service Callout

1. In the Request Actions pane of the Routing Table, click the **Delete** icon to bring up the menu, 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 list.

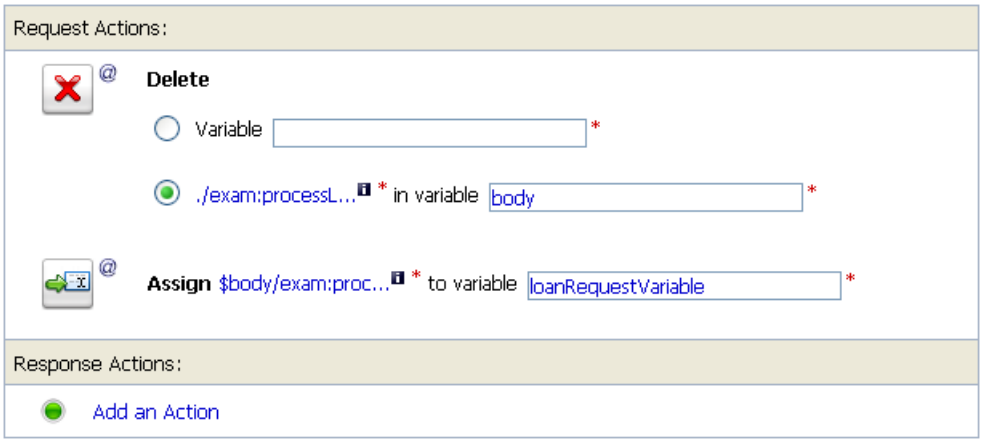
4. Expand the request element list and copy the **loanRequest** element into the XQuery expression text box. The following XQuery expression is written in the text box:

   ```
   $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 5-10.

Figure 5-10 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 the Namespace of the Input Parameter you Assigned for the Service Callout

1. In the Request Actions pane of the Routing Table, click the **Assign** icon to bring up the menu, 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 **Validate** and **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

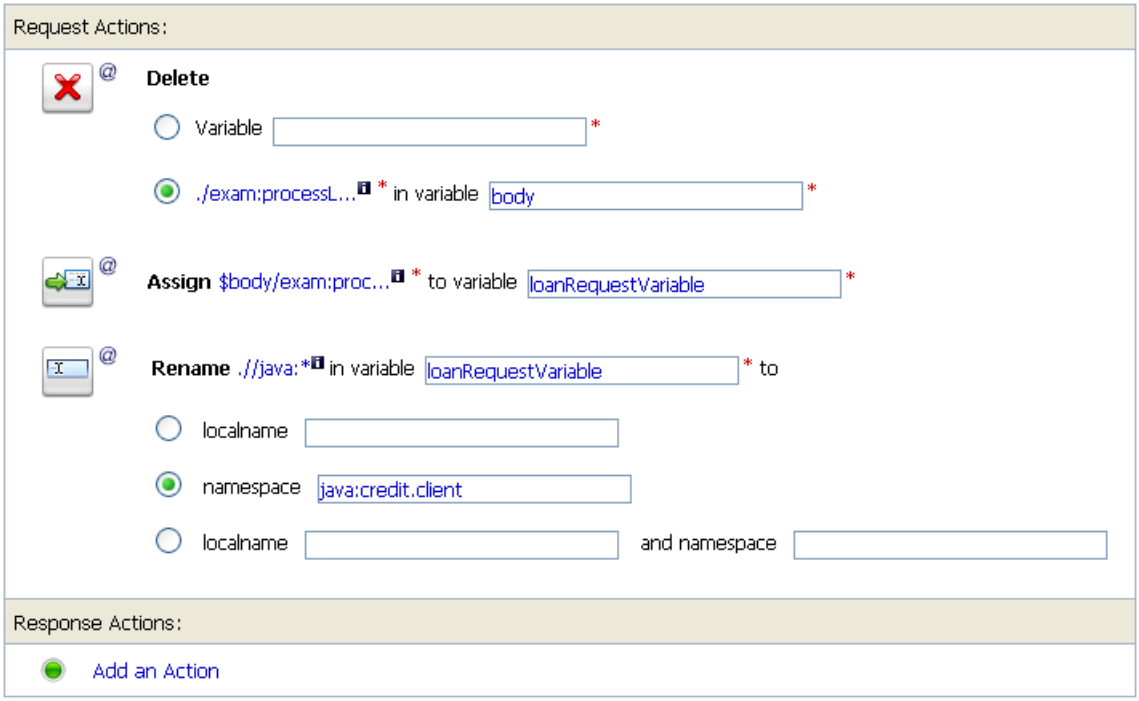
variable. The next task specifies the namespace to replace the namespaces identified by the XPath expression.

6. Select the **namespace** option and enter the following fully qualified new namespace to be substituted in the message: **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 5-11.

Figure 5-11 LoanGateway2 Routing Request Rename Action



Add a Service Callout Action

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

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

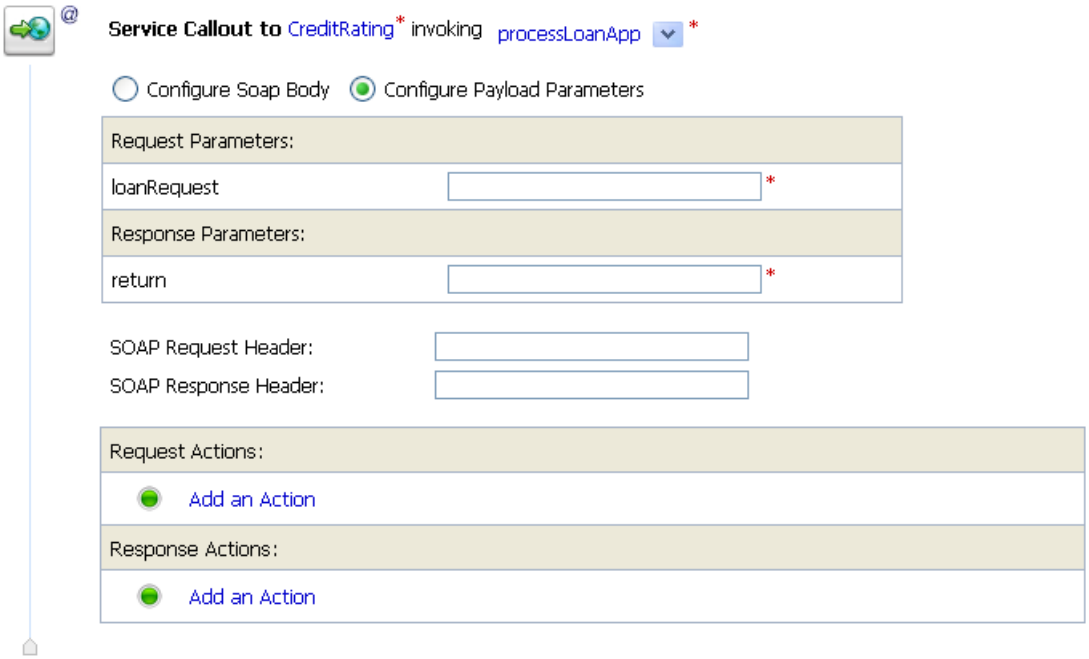
   The Service Callout action is added to the Request Action pipeline in the Routing Table.

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 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 5-12.

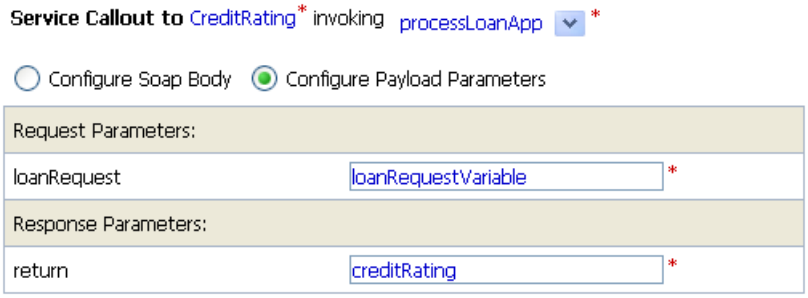
Figure 5-12 Service Callout Action - Configured Operation



4. In the service callout “Request Parameters” section, set the loanRequest parameter to **loanRequestVariable** (the message context variable you defined in the previous section).

5. In the service callout “Response Parameters” section, set the return parameter name to **creditRating** as shown in Figure 5-13.

Figure 5-13 Service Callout Action - Configured Parameters



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, described in this section.
Insert an new element into the message by adding an element action in the configuration, described in this section.

Rename the Message Namespace

1. Click the **Service Callout** icon to bring up the menu, 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 **Validate** and **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.

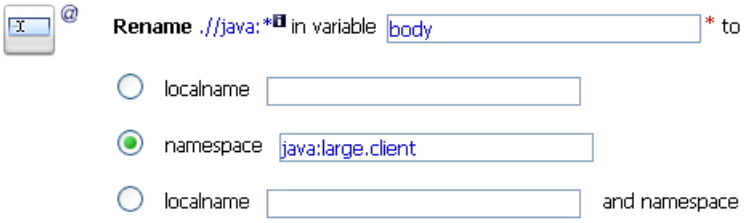
   You have completed creating a condition in which the XPath expression finds all the namespaces with the java prefix in the body context variable. The next task specifies the namespace to replace the namespaces identified by the XPath expression.

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 5-14.

Figure 5-14 Message Namespace Rename Action



Insert a New Element in the Message

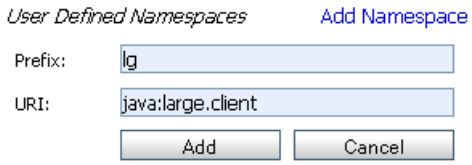
1. Click the **Rename** icon to bring up the menu, and select **Add an Action** > **Message Processing** > **Insert**.

   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 5-15.

Figure 5-15 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 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 list in the Variables Structures pane, select **body**.
c. Expand the request element copy the **Notes** element into the XPath Expression text box.

   The following XPath expression is written to the text box:

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

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 5-16.

Figure 5-16 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 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, described in this section.
Revert the namespace to the namespace in the original message: java:normal.client, described in this section.

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.

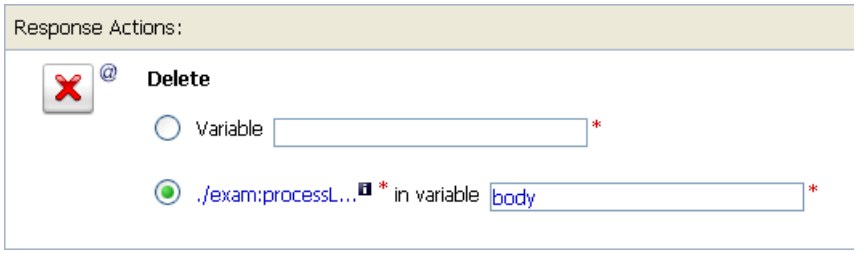
   b. Click **Validate**, then **Save**.

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 5-18.

Figure 5-18 Response Delete Action



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 Configure Message Transformation).

1. Click the **Delete** icon to bring up the menu, 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:

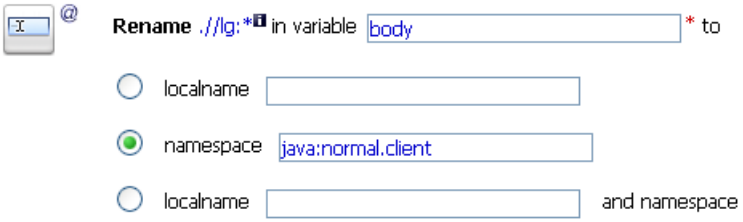
   b. Click **Validate** and **Save**. The Edit Stage Configuration page is displayed.

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 5-19.

Figure 5-19 Add Rename Action



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.

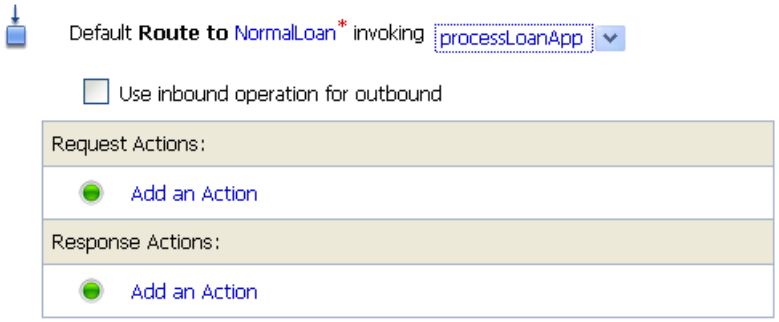
1. Click the **Case** icon  at the top of the Routing Table, then select **Insert Default Case**.

   The default case is added to the bottom of the routing table.

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 list, select the operation **processLoanApp** as shown in Figure 5-21.

Figure 5-21 Add Default Routing Case



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

5. Click **Save**.

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 Oracle WebLogic Server console. For more information see, Load the Client Application.

To test the Routing of the Loan Application Using LoanSaleProcessor Service

1. In Oracle Service Bus Console, make sure the session is activated.

2. Select **Resource Browser** > **Proxy Services** in the left panel.

3. Click the **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see Figure 4-2).

4. 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 5-22.

Figure 5-22 Response From LoanSaleProcessor Business Service

Response Document

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

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. In Oracle Service Bus Console activate the current session.

2. Click on **Resource Browser** > **Proxy Services** in the left panel.

3. Click on **Launch Test Console** icon in Actions column for LoanGateway2 to launch the test console (see Figure 4-2).

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

```
<loanRequest xmlns:java="java:normal.client">
    <!--Optional:-->
    <java:Name>Smith</java:Name>
    <!--Optional:-->
    <java:SSN>1234567</java:SSN>
    <!--Optional:-->
    <java:Rate>5</java:Rate>
    <!--Optional:-->
    <java:Amount>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 5-23.

Figure 5-23 Response from NormalLoanProcessor

The Invocation Trace indicates that the proxy service LoanGateway2 routes the application to NormalLoan 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.

6 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.
By validating the message and reporting errors.

This tutorial includes the following topics:

Prerequisites

You must complete 4 Tutorial 1. Routing a Loan Application and 5 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

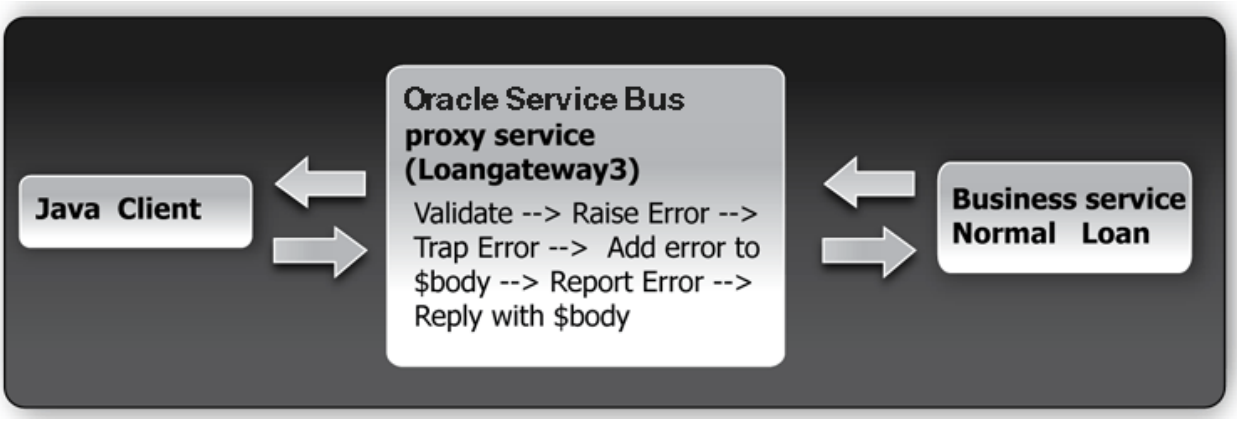
Definition of the Scenario

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 6-1 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 6-1 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.

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 4 Tutorial 1. Routing a Loan Application.

Create a Session

For this tutorial, use the MortgageBroker project folder and the directory structure you created in 4 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

4. Perform the steps in the following sections.

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 4 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 Create a Proxy Service, and use the proxy service name and the endpoint URI specified in the following table.

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

While configuring the proxy service, accept the default settings for the remaining pages. When the Create a Proxy Service - Summary page is displayed, review the summary of configuration settings before to 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

1. **Create** a new session in the Oracle Service Bus Console.

2. Select **Project Explorer**.

   The Project Explorer pane is opened in the navigation pane.

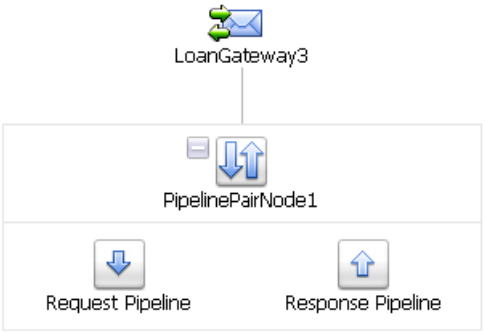
3. In the Actions column associated with the LoanGateway3 proxy service, click the **Edit Message Flow** icon.

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

4. 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 6-2.

Figure 6-2 Add Pipeline Pair



Configure the Request Actions for the LoanGateway3 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. Click the **Add an Action** link, then select **Message Processing** > **Validate** from the 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 list, select **body**.

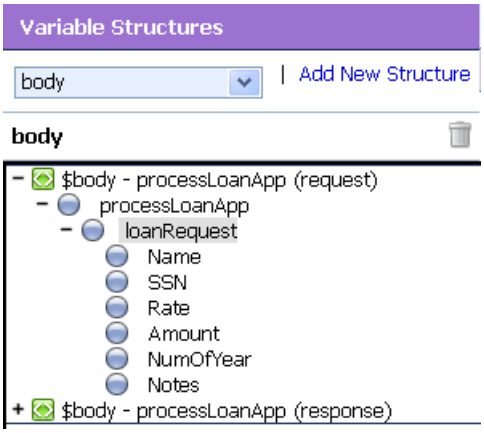
   A structural representation of the body element is displayed in the Variables Structure pane.

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

6. Expand the $body – processLoanApp (request) > processLoanApp element.

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

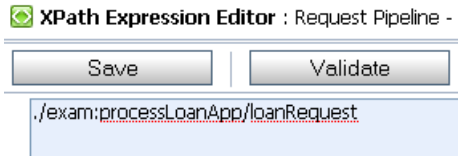
Figure 6-3 Variable Structures Pane



7. Copy the **loanRequest** element into the XQuery Expression text box.

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

Figure 6-4 XQuery Expression Editor



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

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

9. In the **variable** text field, enter **body**.

10. 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.

11. 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.

12. Select the **Raise Error on validation failure** option. The Validate action will be displayed as shown in Figure 6-5.

Figure 6-5 Configured Message Validate Action

13. 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.

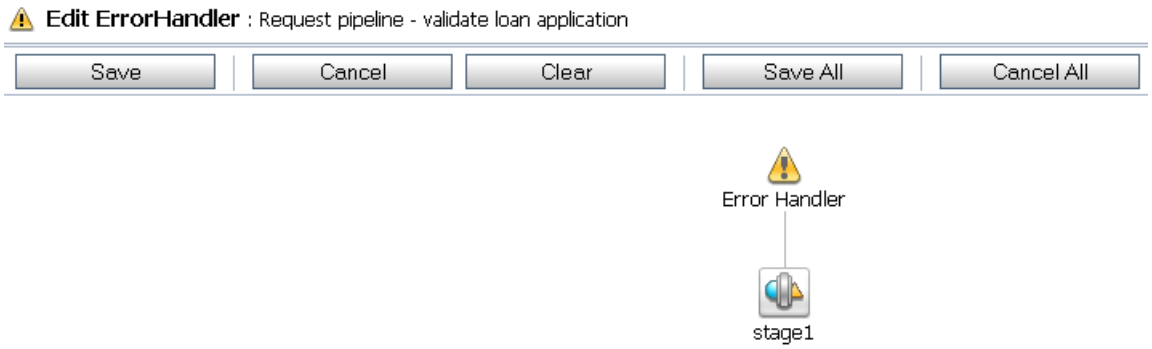
Messages can be validated at the route node or using a separate Validate action (as in this case). When you create a Validation stage, message processing is more modular since the configured validate actions logically break down the message flow. If multiple validations are performed on a message, the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

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 6-6 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

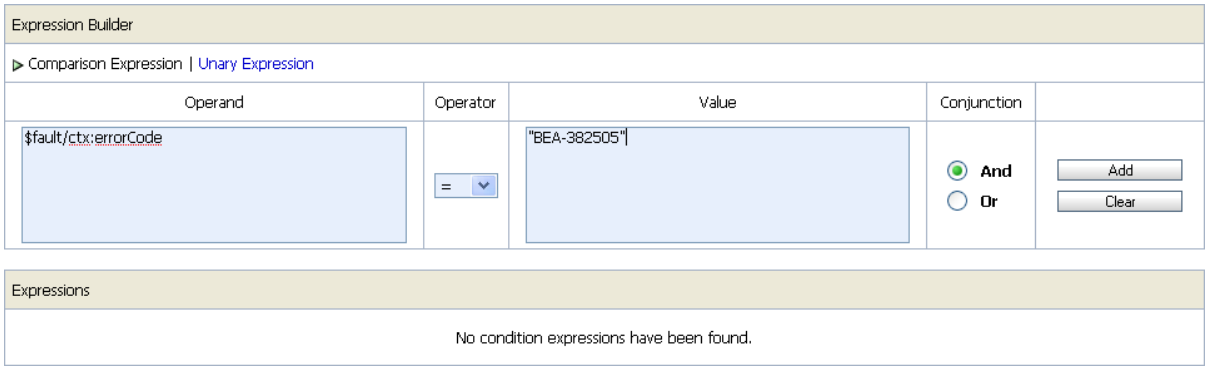
1. Select **Add an Action** > **Flow Control** > **If... Then...**

   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** link above the XQuery editor.
    b. In the Comparison Expression "Operand" pane, enter **$fault/ctx:errorCode**.
    c. Select the **=** Operator from the list.
    d. Enter **"BEA-382505"** (including double quotes) in the **Value** pane.
    e. Select the **And** option for the **Conjunction**.

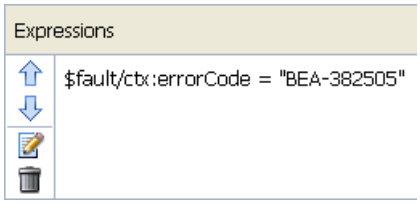
    The Expression Builder pane displays as in Figure 6-7:

Figure 6-7 Conditions in the Expression Builder



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

Figure 6-8 Expressions Textbox



4. Click **Validate** and **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** in the "then" section, and select **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 6-9.

Figure 6-9 Replace Statement

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

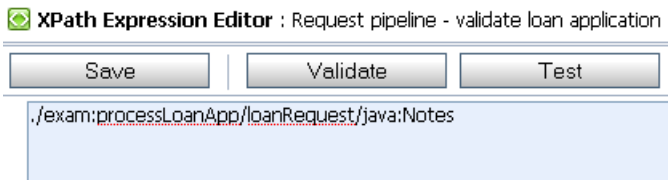
3. In the Variable Structures pane, select **body** from the Select Structure list.

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

4. Expand $body – **processLoanApp (request)** > **processLoanApp** > **loanRequest**, and copy the **Notes** element to the XQuery text editor.

   ./exam:processLoanApp/loanRequest/java:Notes

Figure 6-10 Replace Statement XPath Expression



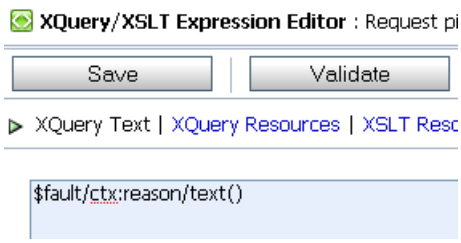
5. Click **Validate** and **Save**.

6. In the **variable** text field, enter **body**.

7. Click the **<Expression>** link to display the XQuery Expression Editor page.

8. In the XQuery Text pane, enter: **$fault/ctx:reason/text()**

Figure 6-11 Replace Statement XQuery Expression

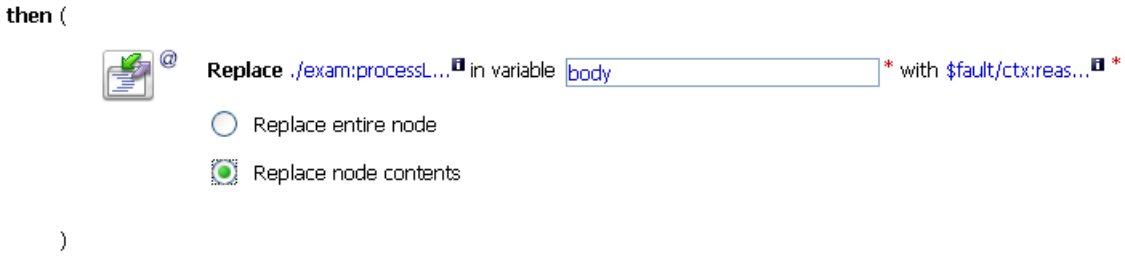


9. Click **Validate** and **Save**.

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

    The Replace action is displayed as shown in Figure 6-12.

Figure 6-12 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.

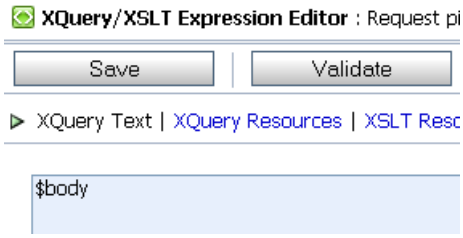
1. Click the **Replace** icon and select **Add an Action** > **Reporting** > **Report**. The report action is added to the page.

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 list.

   Copy the **$body - ProcessLoanApp (request)** element into the XQuery Expression text box.

   The $body XQuery expression is entered in the text box.



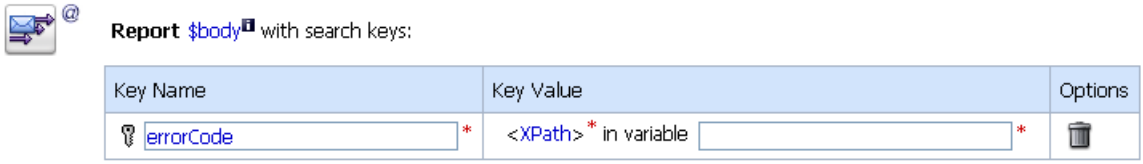
   Click **Validate**, then **Save**.

   The Edit Stage Configuration page is displayed.

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

   Enter a Key Name of **errorCode**.

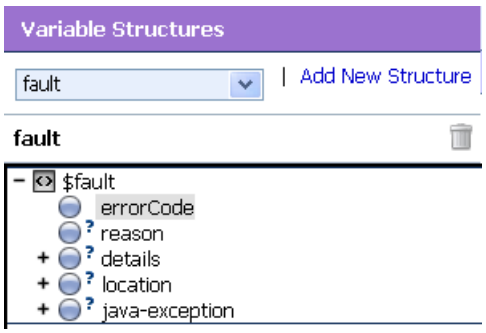
Figure 6-13 Report Action Expression Key



5. In the Key Value field, click **<XPath>** link. The XPath Expression Editor is displayed.

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

Figure 6-14 Fault Variable Structure



7. Expand **$fault** and copy the **errorCode** in the XPath Expression text box.

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

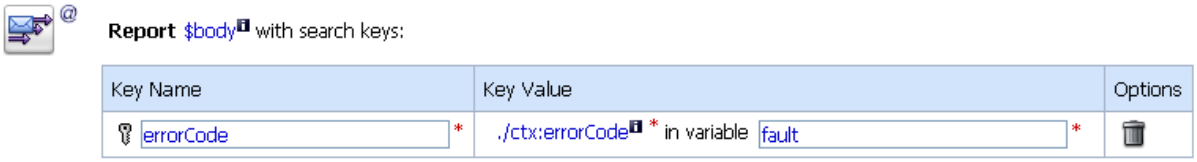
   This step enables you to index the reporting data by error code. The keys are a convenient way to identify a message. You can use the error code to search for reporting entries in the reporting module.

8. Click **Validate** and **Save**.

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

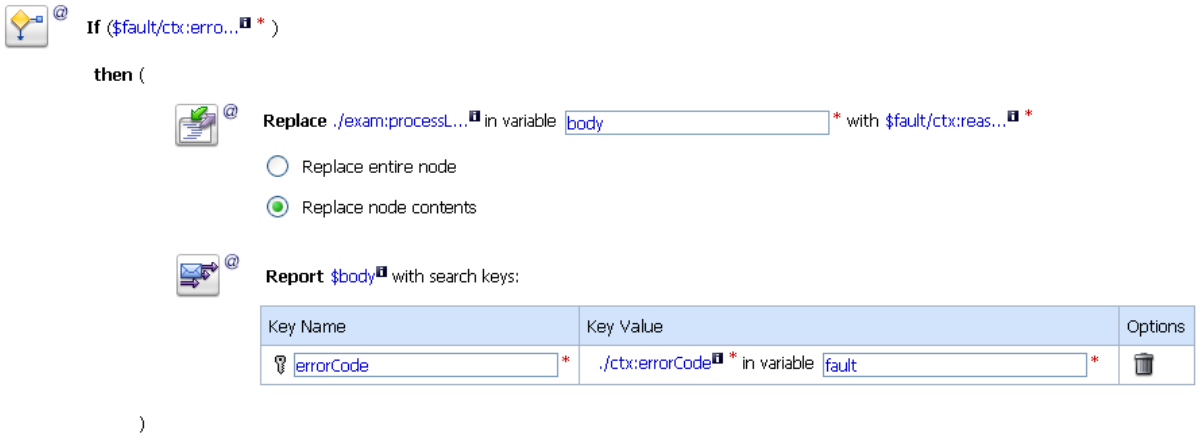
The **Report** action must be displayed as shown in Figure 6-15.

Figure 6-15 Configured Report Action



The Edit Stage Configuration page is displayed as shown in Figure 6-16.

Figure 6-16 Request Actions for LoanGateway3 Proxy Service



Add an Else Action to the Stage Error Handler

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

Add a Replace Action Within the Else Action

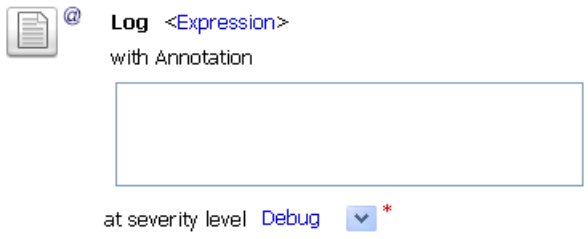
1. Select **Add an Action** > **Message Processing** > **Replace**.
2. Execute the steps in the Add a Replace Action to the Stage Error Handler section.

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 6-17 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.

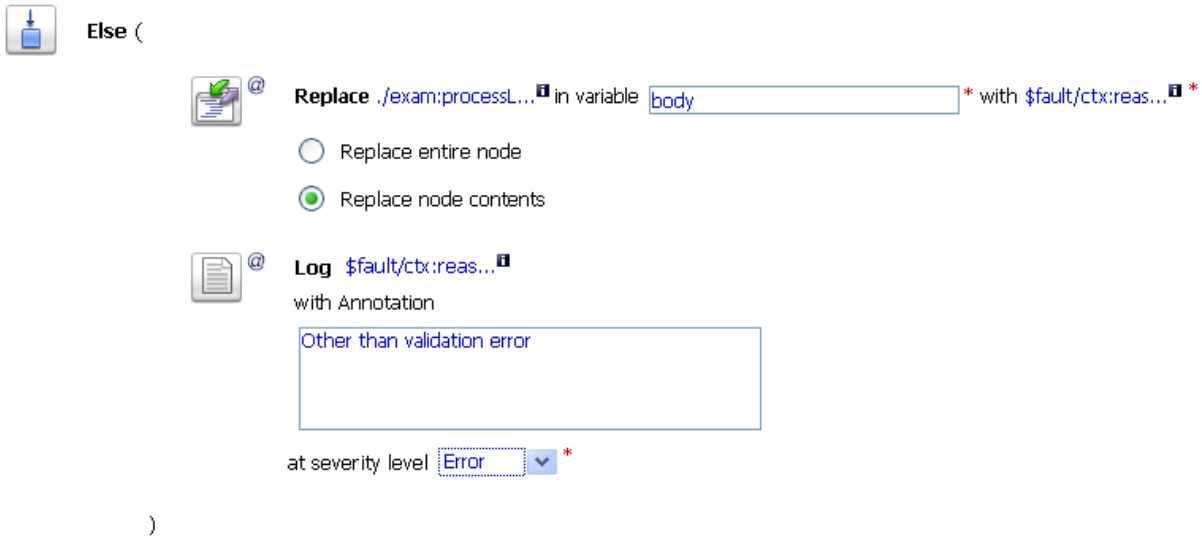
4. Click **Validate** and **Save** to validate and save the expression and return to the Edit Stage Configuration page.

5. Enter **Other than validation error** in the "with Annotation" text box.

6. Select **Error** from the "at severity level" list.

The completed **Else** condition must be displayed as shown in Figure 6-18.

Figure 6-18 LoanGateway3 Proxy Service Request Else Actions

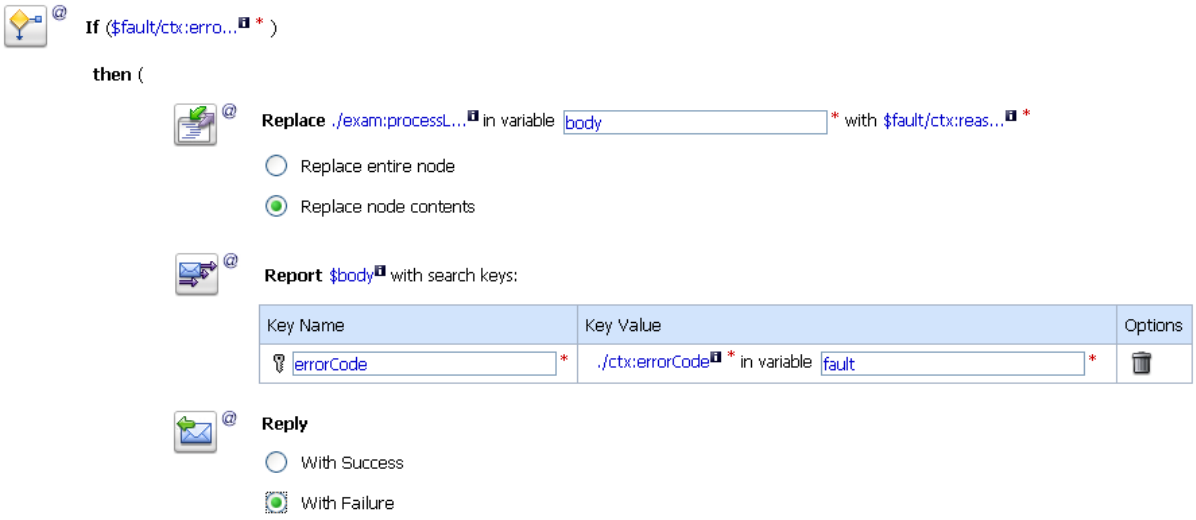


Add a Reply Action to Reply with a Failure Code

1. Click the **Report** icon under the **If...Then** action, and select **Add an Action** > **Flow Control** > **Reply**. The reply action is added to the page.

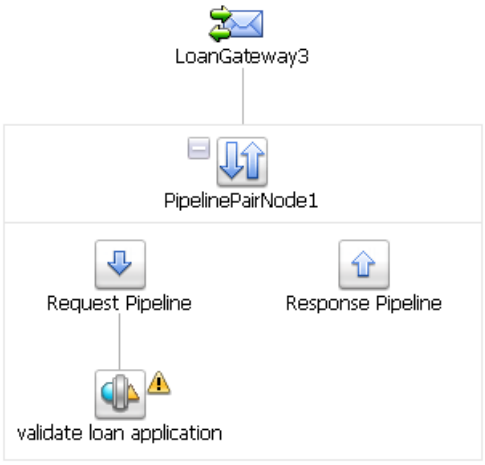
2. Select **With Failure** option. The Report Reply action is displayed as shown in Figure 6-19.

Figure 6-19 Report Reply Action with Failure Code



3. Click **Save** to save the stage configuration and return to the Edit Error Handler page.

4. Click **Save** to save the error handler configuration. You are returned to the Edit Message Flow page.



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**.

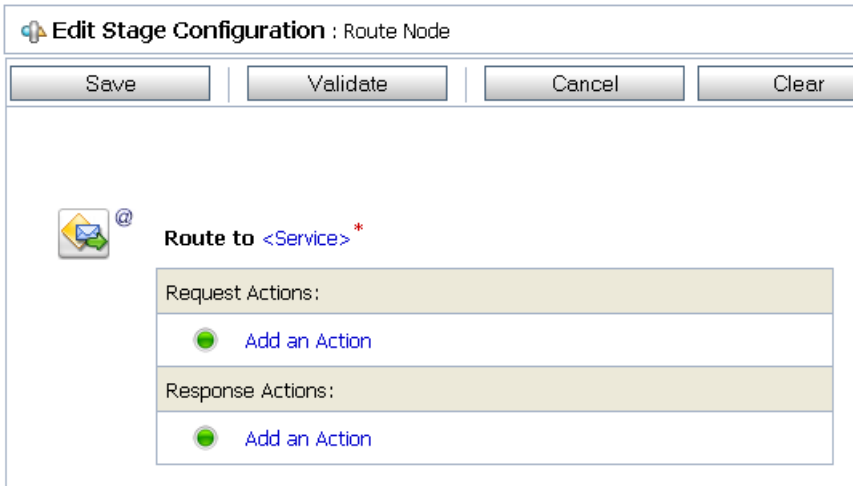
   The Edit Message Flow page is displayed.

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 6-20.

   Figure 6-20 Route to Statement



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

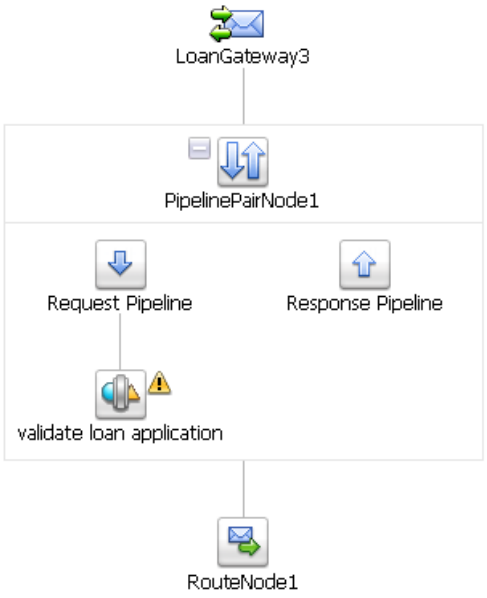
   Select the **NormalLoan** business service, then **Submit**.

5. In the Operation 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 6-21.

Figure 6-21 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). 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.

Testing Prerequisites

To test the loan application using the test console, be sure you have deployed the client JARs in the Oracle WebLogic Server console in the earlier section, Load the Client Application, in “Setting Up the Tutorials.”

To test the Validation of the Loan Application by LoanGateway3

Choose from one of the following scenarios to test the loan application.

Scenario One

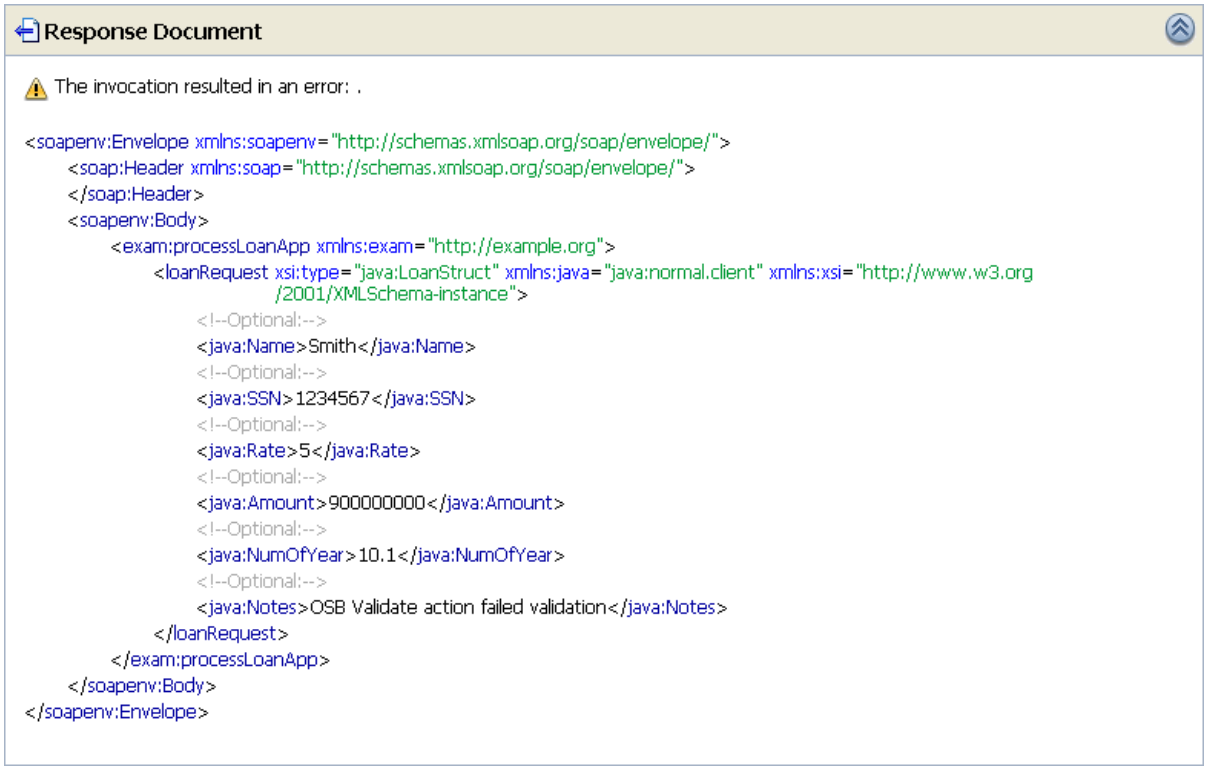
Follow these steps:

1. In Oracle Service Bus Console, make sure the session is activated.
2. Select **Resource Browser** > **Proxy Services** in the left panel.
3. Click the **Launch Test Console** icon in Actions column for LoanGateway3 to launch the test console.
4. Replace the default payload in loanRequest field with the following code:

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

Accept other default settings and click **Execute**. The response as in Figure 6-22.

Figure 6-22 Response from Error Handler



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

Scenario Two

Follow these steps:

1. In Oracle Service Bus Console, make sure the current session is activated.
2. Click on **Resource Browser** > **Proxy Services** in the left panel.
3. Click on **Launch Test Console** icon in Actions column for LoanGateway3 to launch the test console.
4. Replace the default payload in loanRequest field with the following code:

```
<loanRequest xmlns:java="java:normal.client">
    <!--Optional:-->
    <java:Name>Smith</java:Name>
    <!--Optional:-->
    <java:SSN>1234567</java:SSN>
    <!--Optional:-->
    <java:Rate>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 6-23.

Figure 6-23 Response From Normal Loan Processor

Response Document

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

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

Step 7: Using Reporting and Monitoring

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

Monitoring and Alerts

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 in the Oracle Service Bus console. 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.

You can add SLA and pipeline alerts to the services you created in the tutorials. To add alerts, be sure to **Create** a new session in the Oracle Service Bus Console. Also be sure to create a simple **Alert Destination** resource.

SLA Alerts

To add SLA alerts to services, click the service name and add alert rules on the **SLA Alert Rules** page. The aggregation interval on each alert determines how frequently alerts appear in the Dashboard after you execute services.

Pipeline Alerts

To add pipeline alerts:

1. In the **User Preferences**, change the **Dashboard Refresh Rate** to the desired rate and the **Alert History Duration** to the amount of time you want to keep pipeline alerts in the Dashboard.

2. On the **Operational Settings** page for a proxy service, **Enable Pipeline Monitoring** and set a desired **Aggregation Interval**.

3. Edit an existing proxy service message flow to add them. For example, in the LoanGateway3 message flow, edit the Reply stage in the Error Handler to add a **Reporting** > **Alert** action to a specific location.

4. Execute the proxy service to generate alerts.

Be sure to define alert rules so that when you execute the services in the Test Console, the alerts are guaranteed to fire.