Oracle® Fusion Middleware

Administrator's Guide for Oracle Service Bus

11*g* Release 1 (11.1.1.7)

E15867-10

June 2015

Documentation for administrators that describes how to manage the Oracle Service Bus (OSB) runtime environment, including service configuration, configuring message flow, message transformation with XQuery, XSLT, and MFL, import and export, monitoring, reporting, and global JNDI resource management.

ORACLE®

Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus, 11*g* Release 1 (11.1.1.7)

E15867-10

Contents

4 Working with Projects, Folders, and Resources

5 XML Schemas

6 Alert Destinations

7 WSDLs

8 JARs

9 MQ Connections

10 Split-Joins

11 XQuery Transformations

12 Custom WS-Policies

20 Proxy Services: Creating and Managing

21   Proxy Services: Actions

22 Proxy Services: Message Flow

23 Proxy Services: XQuery and XPath Editors

24 Proxy Services: Error Handlers

25 Security Configuration

26 Monitoring

27   Configuration

28   Reporting

29   Import/Export

30   UDDI

31   Global Resources

32   Customization

33   Test Console

Part II User Guide

34 Introduction to the User Guide

35 Starting Oracle Service Bus Servers

36 Configuring Proxy Services and Business Services

37   Modeling Message Flow in Oracle Service Bus

38   Improving Service Performance with Split-Join

39   Message Context

40   Using the Test Console

41 UDDI

42 Extensibility Using Java Callouts and POJOs

43 XQuery Implementation

Part III   Operations Guide

44   Introduction to the Operations Guide

45   Roles in Oracle Service Bus

46   Monitoring Oracle Service Bus at Runtime

47 Reporting

48 Managing Endpoint URIs for Business Services

49 Execution Tracing

50 Throttling

51 Managing Operational Settings Using Smart Search

52 User Preferences

Part IV Appendix

A Error Codes

B Debugging Oracle Service Bus

C Oracle Service Bus APIs

D JMX Monitoring API

E   SNMP Components

F   Monitoring Statistics in Oracle Service Bus

G   Auditing your Oracle Service Bus System

H   XQuery-SQL Mapping Reference

I   Interoperability with WSRP

J   Diagnosing Problems with Oracle Service Bus

List of Examples

List of Figures

List of Tables

Preface

This preface describes the document accessibility features and conventions of this guide--*Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide for Release 11.1.1.7

This guide has been updated in several ways. The following table lists the sections that have been added or changed.

For a list of known issues (release notes), see the "Known Issues for Oracle SOA Products and Oracle AIA Foundation Pack" at `http://www.oracle.com/technetwork/middleware/docs/soa-aiafp-knownissuesindex-364630.html`.

Sections	Changes Made	February 2013
Chapter 1 Interoperability Scenarios and Considerations		
Table 1–1, " Supported Standards and Implementations"	Section updated to reflect support for ProFTPD Server.	X
Table 1–3, " Oracle Family Platforms"	Section updated to clarify support for Oracle JCA Adapters.	X
Table 1–4, " Third-Party Platforms"	Section updated to reflect support for IBM WebSphere MQ version 7.0.1. Support for versions 5.3 and 6.0 is deprecated.	X
Chapter 9 MQ Connections		
Section 9.2, "Adding MQ Connections"	Section updated for MQ Connection Max Wait, which uses milliseconds for the unit of time.	X
Section 9.2, "Adding MQ Connections"	Section updated with the new XA Enabled option, which provides support for XA transactions in MQ connections.	
Chapter 19 Business Services: Creating and Managing		
Table 19–7, " Create/Edit a Business Service - Email Transport Configuration Page"	Section updated to include the new SSL Required option, which provides support for email communication over SSL.	X
Table 19–7, " Create/Edit a Business Service - Email Transport Configuration Page"	Section updated to include the new Socket I/O Timeout option.	X
Section 19–11, " Create/Edit a Business Service - HTTP Transport Configuration Page"	Section updated to include two new properties, Session Stickiness and Session ID Name, which support HTTP session affinity in load balancing scenarios.	X
Section 19–15, " Create/Edit a Business Service - MQ Transport Configuration Page"	Section updated to include support for XA transactions with the MQ Transport.	X

Sections	Changes Made	February 2013
Section 19–17, " Create/Edit a Business Service - SFTP Transport Configuration Page"	Section updated to include a new property to support selecting the preferred cipher suite and other algorithms.	X
Section 19.5.2, "Business Service Policies Page"	Section updated to include support for Oracle WSM policies with non-SOAP bindings that use the HTTP protocol.	X
Chapter 20 Proxy Services: Creating and Managing		
Table 20–4, " Create/Edit a Proxy Service - Email Transport Configuration Page"	Section updated to include the new SSL Required option, which provides support for email communication over SSL.	X
Section 20.5.2, "Proxy Service Policy Page"	Section updated to include support for Oracle WSM policies for non-SOAP bindings that use the HTTP transport.	X
Table 20–10, " Create/Edit a Proxy Service - JMS Transport Configuration Page"	Section updated to include support for configuring a transaction timeout period for JMS proxy services.	X
Table 20–10, " Create/Edit a Proxy Service - JMS Transport Configuration Page"	Section updated to include support for specifying a JMS client ID when using Topics in JMS proxy services.	X
Table 20–11, " Create/Edit a Proxy Service - MQ Transport Configuration Page"	Section updated to include support for XA transactions with the MQ Transport.	X
Table 20–13, " Create/Edit a Proxy Service - SFTP Transport Configuration Page"	Section updated to include a new property to support selecting the preferred cipher suite and other algorithms.	X
Appendix J Diagnosing Problems with Oracle Service Bus		
Appendix J, "Diagnosing Problems with Oracle Service Bus"	New appendix to describe the diagnostic features of Service Bus using Oracle's Diagnostic Framework and WebLogic Diagnostic Framework.	

1

Interoperability Scenarios and Considerations

This chapter lists products, standards, and technologies supported by Oracle Service Bus (OSB), including Oracle and third-party products, protocols, and Web services standards such as SOAP and WS-Security.

This section includes information about Oracle Service Bus interoperability. It includes the following topics:

- Section 1.1, "Supported Configurations"
- Section 1.2, "Supported Standards and Implementations"
- Section 1.3, "Platform Interoperability"
- Section 1.4, "Platform Interoperability Limitations"

1.1 Supported Configurations

For support information on vendor operating systems, JDK, hardware, and database support, see Oracle Fusion Middleware Supported System Configurations at http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html.

1.2 Supported Standards and Implementations

Oracle Service Bus supports the following standards and implementations.

Table 1–1   Supported Standards and Implementations

Standard/Implementation	Version
Email Servers	■ Microsoft Windows IIS SMTP Server ■ Sol/Apache SMTP Server
FTP Servers	■ Microsoft Windows IIS FTP Server ■ Sol/Apache FTP Server ■ ProFTPD Server **Note:** ProFTPD Server is certified with the SFTP Transport and JCA FTP Adapter, but not with the FTP Transport.

Table 1–1 (Cont.) Supported Standards and Implementations

Standard/Implementation	Version
Security Providers	For WebLogic Server 9.2 and later, Oracle Service Bus supports the following security providers: **Oracle WebLogic Server Security Providers** - WebLogic Server default authentication provider - WebLogic Server default credential mapper - WebLogic Server PKI credential mapper/provider - WebLogic Server Java KeyStores (JKS) - WebLogic Server default User Name Token and X509 Token handlers - WebLogic Server XACML authorization provider - WebLogic Server XACML Role Mapping provider **Oracle Platform Security Providers** - WebLogic SAML Identity Assertion Provider V2 - WebLogic SAML Credential Mapping Provider V2
Web Services	- WSDL 1.1 - SOAP 1.1 and 1.2 - SOAP w/ (MIME) Attachments (SwA) - SOAP Message Transmission Optimization Mechanism (MTOM) with XML-binary Optimized Packaging (XOP) - Universal Description, Discovery, and Integration version 3 (UDDI v3) - WS-ReliableMessaging 1.0 - WS-Addressing 1.0 - XACML 2.0 - WS-Inspection - Web Services Interoperability Basic Profile (WS-I BP) 1.1 - Web Services Interoperability Basic Security Profile (WS-I BSP) 1.0 **WS-I Basic Security Profile**: Oracle Service Bus complies with the Basic Security Profile specifications (see supported versions above) from the Web Services Interoperability Organization (WS-I) and considers it to be the baseline for Web Services interoperability. However, in some cases, Oracle Service Bus does not reject SOAP/HTTP messages that are not WS-I compliant. This enables you to build implementations with service endpoints which are not strictly WS-I compliant. When you configure a proxy service or business service, you can use the Oracle Service Bus Administration Console to specify whether you want Oracle Service Bus to enforce WS-I compliance for the service. When you configure WS-I compliance for a proxy service, WS-I compliance checks are performed when the proxy service receives a message as a response from an invoked service with a Service Callout, a route node, or on a proxy service. For information about the types of messages to which the compliance checks are applied and the nature of those checks, see Section 37.16, "WS-I Compliance."

Table 1–1   (Cont.) Supported Standards and Implementations

Standard/Implementation	Version
Web Services Security	**Security using Oracle WSM Policies** - WS-Security 1.0 and 1.1 - WS-Security: Username Token Profile 1.1 - WS-Security: X.509 Token Profile 1.1 - WS-Security: SAML Token Profile 1.1 (with SAML 1.1 in 11.1.1.3), (with SAML 2.0 in 11.1.1.4) - WS-Security: Kerberos Token Profile 1.1 - Policy Advertisement and recognition: - WS-Policy 1.2 (11.1.1.3), 1.5 (11.1.1.4) - WS-PolicyAttachment 1.2 (11.1.1.3), 1.5 (11.1.1.4) - WS-SecurityPolicy 1.1 (11.1.1.3); 1.2 and 1.3 (11.1.1.4) **Java/Oracle Platform Security Providers (Login Modules)** - SAML Login Module - SAML 2 Login Module - Kerberos Login Module - Digest Authenticator Login Module - X509 Certificate Login Module - WSS Digest Login Module - User Authentication Login Module - User Assertion Login Module **Security using WLS 9.x Policies** - WS-Security 1.0 - WS-Security: Username Token Profile 1.0 - WS-Security: X.509 Token Profile 1.0 - WS-Security: SAML Token Profile 1.0 (with SAML 1.0 and 1.1) - Policy Advertisement and recognition: - WS-Policy 1.0 and 1.2 **only** for WS Transport, which supports WS-ReliableMessaging 1.0 - WS-PolicyAttachment 1.0 and 1.2 **only** for WS Transport, which supports WS-ReliableMessaging 1.0 - WS-SecurityPolicy - WebLogic Server-proprietary format that is based on the assertions described in the December 18, 2002 draft version of the Web Services Security Policy Language (WS-SecurityPolicy 1.0) specification; and 1.2 **only** for WS Transport, which supports WS-ReliableMessaging 1.0, to define Transport-level Security assertions **only**.
HTTP	1.01.1
EJB	- 2.1 - 3.0
SNMP	- SNMPv1 - SNMPv2c

Table 1–1   (Cont.) Supported Standards and Implementations

Standard/Implementation	Version
WebLogic JMS	WebLogic Server ■ 8.1 SP4-SP6 ■ 9.0, 9.1, 9.2 ■ 10.0 ■ 10.3.x
Third-party JMS	Any JMS provider that implements the JMS specification is supported through Oracle WebLogic Server as a foreign JMS provider.
Microsoft .NET 1.1 with SOAP 1.1	**Style-encoding: document-literal, rpc-encoded** ■ Oracle Service Bus supports document-literal and interoperates with .NET services. ■ Oracle Service Bus interoperates with .NET rpc-encoded services in cases of inbound and outbound (routing/publish). In these cases, interoperability is possible regardless of parameter types. ■ Oracle Service Bus Service Callouts may fail to interoperate with .NET rpc-encoded services. **Note**: DIME attachments is not supported by Oracle Service Bus. **.NET 1.1 Security Configurations Limitations** The following security configurations in the .NET 1.1 framework are not interoperable with the Oracle Service Bus message-level security: ■ Signing the message body from WebLogic to .NET WSE 2.0 (Webservices Security Extension) is interoperable. However, by default, WSE requires additional headers-for example, `WS-Addressing` and `timestamp`. Therefore, to make Oracle Service Bus message-level security for .NET Web services interoperable, you must remove all of the message predicate other than the message body from .NET security policy configuration ■ To ensure Oracle Service Bus interoperability with .NET, the *replay detection attribute*, `<replayDetection>`, must be set to `disabled` on the .NET side. See also Section 1.4.1, ".NET Interoperability Limitations."
Microsoft .NET 2.0 with SOAP 1.1 and SOAP 1.2	See Section 1.4.1, ".NET Interoperability Limitations."
Microsoft .NET 3.0 with SOAP 1.1 and SOAP 1.2	See Section 1.4.1, ".NET Interoperability Limitations."
Microsoft .NET 3.5 with SOAP 1.1 and SOAP 1.2	See Section 1.4.1, ".NET Interoperability Limitations."
WebLogic JMS Client for Microsoft .Net (for .Net C# client applications)	See *Oracle Fusion Middleware Using the WebLogic JMS Client for Microsoft .NET for Oracle WebLogic Server*.

1.3 Platform Interoperability

See the Oracle Service Bus Release Notes for the latest information about patches or updates that may be required to support your interoperability scenarios.

Oracle Service Bus interoperates with the platforms described in the following tables.

Table 1–2   Oracle WebLogic Family Platforms

Interoperability	Version
Oracle WebLogic Server	■ 10.3.5 (runtime only)
	■ 10.3.6
Oracle SOA Suite	■ 11.1.1.3 and later
WS-* and JMS interoperability with WebLogic Platform	■ 8.1 SP4-SP6 (except WS-Security
	■ 9.0, 9.1, 9.2 (except WS-Security)
	■ 10.0 (except WS-Security)
	■ 10.3.x
Web Services for Remote Portlets (WSRP) with Oracle WebLogic Portal	■ 9.2
	■ 10.0
	■ 10.2
	■ 10.3.x
Oracle WebLogic Portal	■ 8.1 SP6
	■ 9.2
	■ 10.0
	■ 10.2
	■ 10.3
WebLogic Integration	■ 8.1 SP6
	■ 9.2
	■ 10.2
	■ 10.3
MQ event generator and control in WebLogic Integration	■ 8.1 SP4 or later
	■ 9.2

Table 1–3   Oracle Family Platforms

Interoperability	Version
Oracle Service Bus	■ 2.6
	■ 2.6 RP1
	■ 3.0
	■ 10gR3 (10.3)
	■ 10.3.1
	■ 11.1.1.3 and later
Oracle Enterprise Pack for Eclipse (OEPE)	11.1.1.8.0 on Eclipse 3.7.1
Oracle Enterprise Repository	■ 2.6
	■ 3.0 RP1
	■ 10.3
	■ 11.1.1.3 and later
Oracle Service Registry	■ 3.0
	■ 11.1.1.3
Oracle Enterprise Security	■ 3.0
	■ 10.3

Table 1–3 (Cont.) Oracle Family Platforms

Interoperability	Version
Oracle Web Services Manager	■ 10.1.3.x and later ■ 11.1.1
Oracle BPEL Process Manager	■ 10.1.3.4.x and later
Oracle JDeveloper (for developing adapter artifacts)	■ 11.1.1.3 and later
Oracle JCA Adapter for AQ[1][2]	■ 11.1.1.3 and later
Oracle JCA Adapter for Database[1][2]	■ 11.1.1.3 and later
Oracle Adapter for Oracle Applications[1][2]	■ 11.1.1.3 and later
Oracle JCA Adapter for Files/FTP[1][2]	■ 11.1.1.3 and later
Oracle JCA Adapter for Sockets[1][2]	■ 11.1.1.3 and later
Oracle BAM Adapter (Business Activity Monitoring)[1][2]	■ 11.1.1.3 and later
PeopleSoft (Oracle Application Adapters)	■ 11.1.1.4 and later
SAP R/3 (Oracle Application Adapters	■ 11.1.1.4 and later
Siebel (Oracle Application Adapters	■ 11.1.1.4 and later
J.D. Edwards (Oracle Application Adapters	■ 11.1.1.4 and later
Oracle Data Service Integrator	■ 2.5 ■ 3.0 (on WebLogic Server 9.2) ■ 3.01 (on WebLogic Server 9.2) ■ 3.2 (on WebLogic Server 10.0.1) ■ 10*g* Release 3
Oracle Tuxedo/WebLogic Tuxedo Connector	■ 8.1 ■ 9.x ■ 10.0

[1] You must develop these adapters using Oracle JDeveloper 11.1.1.3 or later with Oracle SOA plug-ins installed.

[2] These adapters are installed by default with Oracle Service Bus. Other adapters must be downloaded and installed separately from Oracle Software Delivery Cloud and require additional licensing.

Table 1–4 Third-Party Platforms

Interoperability	Version
IBM WebSphere MQ	■ 7.0 ■ 7.0.1.8 ■ 7.5 **Note**: You must use the MQ 7.0 client libraries when using MQ 7.5 with Service Bus 11.1.1.7. See Section 9.2, "Adding MQ Connections" for information about adding the MQ libraries to the classpath. Supported with SOAP 1.1, not SOAP 1.2. See Section 1.4.3, "WebSphere Interoperability Limitations." **Note**: Versions 5.3 and 6.0 are no longer supported by IBM. These versions are deprecated in Oracle Service Bus.
IBM WebSphere EJB/RMI	6.0

Table 1–4 (Cont.) Third-Party Platforms

Interoperability	Version
IBM WebSphere WS	6.1 (Fixpack 15)
	Supported with SOAP 1.1, not SOAP 1.2.
	See Section 1.4.3, "WebSphere Interoperability Limitations."
Tibco Enterprise Message Service	All versions that meet the JMS 1.2 specification through Oracle WebLogic Server
Apache Axis	■ 1.2.1
	■ 1.4.1
	Supported with SOAP 1.1, not SOAP 1.2.
	See Section 1.4.2, "Apache Axis Interoperability Limitations."
BMC Patrol	`http://www.softwareinnovations.co.uk/products/patrol.html`

1.4 Platform Interoperability Limitations

This section describes interoperability limitations with different platforms.

1.4.1 .NET Interoperability Limitations

- .NET clients that need to communicate with Oracle Service Bus using basic authentication must send the authentication information in the first request. Otherwise, the invocation fails because Oracle Service Bus does not challenge the .NET client for credentials.

- Oracle Service Bus interoperability with .NET using Basic Authentication works successfully when configured with Windows 2003/IIS 6.0; however, interoperability with .NET using Basic Authentication on Windows XP/IIS 5.1 is not supported.

- Message-level security interoperability for .NET clients works only with SOAP 1.1. The WSE Soap Protocol Factory does not support security with SOAP 1.2. See "Message-Level Security with .Net 2.0" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

1.4.2 Apache Axis Interoperability Limitations

Unresolved references when you import RPC-encoded Axis-generated WSDLs

When you import an RPC encoded WSDL, generated by Axis, into Oracle Service Bus, you may experience a warning message indicating that the WSDL contains references that must be resolved.

If you open the structural view of the imported WSDL in the **View a WSDL** page in the Oracle Service Bus Administration Console, unresolved schema imports are displayed in the Imports section.

This issue does not affect your ability to use the WSDL in the Oracle Service Bus environment. You can eliminate the warning by removing unresolved schemas from the WSDL file.

SOAPAction attribute in Axis-generated WSDLs initialized to empty string

The WSDL generated by Axis have the SOAPAction attribute initialized to an empty string. Configuring an Oracle Service Bus business service with this WSDL, causes invocations to this web service to fail generating a "No SOAPAction" fault.

To work around the issue and ensure successful web service invocations from Oracle Service Bus to Axis, you must configure a transport header in the proxy service message flow Specifically, you must add a **Set Transport Headers** request action in the message flow route and enable the **Pass all headers through Pipeline** option.

This issue also causes invocations from the Oracle Service Bus Test Console to fail (and generates a "No SOAPAction" fault) even when the workaround is in place. To make Test Console invocations work, you must set the SOAPAction HTTP header in the **Set Transport Header** request action in the message flow route.

HTTP response and status code for one-way operations

For both document literal and RPC encoded types of web services, on invocation of a one-way operation, Axis is expected to send an empty HTTP response with status code 202 OK to the client. However, Axis sends an non-empty HTTP response with status code 200 OK. The body of this HTTP response contains an empty SOAP envelope.

This causes the Oracle Service Bus proxy or business service to send the same 200 OK response code to their clients violating the expected results.

HTTP response and status code for nne-way operations generating a fault

For both document literal and RPC encoded types of web services, on invocation of a one-way operation generating a fault, Axis is expected to send an empty HTTP response with status code 202 OK to the client. However, Axis sends a non-empty HTTP response with status code 500 Internal Server Error with an empty SOAP envelope as a body.

This causes the Oracle Service Bus proxy or business service to send the same 500 Internal Server Error response to their clients violating the expected results.

1.4.3 WebSphere Interoperability Limitations

HTTP response and status code for one-way operations

For both document literal and RPC encoded types of web services, on invocation of a one-way operation, WebSphere is expected to send an empty HTTP response with status code 202 OK to the client. However, WebSphere sends an non-empty HTTP response with status code 200 OK. The body of this HTTP response contains an empty SOAP envelope.

This causes the Oracle Service Bus proxy or business service to send the same 200 OK response code to their clients violating the expected results.

Part I

Console Help for Oracle Service Bus

This part provides the topics of the Oracle Service Bus Administration Console online help system for managing Oracle Service Bus at runtime. Chapters include:

2

Introduction to Oracle Service Bus

This chapter provides a brief introduction to Oracle Service Bus (OSB) and includes key getting started information such as starting the OSB Administration Console, customizing table views in the OSB Administration Console, and naming OSB resources.

Oracle Service Bus is a configuration-based, policy-driven enterprise service bus. It provides highly scalable and reliable service-oriented integration, service management, and traditional message brokering across heterogeneous IT environments. It combines intelligent message brokering with routing and transformation of messages, along with service monitoring and administration in a unified software product. Oracle Service Bus Administration Console enables you to control the service and policy configurations, and monitor system and operations tasks. Oracle Service Bus relies on Oracle WebLogic Server runtime facilities. For more information, see *Oracle Fusion Middleware Concepts and Architecture for Oracle Service Bus*.

2.1 Starting the Oracle Service Bus Administration Console

1. Start the Oracle Service Bus Admin server, as described in Section 35, "Starting Oracle Service Bus Servers."

2. When the server starts, enter the following URL in your browser:

   ```
   http://hostname:port/sbconsole
   ```

   where host represents the name of the computer on which WebLogic Server is running and port represents the port number.

   For example, if WebLogic Server is running on your local computer using the default port configuration, enter the following URL in your browser:

   http://localhost:7001/sbconsole

   The Oracle Service Bus Administration Console login page is displayed.

3. Enter the user name and the password you specified during the installation process.

   Note: The username and password for the Oracle Service Bus Examples domain is weblogic / welcome1.
   ---

4. Click **Login**.

To log out, click **Logout** on the banner near the top of the right panel of the Oracle Service Bus Administration Console.

2.2 Customizing Table Views

Use the table customizer to display table information according on your specified settings. The default sort order for any table is determined by the first column in the table.

1. Click the **Table Customizer** icon. Additional fields are displayed.

    You can click **Cancel** at any time to close this table and retain the original settings.

2. In the **Columns** field, select the columns you want to display:

    a. Select a column name from the **Available Columns** field.

    b. Click the arrow to move this column name to the **Selected Columns** field.

        An asterisk denotes a required column.

    c. Repeat until you have listed all the column names you want to display in the **Selected Columns** field.

        In the **Selected Columns** field, you can use the Up and Down arrows to reorder the column names as required.

3. To remove columns from being displayed, move the column names from the **Selected Columns** field to the **Available Columns** field.

4. From the **Rows** list, select the number of rows you want to display on a single page.

5. Do one of the following:

    -   To save these settings and close the table customizer, click **Apply**.

    -   To discard your changes and retain the original settings, click **Reset**.

2.3 Resource Naming Restrictions

When naming any directory or resource in an Oracle Service Bus configuration, the following characters are allowed:

-   All Java identifier characters, including Java keywords, as described in the "Identifiers" and "Keywords" sections of the *Java Language Specification* at http://docs.oracle.com/javase/specs/jls/se5.0/html/j3TOC.html

-   Blanks, periods, and hyphens within the names (not leading or trailing)

Characters such as / \ * : " < > ? | are not allowed.

3

Change Center

This chapter provides instructions on using the Change Center in the Oracle Service Bus Administration Console. Procedures include activating sessions, viewing configuration changes, undoing configuration tasks, and resolving conflicts.

The Change Center allows for team collaboration when services and metadata are being configured in Oracle Service Bus. Each user works in a sandbox session until the user is ready to check in the working configuration to the core configuration of the bus. The change center provides multiple levels of undo, and visibility into conflicts, as multiple users work on the configuration.

Most of the Change Center pages display different information based on whether you are in a session (for example, if you have clicked **Edit** in the Change Center) or outside a session. For example, in a session, the **View Configuration Changes** page that appears when you click **View Changes** in the Change Center lists all the changes you have made in that session; outside a session, the page lists all session activations.

Table 3–1 displays Change Center tasks and related session information.

Table 3–1   Change Center Tasks and Session Information

Tasks	Session Information
View conflicts	**Inside session only**: This link displays the number of conflicts occurring in a session. View and resolve all conflicts between changes in the current session and those made by all other sessions within the Oracle Service Bus Administration Console. A conflict occurs if a resource modified in the current session has already been modified and activated by another session. Two changes to the same resource by two sessions do not cause a conflict until one of the sessions is activated. This page also displays semantic errors for resources in the session.
View changes	**Inside session**: View the configuration changes you have made during the current session. **Outside session**: View configuration changes that are caused by previous session activations.
Undo changes	**Inside session**: Undo a change you have made during the current session. **Outside session**: Undo an activation of a session.
View all sessions	**Inside and outside session**: View all existing sessions within the Oracle Service Bus Administration Console.
Activate sessions	**Inside session only**: Activate the session.

Table 3–1 (Cont.) Change Center Tasks and Session Information

Tasks	Session Information
View task details	**Inside session**: View details of a specific change you made in the current session.
	Outside session: View details of specific changes made by a previous session activation.

3.1 Using the Change Center

The Change Center module is the starting point for using the Oracle Service Bus Administration Console to make changes to your configuration. To make configuration changes using the Administration Console, you must use the Change Center to start a session.

1. Click **Edit** or **Create** to begin a session. The name of the session is displayed under **Change Center**.

2. Make the appropriate changes on the relevant page of the Administration Console.

3. Click **Save** on each page where you make a change. All the changes you have made in the current session are saved.

   Click **Discard** under **Change Center**, at any time during the session, to discard the changes you have made so far in the current session.

4. When you have finished making changes, click **Activate** under **Change Center**. Continue in Section 3.2, "Activating Sessions."

3.2 Activating Sessions

To activate a session:

1. Click **Activate** in the **Change Center**. The **Activate Session** page shows the session name, user name, and a description field.

   Creating a session and discarding a session proceed regardless of other activity in the system. However, if another session is in the process of being activated, the **Activate Session** page displays an error indicating the user that has the pending WebLogic Server changes. For information on resolving conflicts between changes made in your session and other activated sessions, see Section 3.9, "Viewing and Resolving Conflicts."

2. If there are validation errors, an error message is displayed, indicating that your session has validation errors. View and fix configuration conflicts before you proceed. For information on fixing conflicts, see Section 3.9, "Viewing and Resolving Conflicts."

   You will not be able to activate the session until you have viewed all conflicts. If new conflicts arise while you view the existing conflicts, before you activate, a message pops up informing you of the new conflicts.

3. Enter a description in the **Description** field. This description is displayed in the **Description** column when you click **View Changes** under **Change Center** to display configuration changes caused by session activations.

4. To activate the session, click **Submit**. If no new conflicts have arisen in the interim, the session ends and the configuration is deployed to the runtime.

> **Note:** When you attempt to activate a session with a JMS endpoint URI on another server (a single server other than the one on which you are working, or a Managed Server in a cluster), ensure that the destination server is available.
>
> Oracle Service Bus does not allow registration of proxy services with JMS transport if the JMS endpoint URL specifies a destination that is unreachable. In other words, for JMS services, Oracle Service Bus checks if the specified connection factory exists; if it does not, a session activation error occurs.

3.3 Exiting Sessions

Click **Exit** under **Change Center** at any time to exit the session. However, the session does not end.

You can click **Edit** under **Change Center** to return to the session and continue making changes. This behavior also applies if you click **Logout** to log out of the Administration Console or close your browser. The session and all changes that you have made in the session persist even if you log out of the Administration Console or the server is restarted.

The session ends only after it has been activated. See Section 3.2, "Activating Sessions."

3.4 Viewing Configuration Changes

This page displays different information based on whether you are in a session (that is, you have clicked **Edit** in the **Change Center**) or outside a session.

- When you are in a session, the View Configuration Changes page displays a list of configuration changes that you have made during the current session. See "Viewing Configuration Changes in a Session" in this section.

- When you are outside a session, the View Configuration Changes page displays a list of configuration changes that were caused by previous session activations. See "Viewing Configuration Changes Caused by Session Activations" in this section.

Viewing Configuration Changes in a Session

1. If you have not already done so, click **Create** to create a new session for making changes to the current configuration. See Section 3.1, "Using the Change Center."

2. Make at least one change to the configuration.

3. Click **View Changes** under **Change Center**. For each change you have made to the configuration during the current session, the **View Configuration Changes** page displays the information shown in Table 3–2.

Table 3–2 In-Session Configuration Changes

Property	Description
Task	A description of the task that was implemented. The task is a link to the Task Details page. See Section 3.7, "Viewing Task Details."
Execution Time	The date and time that the task was executed.
User	The name of the user who implemented the change.

Table 3–2 (Cont.) In-Session Configuration Changes

Property	Description
Task Status	The status of the task:
	Completed—the task was completed.
	Undone—the task was undone.
	Undo Completed—the undo was completed.
Undone By	The name of the user who undid the task. This field will contain `None` if the task has not been undone.
Options	Click the **Undo** icon to reverse the execution of the task.
	When you are working in a session, you can undo tasks in any order.
	As a result of undoing a task, the object of the task reverts to the state it was in before the task in question was performed. Note, however, that any tasks that were performed on the same object after the task that you undo are also undone. See Section 3.6, "Undoing Tasks."

Viewing Configuration Changes Caused by Session Activations

1. Make sure you are not in a session. See Section 3.3, "Exiting Sessions."

2. Click **View Changes** under **Change Center**. For each session you have previously activated, the **View Configuration Changes** page displays the information shown in Table 3–3.

Table 3–3 Activated Configuration Changes

Property	Description
Task	A description of the session that was activated. The task is a link to the Task Details page, which displays the operations that were performed in the session. See Section 3.7, "Viewing Task Details."
Description	An additional description of the session that was activated, if a description was entered when the session was activated using the Activate Session page.
Execution Time	The date and time that the session was activated.
User	The name of the user who activated the session.
Task Status	The status of the session:
	Activated—the session was activated.
	Partially Activated—displayed during session activation if one or more servers in a cluster are inaccessible, thus preventing activation of the session on those servers. When the servers become accessible, any unactivated changes will then be activated on them.
	Undone—the previously activated session was undone and all the operations performed in the session were discarded.
	Undo Activated—the undo was activated.
	In Progress—displayed if a session activation is in progress, as session activations can take a long time.
	Failed—displayed if a session activation fails. Oracle Service Bus tracks session activation failures but not failures due to individual updates inside a session.
Undone By	The name of the user who undid this task. This field will contain `None` if the session activation has not been undone.

Table 3–3   (Cont.) Activated Configuration Changes

Property	Description
Options	Click the **Undo** icon to reverse the session activation and the operations performed in that session. You can undo session activations in any order.
	Click the **Undo into Session** icon to create a new session pre-populated with the tasks needed to undo the changes activated in the session. See Section 3.6.2, "Undoing Into a Session."

3.5 Purging Session Activation History

You can purge sessions activated over a specific period, delimited by start and end dates. This action can only be performed outside a session.

> **Caution:** Purging session activation history involves deleting data that enables multiple levels of undo. In other words, if you purge session activation history for a specific period, you will not be able to undo sessions activated during that period.

1. Ensure that you are not in an active session, and click **View Changes**.

2. On the View Configuration Changes page, click **Purge By Date**. The Purge Session Activation History by Date page is displayed.

3. Enter the Start and End dates, in their respective fields, in the format: MM/DD/YYYY.

    The session activation history for all sessions between the dates specified will be deleted from Oracle Service Bus.

3.6 Undoing Tasks

Use the View Configuration Changes page to undo tasks that you have performed in your Oracle Service Bus configuration during your current session, and to undo session activations outside a session. Oracle Service Bus lets you undo multiple levels of session activation, constrained only by your system resources. For more information, see Section 3.6.1, "Order of Undoing Tasks."

> **Note:** If you upgrade from Oracle Service Bus version 2.1 to version 2.5, you will not be able to undo sessions activated before the upgrade. Sessions activated after the upgrade will be available in the session activation history, for undo.

Undoing a Task in a Session

1. Make sure you are in a session.

2. Click **View Changes** under **Change Center**.

3. In the **Options** column for a specific task, click the **Undo** icon.

    The task is undone.

    You can undo any change in the current session. However after that, if you change your mind, you can undo the undo, or if you again change your mind, you can undo the undo that undid a previous undo, and so on.

Undoing a Session That Was Activated Earlier

1. Make sure you are outside a session.

2. Click **View Changes** under **Change Center**.

3. Do one of the following:

   - In the **Options** column for a specific task, click the **Undo** icon. The session activation is undone.

   - In the **Options** column for a specific task, select the **Undo into Session** icon. A new session is created pre-populated with the tasks needed to undo the changes activated in the session. For more information, see Section 3.6.2, "Undoing Into a Session."

3.6.1 Order of Undoing Tasks

You can undo tasks in any order (provided that individual undo actions result in valid data). The undo operation sets the value of a resource to the value it had before the change to that resource.

In the case that the task that is being undone was one that created an object, there is no previous state to which an object can be returned—in other words, no object existed before this task was performed. Effectively, the undo operation deletes the new object from the session. In this case, errors occur for the objects that reference the one being deleted. You can view such errors on the View Conflicts page in the Change Center.

When you are not working in a session, you can access the View Configuration Changes page to see the sessions that were previously activated. You can undo these sessions. The system does not allow you to undo a session that was previously activated if an error in the runtime configuration would result from the undo action. For example, if you attempt to undo a session activation that results in the removal of an object that is being referenced by another object, that undo action is disallowed.

It is possible to undo an undo action. In the **Options** column of the **Undo of [task]**, click the **Undo** icon. Oracle Service Bus supports unlimited undo. This means you can undo the undo that undid a previous undo, and so on.

3.6.2 Undoing Into a Session

If semantic errors result from undoing a session activation, you are prevented from doing the undo. However there is an alternative in this case. You can undo the session activation and have the changes put into a new session. You can then fix the semantic errors and activate the session. You can also use this capability of undoing into a session to explore the ramifications of a session activation undo. You can examine all the changes that result, and decide if you really want to do the undo. If you decide that you really want to, you can activate the session.

3.7 Viewing Task Details

Use this page to view details of a specific change you made in the current session if you are in the session, and view details of specific changes you made in sessions that have been activated if you are outside a session.

1. Select **View Changes** under **Change Center**.

2. On the View Configuration Changes page, click the name of the task in the **Task** column. The Task Details page displays the information shown in Table 3–4.

Table 3–4   Task Details

Property	Description
Task	The type of task you performed.
	■ Create
	■ Update
	■ Delete
	■ Rename
	■ Move
Resource Type	The Oracle Service Bus resource types that were part of the task.
Resource	The name and path of the resource. The path is the name of the project and folder in which the resource resides.

3. Click **Back** to return to the View Configuration Changes page.

3.8 Viewing All Sessions

Use this page to view all existing sessions within the Oracle Service Bus Administration Console. You can view these sessions if you are currently in a session or outside a session.

You can view all sessions *only* if you are using the Administrator role. For more information, see "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

1. Click **View All Sessions** under **Change Center**. For each existing session, the View All Sessions page displays the information shown in Table 3–5.

Table 3–5   Session Details

Property	Description
Session Name	The name of the session. The name is a link to the session.
User	The name of the user who created the session.
Creation Time	The date and time the session was created.
Last Modified	The date and time a change was last made during the session.
# of Objects Modified	The number of objects that have been changed during the session.

2. To switch to another session, click the link in the **Session Name** column for that session.

> **Note:**   The same user logged in with multiple browsers is not supported. It causes unpredictable behavior in the Administration Console.

3.9 Viewing and Resolving Conflicts

Use this page to view diagnostic messages about errors in your configuration, and to view and resolve conflicts between changes made in your session and other activated sessions. The view conflicts link also displays the number of live conflicts in the session.

Viewing Conflicts

1. Click **View Conflicts** under **Change Center**. The View Conflicts page can display any or all of the following sets of information depending on the nature of the conflicts:

   - Errors (an Error icon denotes non-committable, critical conflicts)—inform you of critical conflicts within your configuration. You cannot commit your changes without resolving the conflicts. See Table 3–6 and the next topic, "Resolving Conflicts."

   - Concurrent Updates (a Warning icon denotes committable, non-critical conflicts)—warn you about incompatible changes with other activated sessions. See Table 3–7.

   - Informational Messages (a Warning icon denotes committable, non-critical conflicts)—inform you of any non-critical conflicts within your configuration.

Table 3–6   Errors

Property	Description
Name	The resource to which the error message refers. The name is a link to that resource.
Path	The project and folder in which the resource resides.
Resource Type	The resource type.
Messages	A description of the conflict. To resolve the conflict, see "Resolving Conflicts" in the following section.

2. To display details of a specific conflict, click the link in the **Name** column for that conflict.

   If the object to which you make changes in a session has changed in the runtime since you began the edit of the current session, the View Conflicts page displays the information shown in Table 3–7.

Table 3–7   Concurrent Updates

Property	Description
Name	The resource in conflict. The name is a link to that resource.
Your Changes	A description of the changes you made to this object in your session.
Other's Changes	A description of the changes another user made to this object in their session.
Synchronize	Click the **Synchronize** icon to return this object to the state in which it is saved in the runtime.

Resolving Conflicts

To resolve a conflict, use the information provided in the Messages column (as described in Table 3–6) to understand the problem, and then edit the object that is causing the conflict to fix the problem.

For the scenario in which you have a concurrent update conflict—that is, a conflict that occurs if a resource is modified in the current session that has already been modified and activated by another session (as described in Table 3–7), you can resolve the conflict in one of two ways:

- Click **Activate** under Change Center. This saves your changes to the runtime, which means you override changes that are deployed to core data by a previous session activation. In other words, the changes saved by the other user in another session are overwritten by your changes in this session.

- Click the **Synchronize** icon in the **Synchronize** column of the table (as described in Table 3–7). This action restores this object, in this session, to the state in which it was saved in the runtime.

4

Working with Projects, Folders, and Resources

This chapter contains procedures for working with Oracle Service Bus (OSB) projects, folders, and resources using the OSB Administration Console, including importing resources from ZIPs, JARs, and URLs.

The Project Explorer organizes configurations and resources into projects and folders. In the Project Explorer, you can display the Projects View Page, which lists all projects in the domain, and Project/Folder View Pages, which list folders and resources in each project.

Projects are non-hierarchical, non-overlapping, top-level grouping constructs.

All Oracle Service Bus resources (for example, services, WS-Policies, WSDLs, XQuery transformations, and so on) reside in exactly one project. Projects do not overlap. Resources can be created directly under a project, or they can be further organized into folders. However, you can reference any resource regardless of the project in which it resides.

When you create a domain in Oracle Service Bus, a default project is created.

4.1 Naming Projects and Folders

Project names and folder names are limited to 64 characters and must follow the Section 2.3, "Resource Naming Restrictions."

The names, length, and levels of nesting of projects and folders are ultimately affected by the limits of your operating system. Creating folders or projects with very long names or deeply nesting folders can fail due to the limitations of the operating system.

4.1.1 Qualifying Resource Names Using Projects and Folders

Projects and folders qualify the names of Oracle Service Bus resources. A reference to a resource is constructed as follows:

project-name/root-folder/.../parent-folder/resource-name

If a resource is located directly under a project, the reference is constructed as follows:

project-name/resource-name

4.2 Projects View Page

The Projects View page lists the projects defined in the current domain. Table 4–1 describes the options and controls on the page.

Table 4–1   Projects View Page - Options and Controls

Option or Control	Description
Enter New Project Name	To create a new project, enter a name in the text field and click **Add Project**. See Section 4.4, "Adding Projects."
	See also Section 4.1, "Naming Projects and Folders."
Name	Click a project name to display the Project/Folder View Page for that project.
Options	For any project listed, you can do any of the following:
	■ Click the **Rename Project** icon to rename the project. See Section 4.5, "Renaming Projects."
	■ Click the **Move Project** icon to move the project. See Section 4.6, "Moving Projects (Converting to a Folder)."
	■ Click the **Clone Project** icon to clone the project. See Section 4.7, "Cloning Projects."
	■ Click the **Delete Project** icon to delete the project. A Deletion Warning icon appears when other resources reference this resource. You can delete this resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. See Section 4.8, "Deleting Projects."
	When you perform any of these actions, the result is saved in the current session. To end the session and deploy the configuration to the runtime, click **Activate** under **Change Center**.

4.3 Project/Folder View Page

The Project/Folder View page displays details of a single project. Table 4–2 describe the options and controls on the page.

Table 4–2   Project/Folder View Page - Options and Controls

Option or Control	Description
References	Click the number in this field (if greater than 0) to display a list of the resources outside of this project on which resources inside this project depend.
Referenced By	Click the number in this field (if greater than 0) to display a list of the resources outside of this project that are dependent on resources inside this project.
Description	Click **Edit Description** to add or edit a description of this project, then click **Submit Description** to update it.
Enter a New Folder Name	To create a new folder in this project, enter a name in the text field and click **Add Folder**. See Section 4.9, "Adding Folders."
	See also Section 4.1, "Naming Projects and Folders."
Name	Click a folder name to display the Project/Folder View Page for that folder.

Table 4–2 (Cont.) Project/Folder View Page - Options and Controls

Option or Control	Description
Options	For any folder listed, you can do any of the following: ■ Click the **Rename Folder** icon to rename the folder. See Section 4.10, "Renaming Folders." ■ Click the **Upgrade to Project or Move Folder** icon to move the folder or to upgrade it to a project. See Section 4.11, "Moving or Upgrading Folders." ■ Click the **Clone Folder** icon to clone the folder. See Section 4.12, "Cloning Folders." ■ Click the **Delete Folder** icon to delete the folder. A Deletion Warning icon appears when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. See Section 4.13, "Deleting Folders." When you perform any of these actions, the result is saved in the current session. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.
Delete	To delete folders, select the check boxes next to the names of the folders to delete, then click **Delete**.
Create Resource	To create a new resource for this project, select a resource type from the Create Resource list. See Section 4.14, "Creating Resources."
Name	Click a resource name to display details about the resource.
Resource Type	The type of this resource.
Actions	The actions available for a resource depend on the type of resource: ■ The **Edit Message Flow** icon appears for proxy services. Click this icon to edit the message of a proxy service. See Section 22.1, "Viewing and Editing Message Flows." ■ The **Launch Test Console** icon appears for business services, proxy services, XQuery transformations, XSL transformations, and MFL files. Click this icon to invoke the Test Console, which you use to validate and test the design of services and transformations. For services, you can only use the Test Console at runtime; that is, when the session is activated. For transformations, you can use the Test Console whether you are inside or outside a session. See Section 33.1, "Testing Services" and Section 33.2, "Testing Transformations." ■ The **Export WSDL** icon appears for WSDL-based proxy services. Click this icon to export a WSDL, which you can then view or modify in an external tools such as an IDE. This is different than the Export Resources functionality in the System Administration module, which you use to move and stage resources between two domains. See Section 4.28, "Exporting a WSDL." ■ The **Generate WSDL** icon appears for transport-typed business services, such as EJB and Flow. Click this icon to generate a WSDL, which you can then view or modify. See Section 4.29, "Generating a WSDL."

Table 4–2 (Cont.) Project/Folder View Page - Options and Controls

Option or Control	Description
Options	The Options column displays the following icons:
	■ Click the **Rename Resource** icon to rename the resource. See Section 4.25, "Renaming Resources."
	■ Click the **Move Resource** icon to move the resource. See Section 4.26, "Moving Resources."
	■ Click the **Clone Resource** icon to clone the resource. See Section 4.27, "Cloning Resources."
	■ Click the **Delete** icon to delete the resource. A Deletion Warning icon appears when other resources reference this resource. You can delete this resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.
	When you perform any of these actions, the result is saved in the current session. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.4 Adding Projects

To add a project:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. In the **Enter New Project Name** field, enter a unique name for the project, then click **Add Project**. See Section 4.1, "Naming Projects and Folders." The new project is saved in the current session.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.5 Renaming Projects

To rename a project:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. Click the **Rename Project** icon in the row of the project you want to rename. The Rename Project page is displayed.

4. In the **New Project Name** field enter the new name for the project. See Section 4.1, "Naming Projects and Folders."

   If you use the name of an existing project, the resources of both projects are merged under that name. For example, consider project MyProject with resources A and B, and project ThisProject with resources C and D. If you rename MyProject as ThisProject, the new ThisProject will have resources A, B, C, and D.

5. Click **Rename**. The project is renamed in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.6 Moving Projects (Converting to a Folder)

When you move a project, the project is converted to a folder in the new location.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. Click the **Move Project** icon in the row of the project you want to move. The Move Project page is displayed with the current location of the project highlighted.

4. Under Location, select a project or folder to "move" the project into. The project is converted to a folder in the new location and is saved in the current session.

   If you move a project to a new location and a folder with the same name already exists in that location, the contents of the project are merged with the contents of the existing folder.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.7 Cloning Projects

Following are the ways to clone (copy) a project:

- Clone the project with a new name at the root level. In this case, the original project and its clone exist as peer projects. The cloned project contains the same folders and resources as the original (at least until you modify the source or the clone). For instructions, see "Cloning a Project at the Root Level" in this section.

- Clone the project using the name of an existing project (other than the source project). In this case, the folders and resources from the source project are merged with the folders and resources of the project whose the name was assigned to the clone. The original project remains unchanged. For instructions, see "Cloning a Project at the Root Level" in this section.

- Clone the project in a new location, in an existing project or project folder. In this case, the cloned project is demoted to a folder in the new location. It contains the same folders and resources as the original project.

   If you clone a project with the same name and in the same location as an existing folder, the contents (subfolders and resources) of the project are merged with the contents of the target folder.

   In either case, the original project remains unchanged.

   For instructions, see "Cloning a Project and Demoting the Clone to a Folder" in this section.

Cloning a Project at the Root Level

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. Click the **Clone Project** icon in the row of the project you want to clone. The Clone Project page is displayed with the current location of the project highlighted.

4. In the **New Name** field, enter a name for the clone. See Section 4.1, "Naming Projects and Folders."

Use a unique name to create the clone as a new project, or use the name of an existing project to merge the clone with that project.

5. In the Location panel, keep the default **Root** location when cloning a project at the root level.

6. Click **Clone Project**. The project is cloned in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

Cloning a Project and Demoting the Clone to a Folder

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. Click the **Clone Project** icon in the row of the project you want to clone. The Clone Project page is displayed with the current location of the project highlighted.

4. In the **New Name** field, optionally enter a new name for the folder. See Section 4.1, "Naming Projects and Folders."

   Use a unique name to create a new folder, or use the name of an existing folder to merge the clone with the existing folder (in the location you will select in the next step).

5. In the Location panel, select a project or folder, then click **Clone Project**. The cloned project is converted to a folder in the new location and is saved in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.8 Deleting Projects

To delete a project:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** > **Projects** to display the Projects View page.

3. Click the **Delete Project** icon in the row of the project you want to delete. The project is deleted in the current session.

   When you delete a project, all resources under the project are deleted. If any resources under this project are referenced by resources under a different project, you can delete the project but with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

   Instead of the **Delete** icon, a Deletion Warning icon is displayed for these resources.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.9 Adding Folders

To add a folder:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the new folder.

3. In the **Enter New Folder Name** field of the Projects View page, enter a unique name for the folder. See Section 4.1, "Naming Projects and Folders."

4. Click **Add Folder**. The new folder is saved in the current session.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.10 Renaming Folders

To rename a folder:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder containing the folder whose name you want to change. The Project/Folder View page is displayed.

3. Click the **Rename Folder** icon in the row of the folder you want to rename. The Rename Folder page is displayed.

4. In the **New Folder Name** field, enter a new name for the folder. See Section 4.1, "Naming Projects and Folders."

5. Click **Rename**. The renamed folder is saved in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.11 Moving or Upgrading Folders

To move or upgrade a folder:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder that contains the folder that you want to move or upgrade. The Project/Folder View page is displayed.

3. Click the **Move Folder** icon in the row of the folder you want to move or upgrade. The Move Folder page is displayed with the current location of the folder highlighted.

4. Determine whether you will move the folder to a new location under a different project or upgrade the folder to a project:

   - To move it to a new location, in the Location panel, select a project or folder to move the folder into.

   - To upgrade the folder to a project, select the **Root** location.

5. Click **Submit**. The folder (or project, if it was upgraded from a folder) is saved in its new location in the current session.

   When you move or upgrade a folder, if another folder already exists in that location with the same name, the contents of the moved or upgraded folder are merged with the contents of the existing folder or project.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.12 Cloning Folders

To clone a folder:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, navigate to the parent project or folder that contains the folder that you want to clone. The Project/Folder View page is displayed.

3. Click the **Clone Folder** icon in the row of the folder you want to clone. The Clone Folder page is displayed with the current location of the folder highlighted.

4. Clone the folder in the same location or a different location, or upgrade the folder to a project:, as follows:

   ■ To clone the folder in the same location, enter a new name in the **New Name** field. See Section 4.1, "Naming Projects and Folders."

   When cloning in the same location, either do not select any location at all in the Location panel (which is equivalent to selecting the current location), or explicitly select the current project and folder.

   ■ To clone the folder in a different location, optionally enter a new name in the **New Name** field, and select a project or folder from the Location area. See Section 4.1, "Naming Projects and Folders."

   If you use the same name as an existing folder, the contents of the original folder are merged with the contents of the target folder.

   ■ To upgrade the folder to a project, select the **Root** location.

5. Click **Clone Folder**. The folder is cloned in the current session.

   When you clone a folder, all folder and resources under the folder are cloned recursively

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.13 Deleting Folders

To delete a folder:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder that contains the folder that you want to delete.

3. On the Project/Folder View page, Click the **Delete Folder** icon in the row of the folder you want to delete. The folder is deleted in the current session.

   When you delete a folder, all resources under the folder are deleted. If any resources under this folder are referenced by resources under a different folder, you can delete the folder but with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

   Instead of the **Delete** icon, a Deletion Warning icon is displayed for these resources.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.14 Creating Resources

To create a resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder in which you want to create the resource.

3. From the **Create Resource** list on the Project/Folder View page, select the type of resource you want to create. Different pages are displayed, depending on the type of resource you select.

4. Follow the prompts to add information to each page to create the resource. For more information about creating each kind of resource, see the appropriate topic, as listed in Table 4–3.

Table 4–3   Creating Resources

To Create this Type of Resource...	See this Topic...
Proxy Service	Section 20.1, "Creating and Configuring Proxy Services"
Business Service	Section 19.1, "Creating and Configuring Business Services"
Split-Join	Section 10.2, "Adding Split-Joins"
WSDL	Section 7.2, "Adding WSDLs"
XML Schema	Section 5.2, "Adding XML Schemas"
WS-Policy	Section 12.3, "Adding Custom WS-Policies"
JCA Bindings	Section 13.2, "Adding JCA Bindings"
XQuery Transformation	Section 11.2, "Adding XQuery Transformations"
XSL Transformation	Section 15.2, "Adding an XSL Transformation"
MFL File	Section 16.2, "Adding MFLs"
Service Account	Section 17.3, "Adding Service Accounts"
Service Key Provider	Section 18.2, "Adding Service Key Providers"
JAR	Section 8.2, "Adding JARs"
Alert Destination	Section 6.2, "Adding Alert Destinations"
MQ Connection	Section 9.2, "Adding MQ Connections"
XML Document	Section 14.2, "Adding XML Documents"
Resources from URL	Section 4.17, "Loading Resources from a URL"
Zipped Resources	Section 4.15, "Loading Resources from a Zip File"

5. When you create the resource, it is created in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.15 Loading Resources from a Zip File

You can select a Zip or JAR file that contains the resources you want to load; Oracle Service Bus locates and imports the resources from this Zip file.

Resources like WSDLs or XML Schemas can define full trees of dependent resources. The bulk load feature lets you upload a set of resources at once; you need not resolve the dependencies manually.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder to which you want to add resources. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Zipped Resources**. The Load Resources page is displayed.

4. In the **Select Zip File** field, specify a Zip file to import. Enter the name of the file (qualified by its path); or, click **Browse**, locate the file, then click **Open**.

5. Review the mapping of resource types to file extensions. Change or add extensions, according to your needs. Separate multiple extensions with commas.

   Each file in the Zip file is a possible resource, and the file extension is the commonly used way to identify the file type. Oracle Service Bus defines a default extension for each resource, but if you have additional extensions to describe those resources, this map enables you to declare those extensions to the loader.

   After you define the Zip file and extension map, the Oracle Service Bus loader scans the contents of the Zip file and tries to associate a resource type with each file. A file without an extension or one that does not have an extension defined in the map is considered *unknown* and is automatically excluded from the load.

   For known files, the name of the resource is the name of the file without its extension. The folder structure of the Zip file is recreated in the target Oracle Service Bus Administration Console folder.

   An extension can only be associated with one file type.

6. Click **Next**.

7. On the Load Resources - Review Loaded Resources page, review the resources you are importing. You can clear the check boxes next to any resources you do not want to include in this import.

8. To import the selected resources, click **Import**.

9. The Load Resources - Import Result page displays the imported resources. Click **Load Another** to import resources from another file, or proceed with other tasks by selecting a different page from the left navigation pane.

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.16 Load Resources - Select Zip File and Review Type Extensions Page

Use the Load Resources - Select Zip File and Review Type Extensions page to import resources from a Zip file, as documented in Section 4.15, "Loading Resources from a

Zip File."

Table 4–4 describes the options and controls on the page.

Table 4–4   Load Resources: Select Zip File and Review Type Extensions Page - Options and Controls

Options and Controls	Description
Select Zip File	Enter the fully qualified name of a Zip file to import, or click **Browse** to select it.
Resource Type and File Extension	Each file in the Zip file is a possible resource, and the file extension is the commonly used way to identify the file type. Oracle Service Bus defines a default extension for each resource, but if you have additional extensions to describe those resources, this map enables you to declare those extensions to the loader.
	After you define the Zip file and extension map, the Oracle Service Bus loader scans the contents of the Zip file and tries to associate a resource type with each file. A file without an extension or one that does not have an extension defined in the map is considered unknown and is automatically excluded from the load.
	For known files, the name of the resource is the name of the file without its extension. The folder structure of the Zip file is recreated in the target Oracle Service Bus Administration Console folder.

4.17 Loading Resources from a URL

You can perform a bulk import of resources available at a URL or on the file system. Bulk import lets you import a root resource, such as a WSDL, along with its dependents, such as other WSDLs and schemas. The dependency map is resolved automatically.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder to which you want to add resources. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Resources from URL**. The Load Resources - Load Resources from URL page is displayed.

4. In the **URL/Path** field, enter the URL or path of the resource.

   To specify a local resource, you can use the `file` protocol with a file name fully qualified by its path, for example:

   file:///c:/osbresources/ForeachAction.jar

5. In the **Resource Name** field, enter the resource's name.

6. From the **Resource Type** list, select the type of the resource.

7. Click **Next** to display the Load Resources - Review Loaded Resources Page. Clear the check boxes next to any resources you do not want to include in this import.

8. To import the selected resources, click **Import**.

9. The Load Resources - Import Result page displays the imported resources. Click **Load Another** to import resources from another file, or proceed with other tasks by selecting a different page from the left navigation pane.

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.18 Load Resources - Load Resources from URL Page

Use the Load Resources - Load Resources from URL page to import resources from a file available on the Web, as documented in Section 4.17, "Loading Resources from a URL."

Table 4–5 describes the options and controls on the page.

Table 4–5   Load Resources: Load Resources from URL Page - Options and Controls

Options and Controls	Description
URL/Path	Enter the URL where the file is located.
	To specify a local resource, you can use the `file` protocol with a file name fully qualified by its path, for example:
	file:///c:/osbresources/ForeachAction.jar
Resource Name	Enter a name for the resource.
Resource Type	Select the type of resource from the list.

4.19 Load Resources - Review Loaded Resources Page

The Load Resources - Review Loaded Resources page displays details about resources being imported from a Zip or other file, as documented in Section 4.15, "Loading Resources from a Zip File" or Section 4.17, "Loading Resources from a URL."

Table 4–6 describes the options and controls on the page.

Table 4–6   Load Resources: Review Loaded Resources Page - Options and Controls

Options and Controls	Description
File Name	The file name of the resource, including the file extension.
Name	The name of the resource.
Path	The path of the resource, which is the project and folder where the resource resides.
Resource Type	Lists the supported resource types.

A Warning icon next to file name indicates that the resource type is unknown. (A file without an extension or one that does not have an extension defined in the map is considered *unknown*.) Files of unknown file types cannot be imported.

4.20 Load Resources - Import Result Page

The Load Resources - Import Result page displays the results of loading resources from an imported Zip file or from a URL.

Depending on whether you have loaded resources from a Zip file or a URL, make sure you have completed the steps in either Section 4.15, "Loading Resources from a Zip File" or Section 4.17, "Loading Resources from a URL," and make sure you have completed the steps in Section 4.19, "Load Resources - Review Loaded Resources Page."

The Import Result page is displayed, which displays the following information.

Table 4–7   Project/Folder View Page - Options and Controls

Option or Control	Description
Status	The status of the resource; for example, whether or not it has been imported successfully. A WSDL may be temporarily marked as invalid until all of its dependencies are imported. When the import action is completed, however, the configuration service re-validates the files marked invalid.
Name	The name of the resource.
Path	The path of the resource, which is the project and folder where the resource resides.
Resource Type	Displays the imported resource type.
Error Message	Displays the details of the error message if the resource has not been imported successfully.

4.21 View References Page

Depending on whether you clicked **References** or **Referenced By** on a Project/Folder View page, the View References page displays a list of:

- Resources outside of the current project or folder that are referenced by resources inside the project or folder.

- Resources outside of the current project or folder that reference resources inside the project or folder.

Table 4–8 describes the options and controls on the page.

Table 4–8   Project References Page - Options and Controls

Option or Control	Description
Name	Click the name of the resource to display details of the resource.
Resource Type	Displays Project, Folder, or any of the supported Oracle Service Bus resource types.

4.22 Viewing References to Resources

You can view references to resources whether or not you are in a session. See Section 3.1, "Using the Change Center."

1. Click **References** or **Referenced By** on a project, folder or resource details page in the Project Explorer or Resource Browser modules.

   - Click **References** to display resources outside of a project or folder that are referenced by resources inside the project or folder.

   - Click **Referenced By** to display resources outside of a project or folder that reference resources inside the project or folder.

   The View References page is displayed.

2. Click a name of a resource to display details about it, or click **Back** to return to the previous page.

4.23 View Change History Page

The View Change History page displays the ten most recent changes to a resource, across all session activations.

Table 4–9 describes the options and controls on the page.

Table 4–9   Change History Page - Options and Controls

Option or Control	Description
Task	The type of task performed: ■ Create ■ Update ■ Clone ■ Rename ■ Move
User	The user who created or edited this resource.
Execution Time	The date and time that the user created or edited this resource.

4.24 Using the Resource Browser

The Resource Browser provides tools for locating resources and for performing simple actions on them, such as Display, Edit, Export, and Delete.

1. To display the Resource Browser, select **Resource Browser** from the left navigation pane.

2. On the left side of the page, click the name of the type of resource you want to locate. Table 4–10 shows the items you can select and the tasks associated with the page displayed for each one.

Table 4–10   Resource Types Available from the Resource Browser

Resource Type	Associated Task
Proxy Services	Section 20.4, "Locating Proxy Services"
Business Services	Section 19.4, "Locating Business Services"
Split-Joins	Section 10.1, "Locating Split-Joins"
WSDLs	Section 7.1, "Locating WSDLs"
XML Schemas	Section 5.1, "Locating XML Schemas"
WS-Policies	Section 12.2, "Locating Custom WS-Policies"
JCA Bindings	Section 13.1, "Locating JCA Bindings"
XQueries	Section 11.1, "Locating XQuery Transformations"
XSLTs	Section 15.1, "Locating XSL Transformations"
MFLs	Section 16.1, "Locating MFLs"
Service Accounts	Section 17.2, "Locating Service Accounts"
Service Key Providers	Section 18.1, "Locating Service Key Providers"
JARs	Section 8.1, "Locating JARs"
Alert Destinations	Section 6.1, "Locating Alert Destinations"
MQ Connection	Section 9.1, "Locating MQ Connections"

Table 4–10   (Cont.) Resource Types Available from the Resource Browser

Resource Type	Associated Task
XML Documents	Section 14.1, "Locating XML Documents"

4.25 Renaming Resources

To rename a resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder containing the resource you want to rename. The Project/Folder View page is displayed.

3. Click the **Rename** icon in the row of the resource you want to rename. The Rename page is displayed.

4. In the **New Resource Name** field, enter a new name for the resource. See Section 4.1, "Naming Projects and Folders."

5. Click **Rename**.

> **Note:**   If you rename a business service imported from the UDDI registry, the service will become detached from the registry

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.26 Moving Resources

To move a resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder containing the resource you want to move. The Project/Folder View page is displayed.

3. Click the **Move** icon in the row of the resource you want to move. The Move page is displayed with the current location of the resource highlighted.

4. In the Location field, select the project and folder to which you want to move the file.

5. Click **Move File**. The resource is moved to the new location in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.27 Cloning Resources

To clone a resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder containing the resource you want to clone. The Project/Folder View page is displayed.

3. Click the **Clone Resource** icon in the row of the resource you want to clone. The Clone Resource page is displayed with the current location of the resource highlighted.

4. You can clone the resource in the same location or a different location:

    ■ To clone the resource in the same location, enter a new name in the **New Resource Name** field. See Section 4.1, "Naming Projects and Folders."

       To clone in the same location, either do not select any location at all in the Location panel (which is equivalent to selecting the current location), or explicitly select the current project and folder.

    ■ To clone the resource in a different location, optionally enter a new name in the **New Resource Name** field, and select a project, folder, or both from the Location area.

5. Click **Clone Resource**. The resource is cloned in the current session.

    If you clone a business service imported from the UDDI registry, the cloned service will be detached from the registry

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

4.28 Exporting a WSDL

Use the Project/Folder View page to export a WSDL associated with a proxy service or a business. The WSDL is exported as a JAR file. You can export a WSDL only when you are outside a session. See Section 3.1, "Using the Change Center."

Use the Export WSDL feature to make a WSDL available for consumption by external tools such as IDEs. This is different than the Export Resources feature in the System Administration module, which you use to move and stage resources between two domains.

1. From the left navigation pane, click a project or folder from under Project Explorer. The Project/Folder View page is displayed.

2. Click the **Export WSDL** icon in the row of the proxy service whose WSDL you want to export. The File Download dialog box is displayed.

3. In the File Download dialog box, click **Open** to open the JAR file or click **Save** to save the JAR file to your desktop.

4.29 Generating a WSDL

A generated WSDL is a WSDL resource that Oracle Service Bus generates for a service that did not start with a WSDL resource but which can be described using a WSDL. For example, a WSDL created from an EJB-based service is a generated WSDL. For more information, see "How WSDL is Used in Oracle Service Bus" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Use the Project/Folder View page to generate a WSDL associated with an EJB or Flow transport-typed business service.

1. From the left navigation pane, click a project or folder from under Project Explorer. The Project/Folder View page is displayed.

2. Click the **Generate WSDL** icon in the row of the business service whose WSDL you want to generate. The Generate WSDL Resource dialog box is displayed with

the current location (project name and the name of the folder in which the
business service resides) highlighted.

3. In the Generate WSDL Resource dialog box, specify a **New WSDL Resource
Name** and location for the WSDL, and click **Generate WSDL**.

Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

After you finish

Continue in Section 7.3, "Editing WSDLs" to view or modify the generated WSDL.

4.30 Viewing Resources in a Web Browser

You can view/access Oracle Service Bus resources in a standard Web browser using
the following URLs:

- Path for **WSDL** resources:

  `http://host:port/sbresource?WSDL/project_path/wsdlname`

  or

  `http://host:port/proxy_service_endpoint_URI?WSDL` (for WSDL-based HTTP
  proxy services)

- Path for WSDL-based **Proxy Service** resources to display the WSDL:

  `http://host:port/sbresource?PROXY/project_path/proxyname`

  `http://host:port/sbresource?ORAPROXY/project_path/proxyname` (for proxy
  services with Oracle Web Services Manager policies attached)

- Path for WSDL-based **Business Service** resources to display the WSDL:

  `http://host:port/sbresource?BIZ/project_path/bizname`

  `http://host:port/sbresource?ORABIZ/project_path/bizname` (for business
  services with Oracle Web Services Manager policies attached)

- Path for **WSS** resources:

  `http://host:port/sbresource?POLICY/project_path/policyname`

- Path for **MFL** resources:

  `http://host:port/sbresource?MFL/project_path/mflname`

- Path for **SCHEMA** resources:

  `http://host:port/sbresource?SCHEMA/project_path/schemaname`

> **Note:** If you use special characters in your resource names, the URLs
> used to expose the resources in Oracle Service Bus must be encoded in
> UTF-8 in order to escape special characters.

You can also retrieve WSDLs containing Oracle Web Services Manager policies so that
the policies conform to supported WS-Policy and WS-Security Policy standards. For
more information, see "Advertising WSDLs to Support WS Standards" in the *Oracle
Fusion Middleware Developer's Guide for Oracle Service Bus*.

4.31 Deleting Resources

To delete a resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder containing the resources you want to delete.

3. Click the **Delete** icon in the row of the resource you want to delete. The resource is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

5

XML Schemas

This chapter describes how to create, locate, edit, delete, and fix unresolved references to XML Schema resources using the Oracle Service Bus Administration Console.

Schemas describe types for primitive or structured data. XML schemas are an XML vocabulary that describe the rules that XML business data must follow. XML schemas specify the structure of documents, and the data type of each element and attribute contained in the document.

You use XML schemas as references for WSDL resources. You can also use XML schemas in the Message Flow of proxy services to validate an element specified with an XPath expression. See Section 21.24, "Adding Validate Actions."

5.1 Locating XML Schemas

To locate schemas:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the XML schema.

   - Select **Resource Browser** > **XML Schemas**. The Summary of XML Schemas page displays the information shown in Table 5–1. For a more detailed description of the properties, see Section 5.3, "Editing XML Schemas."

2. To locate a specific XML schema:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - To restrict the number of items in the list, you can filter by name, path, and namespace. In the **Name**, **Path**, and **Namespace** fields, enter the name, path and namespace of the search target(s), then click **Search**.

     The path is the project name and the name of the folder in which the XML schema resides.

   - Click **View All** to remove the search filters and display all XML schemas.

Table 5–1   XML Schema Information

Property	Description
XML Schema Name	The unique name assigned to the XML schema. The name is a link to the XML Schema Details page. See Section 5.3, "Editing XML Schemas."

Table 5–1   (Cont.) XML Schema Information

Property	Description
Path	The path is the project name and the name of the folder in which the XML schema resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
XML Schema Namespace	The namespace used to qualify any of the definitions included in the XML schema.
Options	Click the **Delete** icon to delete a specific XML schema. See Section 5.4, "Deleting XML Schemas."

5.2 Adding XML Schemas

To add a schema:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the new XML schema. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **XML Schema** to display the Create a New XML Schema Resource page.

4. In the **Resource Name** field, enter a unique name for this XML schema. This is a required field.

    Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the XML schema.

6. In the **XML Schema** field, do one of the following:

    ■   Enter text for the new XML schema.

    ■   Click **Browse** to locate and import an existing XML schema.

    ■   Copy and paste text from an existing XML schema into this field.

    This is a required field.

7. Click **Save**. The new XML schema is saved in the current session.

    When you click **Save**, if there are any unresolved references for the new XML schema, the system displays them. See Section 5.5, "Viewing Unresolved XML Schema References."

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

5.3 Editing XML Schemas

To edit a schema:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the XML schema, as described in Section 5.1, "Locating XML Schemas."

3. Click the XML schema name. The View XML Schema Details page displays the information shown in Table 5–2, Table 5–3, and Table 5–4.

Table 5–2   XML Schema Details

Property	Description
Last Modified By	The user who created this XML schema or imported it into the configuration.
Last Modified On	The date and time that the user created this XML schema or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this XML schema references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this XML schema. If such references exist, click the numeric link to view a list of the objects. For example, if this XML schema is referenced by a specific WSDL, the WSDL is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this XML schema, if one exists.

The structural view of the XML schema is displayed by default, which includes the information shown in Table 5–3.

Table 5–3   XML Structure Details

Property	Description
Target Namespace	The namespace used to qualify any of the definitions included in the XML schema.
XML Schema Definitions	The attributes and groups associated with the XML schema: ■ XML Schema Types ■ XML Schema Elements ■ XML Schema Imports ■ XML Schema Includes/Redefines
Schema State	Status of the XML schema: ■ Valid An XML schema is shown as valid when all of the locations for XML schemas included by the current XML schema are specified. Furthermore, all of the locations for nested XML schemas (XML schemas subsequently included by the included XML schemas) are specified. ■ Invalid An XML schema is shown as invalid when one or more of the locations for the XML schema is not specified. Additionally, the location for one or more nested XML schemas (XML schemas subsequently included by the included XML schemas) may not be specified. To resolve an invalid XML schema, click **Edit References**. See Section 5.6, "Resolving Unresolved XML Schema References."

4. Click **Text View** to display a text view of the XML schema details. The text view includes the information shown in Table 5–4.

Table 5–4   XML Schema Text Details

Property	Description
Target Namespace	The namespace used to qualify any of the definitions included in the XML schema.
XML Schema	The text for this XML schema.

5. To make changes to the fields, click **Edit**. See Section 5.2, "Adding XML Schemas" for descriptions of the fields.

6. Make the appropriate edits.

7. Click **Save** to commit the updates in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

5.4 Deleting XML Schemas

To delete a schema:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **XML Schemas**.

3. Click the **Delete** icon in the Options field of the XML schema you want to delete. The XML schema is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

5.5 Viewing Unresolved XML Schema References

For each unresolved XML schema reference, this page displays the information shown in Table 5–5.

Table 5–5   Unresolved XML Schema Details

Property	Description
Schema Name	The unique name assigned to the XML schema. Click the name of a specific XML schema to view details for that XML schema. See Section 5.3, "Editing XML Schemas."
Namespace	The namespace used to qualify any of the definitions included in the XML schema.

To locate a specific reference:

- Resort the list. Click an underlined column name. Ascending and descending arrows indicate the sort order. Click the column name to change the sort order.

- Scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

5.6 Resolving Unresolved XML Schema References

Use the Edit the References of an XML Schema Resource page to resolve unresolved XML schema references by configuring the mapping for XML schema references such as XML schema imports.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. On the Summary of XML Schemas page, in the **XML Schema Name** column, click the appropriate XML schema name to view details of the unresolved reference. The View XML Schema Details page is displayed.

3. Click **Edit References** to display the Edit the References of an XML Schema Resource page.

4. In the **Resource Type** field, select **XML Schema**.

5. Click **Browse**. The XML Schema Browser is displayed.

6. In the XML Schema Browser, select an XML schema, then select a definition from the Definitions pane.

7. Click **Submit**. The XML schema you selected is displayed in the **Resource Name** field.

8. Click **Save** to resolve the reference.

9. On the XML Schema Details page, click **OK**. The state of the XML schema is displayed as Valid.

6

Alert Destinations

This chapter describes how to create, locate, edit, and delete Alert Destination resources using the Oracle Service Bus Administration Console.

An alert destination resource captures a list of recipients that can receive alert notifications from the Oracle Service Bus. In typical system monitoring contexts, alerts generated by Oracle Service Bus bear significance to a finite set of users.

In Oracle Service Bus, you can configure each alert destination resource to include a set of recipients according to a given context. Alert destinations are used by Alert actions configured in the message flow, and also by SLA alert rules. An alert destination can include one or more of the following types of destinations: Administration Console (default), Reporting Data stream, SNMP trap, email, JMS queue, or JMS topic.

In the case of email and JMS destinations, a destination resource can include a list of email addresses or JMS URIs, respectively.

You can re-use alert destinations across alert configurations for services. See Section 21.25, "Adding Alert Actions."

6.1 Locating Alert Destinations

You can view a list of alert destinations in the Summary of Alert Destinations page. If you have not yet created an alert destination, and want to do so, create a new alert Destination resource from the Project Explorer. To learn how, see Section 6.2, "Adding Alert Destinations."

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the alert destination.

   - Select **Resource Browser** > **Alert Destinations**. The Summary of Alert Destinations page displays the information shown in Table 6–1.

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, enter the name and path of the search target(s), then click **Search**.

   The path is the project name and the name of the folder in which the alert destination resides.

   Click **View All** to remove the search filters and display all alert destinations.

Table 6–1   Alert Destination Details

Property	Description
Name	The name of the alert destination resource. The name is a link to the View an Alert Destination - Configuration page. See Section 6.4, "Editing Alert Destinations."
Path	The path is the project name and the name of the folder in which the alert destination resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Click the **Delete** icon to delete a specific Alert Destination. See Section 6.5, "Deleting Alert Destinations."

6.2 Adding Alert Destinations

An alert destination is a destination address for alert notifications in Oracle Service Bus. You can create alert destination resources in the Create Alert Destination page.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the alert destination. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Alert Destination** to display the Create Alert Destination page.

4. In the **Resource Name** field, enter a unique name for this alert destination. This is a required field.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the alert destination.

6. Select any or all of the following destinations to be included in this alert destination resource.

   > **Note:**   Alerts are aggregated in the Oracle Service Bus Administration Console at runtime, where you can view them under **Operations** > **Dashboard**.

   - **SNMP Trap**: If you specify SNMP Trap, alerts are sent as SNMP traps, and can be processed by any third-party enterprise monitoring systems (ESM).

   - **Reporting**: If you specify Reporting, alerts are sent to the Oracle Service Bus Reporting module and can be captured using a custom Reporting Provider that can developed using the Reporting APIs provider by Oracle Service Bus. This allows third-parties to receive and process alerts in custom Java code.

   - **Alert Logging**: If you select **Yes**, alerts sent to this alert destination are logged to the alert log.

7. Add an email recipient to the alert destination resource definition. See Section 6.3.1, "Adding Email Recipients."

8. Add a JMS Destination to the alert destination resource definition. See Section 6.3.2, "Adding JMS Destinations."

9. Click **Save**. The new alert destination is saved in the current session.

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

6.3 Adding Email and JMS Recipients

You can add multiple email and JMS recipients to an alert destination.

This section includes the following tasks:

- Section 6.3.1, "Adding Email Recipients"
- Section 6.3.2, "Adding JMS Destinations"

6.3.1 Adding Email Recipients

Before you add an email destination, you must configure an SMTP Server (see Section 31.7, "Adding SMTP Servers"), or a JavaMail session in WebLogic Server (see "Configure access to JavaMail" in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*. If there are no SMTP Server resources or JavaMail sessions available, configured, you cannot configure an email recipient.

1. Select **Send an alert via email** from the list and click **Add** to display the Add Email Recipient page.

2. In the **Mail Recipients** field, enter an email recipient in the format: `mailto:username@hostname`

   This is a required field.

   You can specify multiple email recipients by entering the user names and hostnames in a comma-separated list. For example, `mailto:username@hostname [,username_1@hostname_1]...[,username_n@hostname_n]`

   Only the first mail recipient must be prefixed with the text "`mailto:`".

3. In the **SMTP Server** field, select the name of the SMTP Server for the outgoing email. This field is not required if a mail session is selected in the next step.

4. In the **Mail Session** field, select an available mail session. This field is not required if an SMTP Server is selected in the previous step.

5. In the **From Name** field, provide a sender's name for the alert notification. This field is optional.

6. In the **From Address** field, provide a valid email address. This field is required if a value for "From Name" field is specified.

7. In the **Reply To Name** field, provide a name to which a reply may be addressed. This field is optional.

8. In the **Reply To Address** field, provide an email address to which a reply may be sent. This field is required if a value for "Reply To Name" field is specified.

9. In the **Connection Timeout** field, enter the number of seconds a connection must wait for a response from the server before timing out. The default value is 0.

10. In the **Request Encoding** field, enter a character set encoding value. The default encoding value is `iso-8859-1`.

11. Click **Save**. The email recipient is added to the list of email recipients for this alert destination and saved in the current session.

6.3.2 Adding JMS Destinations

To add JMS destinations:

1. Select **Send an alert to a JMS Destination** from the list and click **Add** to display the Add JMS Destination page.

2. In the destination URI field, enter a JMS destination URI in the format:
   `jms://host:port/factoryJndiName/destJndiName`

   This is a required field.

   > **Note:** While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

3. In the **Destination Type** field, select **Queue** or **Topic**.

4. In the **Message Type** field, select **Bytes** or **Text**.

5. In the **Request Encoding** field, enter a character set encoding value. The default encoding value is `UTF-8`.

6. Click **Save**. The JMS destination is added to the list of JMS destinations to be included for this alert destination and saved in the current session.

6.4 Editing Alert Destinations

To edit alert destinations:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the alert destination, as described in Section 6.1, "Locating Alert Destinations" or Section 27.1.5, "Finding Alert Destinations."

3. Click the alert destination name. The View Alert Destination - Configuration page displays the information shown in Table 6–2.

Table 6–2   Alert Destination Configuration Details

Property	Description
Last Modified By	The user who created or edited this alert destination.
Last Modified On	The date and time that the user created or edited this alert destination. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this alert destination references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this alert destination. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this alert destination, if one exists.

Table 6–2   (Cont.) Alert Destination Configuration Details

Property	Description
SNMP Trap	Yes or No, to indicate whether the SNMP Trap option was specified as a destination for this alert when it was configured.
Reporting	Yes or No, to indicate whether the Reporting option was specified as a destination for this alert when it was configured.
Alert Logging	Yes or No, to indicate whether Alert Logging is enabled.
Email & JMS	Displays a list of email recipients or JMS destinations, or both, if they were configured.

4. You can change your selection of the SNMP Trap, Reporting, or Alert Logging setting.

5. To edit or delete an email recipient or JMS destination, click **Edit** at the bottom of the page.

    a. To edit the email recipient, in **Email** click the **Edit** icon under Options, for the email recipient you want to edit. Use the Edit Email Recipient page to make changes to the email recipient information. See Section 6.3.1, "Adding Email Recipients."

    b. To edit the JMS destination, in **JMS** click the **Edit** icon under Options, for the JMS destination you want to edit. Use the Edit JMS Destination page to make changes to the JMS destination information. See Section 6.3.2, "Adding JMS Destinations."

    c. To delete one or all email recipients or JMS destinations, click the **Delete** icon under Options, for the email recipients or JMS destinations you want to delete.

6. Click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

6.5 Deleting Alert Destinations

To delete an alert destination:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Alert Destinations** to display the Summary of Alert Destinations page.

3. Click the **Delete** icon in the Options field of the alert destination you want to delete. The alert destination is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

To delete particular email recipients or JMS destinations configured within an alert destination, see Section 6.4, "Editing Alert Destinations."

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

7

WSDLs

This chapter describes how to create, locate, edit, delete, and fix unresolved references to WSDL resources using the Oracle Service Bus Administration Console.

A WSDL (Web Service Definition Language) is the formal description of a Web Service; in Oracle Service Bus, it describes a proxy service or a business service. A WSDL is used to describe what a Web Service can do, where it resides, and how to invoke it.

You can base SOAP and XML services on an existing WSDL resource. A WSDL document is available for proxy and business services for any transport. See Section 20.2.1, "General Configuration Page" and Section 19.1, "Creating and Configuring Business Services."

For more information, see "How WSDL is Used in Oracle Service Bus" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

7.1 Locating WSDLs

To locate WSDLs:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View Page or the Project/Folder View page. Then navigate through projects and folders to find the WSDL.

   - Select **Resource Browser** > **WSDLs**. The Summary of WSDLs page displays the information shown in Table 7–1. For a more detailed description of the properties, see Section 7.3, "Editing WSDLs."

2. To locate a specific WSDL:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match *all* the criteria will be returned.

   - To restrict the number of items in the list, you can filter by name, path, and namespace. In the **Name**, **Path**, and **Namespace** fields, enter the name, path, and namespace of the search target(s), then click **Search**.

     The path is the project name and the name of the folder in which the WSDL resides.

   - Click **View All** to remove the search filters and display all WSDLs.

Table 7–1   WSDL Details

Property	Description
WSDL Name	The unique name assigned to the WSDL. The name is a link to the View WSDL Details page. See Section 7.3, "Editing WSDLs."

Table 7–1   (Cont.) WSDL Details

Property	Description
Path	The path is the project name and the name of the folder in which the WSDL resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
WSDL Namespace	The address of the WSDL. It is displayed in the format: `http://example.com/example/example`
Options	Click the **Delete** icon to delete a specific WSDL. See Section 7.4, "Deleting WSDLs."

7.2 Adding WSDLs

Before You Begin: If the WSDL resource you want to create contains URL references to external schemas that do not currently exist in Oracle Service Bus, such as http://www.w3.org/2001/XMLSchema.xsd, you must import those URL-referenced schemas—and any dependent schemas—into Oracle Service Bus by creating XML Schema resources. WSDL resources in Oracle Service Bus can only reference locally available schemas. For more information, see Section 5.2, "Adding XML Schemas."

To add a WSDL:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the WSDL. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **WSDL** from under Interface to display the Create a New WSDL Resource page.

4. In the **Resource Name** field, enter a unique name for this WSDL. This is a required field.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the WSDL.

6. In the **WSDL** field, do one of the following:

   - Enter text for the new WSDL.

   - Click **Browse** to locate and import an existing WSDL.

   - Copy and paste text from an existing WSDL into this field.

   This is a required field.

7. Click **Save**. The new WSDL is saved in the current session.

   When you click **Save**, if there any unresolved references for the new WSDL, the system displays them. See Section 7.5, "Viewing Unresolved WSDL References."

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

7.3 Editing WSDLs

To edit a WSDL:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the WSDL, as described in Section 7.1, "Locating WSDLs."

3. Click the WSDL name. The View WSDL Details page displays the information shown in Table 7–2, Table 7–3, and Table 7–4.

Table 7–2   WSDL Details

Property	Description
Last Modified By	The user who created this WSDL or imported it into the configuration.
Last Modified On	The date and time that the user created this WSDL or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this WSDL references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this WSDL. If such references exist, click the numeric link to view a list of the objects. For example, if you select this WSDLs port or binding as the service type for a specific business service or proxy service, the business service or proxy service is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this WSDL, if one exists.

The structural view of the WSDL is displayed by default, which includes the information shown in Table 7–3.

Table 7–3   WSDL Structure Details

Property	Description
Target Namespace	The namespace used to qualify any of the definitions included in the WSDL.
WSDL Definitions	The attributes and groups associated with the WSDL:
	■ WSDL Port Types
	■ WSDL Bindings
	■ WSDL Ports
	■ XML Schema Types
	■ XML Schema Elements
	■ WSDL Imports
	■ XML Schema Imports
	■ WS-Policy References
	■ Implicit WS-Policy References

Table 7–3   (Cont.) WSDL Structure Details

Property	Description
WSDL State	The status of the WSDL: ■ Valid All of the locations for XML schemas or WSDLs included by the current WSDL are specified and are valid. Furthermore, all of the locations for nested XML schemas or WSDLs (XML schemas or WSDLs subsequently included by the included WSDLs or XML schemas) are specified and are valid. ■ Invalid One or more of the locations for XML schemas or WSDLs included by the current WSDL is not specified or is not valid. Additionally, the location for one or more of the nested XML schemas or WSDLs (XML schemas or WSDLs subsequently included by the included WSDLs or XML schemas) may not be specified or may not be valid.To resolve an invalid XML schema or WSDL, click Edit References. See Section 7.6, "Resolving Unresolved WSDL References."

4. Click **Text view** to display a text view of the WSDL details. The text view includes the information shown in Table 7–4.

Table 7–4   WSDL Text Details

Property	Description
Target Namespace	The namespace used to qualify any of the definitions included in the WSDL.
Text	The text for this WSDL.

5. To make a change to the fields, click **Edit**. See Section 7.2, "Adding WSDLs" for descriptions of the fields.

6. Make the appropriate edits.

7. Click **Save** to commit the updates in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

> **Note:** The View WSDL Details page may also include a **Separate Callbacks** button. This button is only displayed if the WSDL has port types with callback operations that can be rewritten. If you click this button, the WSDL is rewritten.

7.4 Deleting WSDLs

To delete a WSDL:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **WSDLs**.

3. Click the **Delete** icon in the Options field of the WSDL you want to delete. The WSDL is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a

warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

7.5 Viewing Unresolved WSDL References

The View Unresolved WSDL References page displays the information shown in Table 7–5.

Table 7–5   Unresolved WSDL Reference Details

Property	Description
WSDL Name	The name assigned to the WSDL. Click the name of a specific WSDL to view details for that WSDL.
Namespace	The address of the WSDL. It is displayed in the format: `http://example.com/example/example`

To locate a specific reference:

- Resort the list. Click an underlined column name. Ascending and descending arrows indicate the sort order. Click the column name to change the sort order.

- Scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

7.6 Resolving Unresolved WSDL References

Use the Edit the References of a WSDL Resource page to resolve unresolved WSDL references by configuring the mapping for WSDL references such as WSDL imports and XML schema imports. A WSDL may also have an unresolved reference to a WS-Policy.

Resolving an Unresolved WSDL or XML Schema Reference

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **WSDLs**.

3. On the Summary of WSDLs page, in the WSDL Name column, click the appropriate WSDL name to view details of the unresolved reference. The View WSDL Details page is displayed.

4. Click **Edit References** to display the Edit the References of a WSDL Resource page.

5. In the **Resource Type** field, select **WSDL** or select **XML Schema**.

6. Click **Browse**. Depending on the resource type, the WSDL Browser or the XML Schema Browser is displayed.

7. In the WSDL Browser or XML Schema Browser, select a WSDL or XML Schema, then select a definition from the Definitions pane.

8. Click **Submit**. The WSDL or XML Schema you selected is displayed in the **Resource Name** field.

9. Click **Save** to resolve the reference.

10. On the View WSDL Details page, click **OK**. The state of the WSDL is displayed as Valid.

Resolving an Unresolved WS-Policy Reference

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **WSDLs**.

3. On the Summary of WSDLs page, in the WSDL Name column, click the appropriate WSDL name to view details of the unresolved reference. The View WSDL Details page is displayed.

4. Click **Edit References** to display the Edit the References of a WSDL Resource page.

5. In the **WS-Policy Name** field, click **Browse**. The Policy Browser is displayed.

6. In the Policy Browser, select a WS-Policy.

7. Click **Submit**. The WS-Policy you selected is displayed in the **WS-Policy Name** field.

8. Click **Save** to resolve the reference.

9. On the View WSDL Details page, click **OK**. The state of the WSDL is displayed as Valid.

8

JARs

This chapter describes how to create, locate, edit, and delete JAR resources using the Oracle Service Bus Administration Console.

A JAR (Java ARchive) is a zipped file that contains a set of Java classes. It is used to store compiled Java classes and associated metadata that can constitute a program. A JAR acts like a callable program library for Java code elements (so that a single compilation link provides access to multiple elements, rather than requiring bindings for each element individually). To learn more about JARs, see http://docs.oracle.com/javase/tutorial/deployment/jar/.

JAR resources in Oracle Service Bus are used in:

- Java callout actions
- EJB-based business services
- Tuxedo-based proxy and business services

8.1 Locating JARs

To locate JARs:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the JAR.

   - Select **Resource Browser** > **JARs**. The Summary of JARs page displays the information shown in Table 8–1. For a more detailed description of the properties, see Section 8.3, "Editing JARs."

2. To restrict the number of items in the list or locate a specific JAR, search for a JAR using name and path filters. In the **Name** and **Path** fields, enter the name and path of the search target(s), then click **Search**.

   The path is the project name and the name of the folder in which the JAR resides.

   This search supports wildcard characters. For example, you can enter "j*" in the **Name** field to get a list of JARs that start with "j".

   Click **View All** to remove the search filters and display all JARs.

Table 8–1   JAR Details

Property	Description
JAR Name	The unique name assigned to the JAR. The name is a link to the View a JAR Resource - JAR Configuration page. See Section 8.3, "Editing JARs."

Table 8–1   (Cont.) JAR Details

Property	Description
Path	The path is the project name and the name of the folder in which the JAR resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
No. of Classes	The number of classes the JAR contains.
Size	The size of the JAR in bytes.
Options	Click the **Delete** icon to delete a specific JAR. See Section 8.5, "Deleting JARs."

8.2 Adding JARs

To add a JAR:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the JAR. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **JAR** from under Utility to display the Create a New JAR Resource page.

4. In the **Resource Name** field, enter a unique name for this JAR. This is a required field.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the JAR.

6. In the **JAR Source File** field, do one of the following:

   ■ Enter the path to the location of the JAR.

   ■ Click **Browse** to locate and import an existing JAR.

   This is a required field.

7. Click **Save**. The JAR resource is created and saved in the current session.

   If the JAR loads without problems, the contents of the JAR will be available for viewing. To learn how to view JAR details, see Section 8.3, "Editing JARs."

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

8.3 Editing JARs

Use the View a JAR Resource - Configuration page to update JAR files. Updating a JAR file essentially involves pointing the resource to a new version of the JAR, as you cannot edit a JAR using Oracle Service Bus.

1. If you have not already done so, click **Create** to create a new session, or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the JAR, as described in Section 8.1, "Locating JARs."

3. Click the JAR name. The View a JAR Resource - JAR Configuration page displays the information shown in Table 8–2.

Table 8–2   JAR Information

Property	Description
Last Modified By	The user who imported this JAR into the configuration.
Last Modified On	The date and time that the user imported this JAR into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of JARs that this JAR references. If such references exist, click the numeric link to view a list of the JARs. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this JAR. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this JAR, if one exists.

The JAR configuration is displayed by default, which includes the information shown in Table 8–3.

Table 8–3   JAR Configuration Details

Property	Description
JAR Configuration	The list of classes and interfaces in this JAR.
	Click (+) beside the name of a Java class to display a list of methods included in the class.

4. To make a change to the fields, click **Update**.

   The only field you can edit is **Description**. You cannot edit a JAR in Oracle Service Bus. To update the JAR, you must select a new JAR to use. To learn how, see Section 8.2, "Adding JARs."

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

Note: If you change the JAR used by an EJB Business Service (point the JAR resource to a new version of the JAR), you need to redeploy the EJB. You will then have to edit any EJB service that uses this JAR, and reselect the JAR resource, save, and activate. That will repackage the EJB business service to use the new JAR.

Java callout actions and Tuxedo-based services, however, will automatically pick up the new JAR.

8.4 Editing JAR Dependencies

In Oracle Service Bus, JAR resources can contain classes that depend on other classes in different JARs. Use the Edit JAR Dependencies page to add or remove referenced JAR resources.

1. If you have not already done so, click **Create** to create a new session, or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the JAR, as described in Section 8.1, "Locating JARs."

3. Click the JAR name.

On the View a JAR Resource - JAR Configuration page, a numeric indicator on the Dependencies tab displays the number of referenced JARs configured for this JAR resource, or a zero if there are none.

4. Select the **Dependencies** tab to display referenced JARs or to add them.

The JAR dependencies page includes the information shown in Table 8–4.

Table 8–4   JAR Dependencies Information

Property	Description
Name	The name of the referenced JAR. The name is a link to the View a JAR Resource - JAR Configuration page. See Section 8.3, "Editing JARs."
Path	The path is the project name and the name of the folder in which the JAR resides.

5. To add or remove referenced JARs, click **Edit**.

The Edit JAR dependencies page includes the information shown in Table 8–5.

Table 8–5   JAR Dependencies Details

Property	Description
Manifest Class Path	Class-Path entries in the JAR manifest file.
Name	The name and path of the referenced JAR resource.
Options	■ Use the Up and Down arrows to change the referenced JARs order of precedence. ■ Click the **Delete** icon to remove a referenced JAR.

6. Click **Add JARS**.

7. In the JARs Browser, locate or select referenced JAR resources and click **Submit**.

For example, if you are editing JAR A and adding JARs B and C, JAR A references (depends on) JAR B and JAR A references (depends on) JAR C.

Use the arrows in the Options column to reorder the hierarchy of referenced JARs or the **Delete** icon to remove them.

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

8.5 Deleting JARs

To delete a JAR:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **JARs**.

3. Click the **Delete** icon in the Options field of the JAR you want to delete. The JAR is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning

confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

9

MQ Connections

This chapter describes how to create, locate, edit, and delete MQ Connection resources using the Oracle Service Bus Administration Console.

> **Note:** See Chapter 1, "Interoperability Scenarios and Considerations" for the supported versions of WebSphere MQ on Oracle Service Bus. The WebSphere MQ client library is not bundled with the Oracle Service Bus installer. Get the WebSphere MQ client library from your WebSphere MQ installation.

MQ Connections are sharable resources that can be reused across multiple MQ proxy and business services. MQ proxy and business services must connect to an MQ queue manager before accessing an MQ queue. MQ Connection resources provide the connection parameters required for connecting to a MQ queue manager.

Each MQ Connection resource has a connection pool. Every business or proxy service using a given MQ Connection resource to get a connection to a given queue manager uses the same connection pool that was created for that resource. Thus, multiple business services and proxy services using the same queue manager share a connection pool.

To learn more about Oracle Service Bus MQ Connection resources and native MQ transports, see "MQ Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

To learn more about WebSphere MQ Fundamentals, see
http://www.redbooks.ibm.com/redbooks/SG247128/wwhelp/wwhimpl/java/html/wwhelp.htm.

9.1 Locating MQ Connections

Locating MQ Connection resources:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the MQ Connection resource.

   - Select **Resource Browser** > **MQ Connections**. The Summary of MQ Connections page displays the information shown in Table 9–1. For a more detailed description of the properties, see Section 9.3, "Editing MQ Connections."

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, enter the name and path of the search target(s), then click **Search**.

3. Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - Filter by **Queue Manager Name**. Enter any string, including wildcard characters, then click **Search**.

   - Filter by **Queue Manager Channel Name**. Enter any string, including wildcard characters, then click **Search**.

   - Click **View All** to remove the search filters and display all MQ Connection resources.

Table 9–1   MQ Connection Details

Property	Description
MQ Connection Name	The name assigned to the MQ Connection resource. The name is a link to the MQ Connection Configuration page. See Section 9.3, "Editing MQ Connections."
Path	The path is the project name and the name of the folder in which the MQ Connection resource resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Click the **Delete** icon to delete a specific MQ Connection resource. See Section 9.4, "Deleting MQ Connections."

9.2 Adding MQ Connections

Before you begin

See Chapter 1, "Interoperability Scenarios and Considerations" for the supported versions of WebSphere MQ on Oracle Service Bus. The WebSphere MQ client library is not bundled with the Oracle Service Bus installer. Get the WebSphere MQ client library from your WebSphere MQ installation.

To configure and use MQ Connection resources and the MQ transport in Oracle Service Bus, you must add the MQ client library to the classpath. One option is to copy the com.ibm.mq.jar library to your DOMAIN_NAME/lib directory, where *DOMAIN_NAME* is the name of the directory in which you located the domain, typically *MW_HOME*/user_projects/domains/*DOMAIN_NAME*.

To add an MQ Connection resource

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the MQ Connection resource. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **MQ Connection** from under Utility to display the Create a New MQ Connection Resource page.

   Specify the connection parameters shown in Table 9–2.

Table 9–2   MQ Connection Parameters

Property	To create or edit...
Resource Name	Enter a unique name for this MQ Connection resource.
	Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance. Do not include spaces in the name.
Resource Description	Enter a description for the MQ Connection resource.
Connection Type	Select one of the following modes for connecting to the MQ queue manager:
	■ `tcp mode`—Use TCP/IP to connect to a queue manager that does not reside on the same computer as Oracle Service Bus.
	■ `binding mode`—Use the bindings mode to connect to a queue manager that is located on the same computer as Oracle Service Bus.
MQ Host Name	For `tcp mode` connections only:
	Enter the host name of the MQ queue manager.
MQ Port Number	For `tcp mode` connections only:
	Enter the port number of the MQ queue manager listener.
MQ Queue Manager Name	Enter the name of the MQ queue manager to which to connect.
Queue Manager CCSID	For `tcp mode` connections only:
	The coded character set identifier (CCSID) to be used when establishing a connection. The CCSID is used mainly for internationalization support.
	To learn more, see WebSphere MQ Fundamentals at `http://www.redbooks.ibm.com/redbooks/SG247128/wwhelp/wwhimpl/java/html/wwhelp.htm`.
MQ Queue Manager Channel Name	For `tcp mode` connections only:
	Enter the queue manager server connection channel name.
SSL Required	For `tcp mode` connections:
	Select the check box to use SSL for sending messages. Only server-side SSL is enabled unless the **2-way SSL Required** option is *also* selected.
Cipher Suite	This option is available only when the **SSL Required** check box is selected.
	Select the Cipher Suite algorithm to be used by SSL.
	The Cipher Suite algorithm is used to encrypt and decrypt message communications between the WebSphere MQ server and the WebSphere MQ client. Thus a Cipher Suite algorithm must be specified when using SSL to communicate with a WebSphere MQ server.
2-way SSL Required	This option is available only when the **SSL Required** check box is selected.
	Select the check box to enable both client-side and server-side SSL authentication. Clear the check box for 1-way SSL for only server-side authentication.

Table 9–2 (Cont.) MQ Connection Parameters

Property	To create or edit...
Reference to the Service Key Provider	If you selected **2-way SSL Required**, you must provide a reference to the service key provider for obtaining the appropriate key store and trust store information. A service key provider contains Public Key Infrastructure (PKI) credentials that proxy services use for decrypting inbound SOAP messages and for outbound authentication and digital signatures.
	Enter the path (project/folder) and name of a service key provider, or click **Browse** to select one from the Select Service Key Provider page.
	For more information about using service providers, see "Service Key Providers" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus.*
Reference to the Static Service Account	For `tcp mode` connections only:
	Required for user name and password authentication. Enter the path (project/folder) and name of a static service account, or click **Browse** to select service accounts from a browser.
	A service account provides a user name and password that proxy services and business services use for outbound authentication or authentication to a local or remote resource.
WebSphere MQ Version	Select the WebSphere MQ version:
	- 5.3
	- 6.0
	- 7.0
	Note: Versions 5.3 and 6.0 are no longer supported by IBM. These versions are deprecated in Oracle Service Bus.
MQ Connection Pool Size	Enter the number of MQ connections to be stored in the connection pool for reuse.
MQ Connection Timeout	Enter the time interval in seconds after which unused connections are destroyed. The default is 1800 seconds.
MQ Connection Max Wait	Enter the maximum amount of time (in milliseconds) to wait for a connection to become available. If a connection is not made within that time interval, Oracle Service Bus throws an exception. The default is 3000 milliseconds.
XA Enabled	Select this option if the transactions handled by this connection are distributed (XA) transactions.
	For XA transactions, the MQ queue must be configured for persistence. If you are using version 5.3 or 6.0 (both deprecated), add the `com.ibm.mqetclient.jar` file to your classpath.

4. Click **Save**. The MQ Connection resource is created and saved in the current session.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

9.3 Editing MQ Connections

To edit an MQ Connection resource:

1. If you have not already done so, click Create to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the MQ Connection resource, as described in Section 9.1, "Locating MQ Connections."

3. Click the MQ Connection resource name.

   The View MQ Connection - Configuration page displays the information shown in Table 9–3.

Table 9–3   MQ Connection Details

Property	Description
Last Modified By	The user who created this MQ Connection resource or imported it into the configuration.
Last Modified On	The date and time that the user created this MQ Connection resource or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this MQ Connection resource references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this MQ Connection resource. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this MQ Connection resource, if one exists.

4. To make a change to the fields, click **Edit**. For descriptions of the fields, see Section 9.2, "Adding MQ Connections."

   You cannot change the Resource Name field.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

9.4 Deleting MQ Connections

To delete an MQ Connection resource:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **MQ Connections** to display the Summary of MQ Connections page.

3. Click the **Delete** icon in the Options field of the MQ Connection resource you want to delete. The MQ Connection resource is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

10

Split-Joins

This chapter describes how to create, locate, edit, and delete Split-Join resources using the Oracle Service Bus Administration Console.

Split-Joins let you send message invocations in parallel (in addition to sequentially) and to aggregate the responses. To learn more about Split-Joins, see "Improving Service Performance with Split-Join" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*. See also, "Working with Split-Join" and the "Split-Join User Interface Reference" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

You access Split-Joins through Flow transport-typed business services. For more information, see the Section 19.2.10, "Flow Transport Configuration Page."

10.1 Locating Split-Joins

To locate Split-Joins:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the Split-Join.

   - Select **Resource Browser** > **Split-Joins**. The Summary of Split-Joins page displays the information shown in Table 10–1. For a more detailed description of the properties, see Section 10.3, "Editing Split-Joins."

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, enter the name and path of the search target(s), then click **Search**.

3. Click **View All** to remove the search filters and display all Split-Joins.

Table 10–1   Split-Join Details

Property	Description
Split-Join Name	The name assigned to the Split-Join. The name is a link to the Split-Join Configuration page. See Section 10.3, "Editing Split-Joins."
Path	The path is the project name and the name of the folder in which the Split-Join resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Click the **Delete** icon to delete a specific Split-Join. See Section 10.4, "Deleting Split-Joins."

10.2 Adding Split-Joins

Before you begin

Split-Joins require a .flow file which is created when you create a Split-Join using the Oracle Service Bus plug-ins for Eclipse or when you import a Split-Join configuration JAR into the plug-in. The .flow file might have references to proxy services, business services, or to external WSDL resources. Resources on which the Split-Join depends must be present on the server before activating the Split-Join.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the Split-Join. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Split-Join** to display the Create a New Split-Join Resource page.

   Specify the parameters shown in Table 10–2.

Table 10–2   Split-Join Parameters

Property	To create or edit...
Resource Name	Enter a unique name for this Split-Join.
	Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.
Resource Description	Enter a description for the Split-Join.
Split-Join	Click **Browse** to locate a .flow file on your system, typically located under *MW_HOME*\user_projects\workspaces*your_workspace*.
Dispatch Policy	Select the instance of WebLogic Server Work Manager used to create threads (parallel branches) to execute the Split-Join. The default Work Manager is used if no other Work Manager exists.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

4. Click **Save**. The Split-Join is created and saved in the current session.

   > **Note:** Resources on which the Split-Join depends must be present before activating the Split-Join.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

10.3 Editing Split-Joins

To modify Split-Join properties:

1. If you have not already done so, click Create to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the Split-Join, as described in Section 10.1, "Locating Split-Joins."

3. Click the Split-Join name.

   The View a Split-Join - Configuration page displays the information shown in Table 10–3.

Table 10–3   Split-Join Details

Property	Description
Last Modified By	The user who created this Split-Join or imported it into the configuration.
Last Modified On	The date and time that the user created this Split-Join or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this Split-Join references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this Split-Join. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this Split-Join, if one exists.

4. You can make the following operational settings to a Split-Join:

Table 10–4   Split-Join Operational Settings

Property	Description
Logs	To capture log information in a Split-Join, select **Enable Logging** and select the logging level you want to capture.
Execution Tracing	To enable runtime tracing, select **Enabled**.

5. Click **Update** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

10.4 Deleting Split-Joins

To delete a Split-Join:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Split-Joins** to display the Summary of Split-Joins page.

3. Click the **Delete** icon in the Options field of the Split-Join you want to delete. The Split-Join is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

11

XQuery Transformations

This chapter describes how to create, locate, edit, and delete XQuery Transformation resources using the Oracle Service Bus Administration Console.

XQuery transformation maps can describe XML-to-XML, XML to non-XML, and non-XML to XML mappings.

11.1 Locating XQuery Transformations

To locate XQuery Transformations:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View Page or the Project/Folder View page. Then navigate through projects and folders to find the XQuery.

   - Select **Resource Browser** > **XQueries**. The Summary of XQueries page displays the information shown in Table 11–1.

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, under Search, enter the target's name, path, or both, and then click the **Search** button.

   The path is the project name and the name of the folder in which the XQuery resides.

   Wildcard characters * and ? are allowed. Search is case-sensitive.

   Click **View All** to display all XQueries in the domain. This clears the search parameters from the previous search.

Table 11–1   XQuery Transformation Details

Property	Description
XQuery Name	The name assigned to the XQuery transformation. Click the name to display the View XQuery Transformation Details page. See Section 11.3, "Editing XQuery Transformations."
Path	The path is the project name and the name of the folder in which the XQuery resides, for example, `MyProject/MyFolder/MyResource`. Click the path of an XQuery to display the Section 4.3, "Project/Folder View Page." that contains it.
Actions	Click the **Launch Test Console** icon to invoke the Test Console, which you use to validate and test the design of your services and transformations. For XQuery transformations, you can use the Test Console whether you are inside or outside a session. To learn more, see Section 4.2, "Projects View Page."

Table 11–1 (Cont.) XQuery Transformation Details

Property	Description
Options	If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."
	Click the **Delete** icon to delete the XQuery. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. For more information, see Section 11.4, "Deleting an XQuery Transformation."

11.2 Adding XQuery Transformations

To add an XQuery Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** to display the Section 4.2, "Projects View Page." or Section 4.3, "Project/Folder View Page."

3. Navigate to the project or a folder to which you want to add the XQuery transformation.

4. From the **Create Resource** list, select **XQuery**. The Create a New XQuery page is displayed.

5. In the **Resource Name** field, enter a unique name for this XQuery transformation.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

6. In the **Resource Description** field, enter a description for the XQuery transformation.

7. In the **XQuery** field, do one of the following:

   - Enter text for the new XQuery transformation.

   - Click **Browse** to locate and import an existing XQuery transformation.

   - Copy and paste text from an existing XQuery transformation into this field.

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, select **Activate** under Change Center.

11.3 Editing XQuery Transformations

To edit an XQuery Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the XQuery transformation, as described in Section 11.1, "Locating XQuery Transformations."

3. Click the XQuery transformation name. The View XQuery Transformation Details page displays the information shown in Table 11–2.

Table 11–2   XQuery Transformation Resource Details

Property	Description
Last Modified By	The user who created this XQuery transformation or imported it into the configuration.
Last Modified On	The date and time that the user created this XQuery transformation or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this XQuery transformation references. If such references exist, click the numeric link to view a list of the objects. To learn more, see Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this XQuery transformation. If such references exist, click the numeric link to view a list of the objects. For example, if you assigned this XQuery transformation to a variable in the Message Flow of a specific proxy service, the proxy service is listed as a reference when you click the link.To learn more, see Section 4.22, "Viewing References to Resources."
Description	A description of this XQuery transformation, if one exists.
XQuery	The full text of the XQuery transformation.
XQuery Variables	The XQuery variables selected for this XQuery transformation.

4. To make a change to the fields, click **Edit**. See Section 11.2, "Adding XQuery Transformations" for a description of the fields.

You cannot change the Resource Name field.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

11.4 Deleting an XQuery Transformation

To delete an XQuery Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **XQueries**. The Summary of XQueries page is displayed.

3. Click the **Delete** icon in the row of the XQuery transformation you want to delete. The XQuery transformation is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

12

Custom WS-Policies

This chapter describes how to create, locate, edit, and delete Custom WS-Policy resources using the Oracle Service Bus Administration Console.

> **Note:** In future releases of Oracle Service Bus, Oracle Web Services Manager policies will replace WLS 9.2 policies. While this version of Oracle Service Bus continues to support WLS 9.2 policies, you should consider using Oracle Web Services Manager policies for new service creation to prepare for the eventual deprecation of WLS 9.2 policy support. For more information, see "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Web Services Policy Framework (WS-Policy) is an extensible XML-based framework that extends the configuration of a Web Service with domain specific security assertions and specifies the security requirements, expectations, and capabilities of the Web Service. In Oracle Service Bus, one of the primary uses of WS-Policy is configuring message-level security in proxy services and business services.

12.1 Using Oracle Service Bus WS-Policies

Oracle Service Bus includes three simple WS-Policy files that you can use to require clients to authorize, digitally encrypt, or digitally sign SOAP messages: `Auth.xml`, `Encrypt.xml`, and `Sign.xml`. Oracle recommends that unless you have specific security needs, you use these pre-packaged files as often as possible.

For more information about using these policies, see "Using WS-Policy in Oracle Service Bus Proxy and Business Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

12.1.1 Using Custom WS-Policies

If the Oracle Service Bus WS-Policy statements do not meet your security needs, you can write your own WS-Policies (custom WS-Policies), import them to Oracle Service Bus, and refer to them from the WSDL. (The Oracle Service Bus WS-Policy statements are read-only.)

12.2 Locating Custom WS-Policies

The Summary of WS-Policies page lists the custom Web Service Policies (WS-Policies) that you have added to the current Oracle Service Bus domain. The Oracle Service Bus

Administration Console does not display the Oracle Service Bus WS-Policies, which are pre-packaged and available to all Oracle Service Bus domains.

1. Do either of the following:

   ■ Select **Project Explorer** to display the Projects View Page or the Project/Folder View page. Then navigate through projects and folders to find the WS-Policy.

   ■ Select **Resource Browser** > **WS-Policies**. The Summary of WS-Policies page displays the information shown in Table 12–1. For a more detailed description of the properties, see Section 12.2, "Locating Custom WS-Policies."

2. To search for a custom WS-Policy, enter part or all of the WS-Policy name in the **Name** field. You can also enter part or all of the WS-Policy project name and folder in the **Path** fields. Click **Search**.

   Click **View All** to remove the search filters and display all WS-Policies.

Table 12–1   WS-Policy Information

Property	Description
WS-Policy Name	The unique name assigned to the WS-Policy. Click the name to see the View WS-Policy Details page. See Section 12.4, "Editing Custom WS-Policies."
Path	The project name and the name of the folder in which the WS-Policy resides. Click the name to see the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Contains a **Delete** icon. If a business service or proxy service has been configured to use a WS-Policy, a Deletion Warning icon indicates that you can delete the WS-Policy with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted WS-Policy. See Section 12.5, "Deleting Custom WS-Policies."

12.3 Adding Custom WS-Policies

You can either import an XML file that contains your WS-Policy or write the WS-Policy directly in the Oracle Service Bus Administration Console.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the WS-Policy. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **WS-Policy** from under Interface to display the Create a New WS-Policy page.

4. In the **Resource Name** field, enter a unique name for this WS-Policy.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description of this WS-Policy.

6. In the **Policy** field, do one of the following:

   ■ Click **Browse** to locate and import an XML file that contains your custom WS-Policy.

   ■ Enter text for the new WS-Policy.

   ■ Copy and paste text from an existing WS-Policy into this field.

7. Click **Save**. The WS-Policy is saved in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

12.4 Editing Custom WS-Policies

Use the View WS-Policy Details page to view and change details of a specific custom Web Service Policy (WS-Policy). The Oracle Service Bus Administration Console does not display the Oracle Service Bus WS-Policies, which are read-only.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the WS-Policy, as described in Section 12.2, "Locating Custom WS-Policies."

3. Click the WS-Policy name. The View WS-Policy Details page displays the information shown in Table 12–2.

Table 12–2   WS-Policy Details

Property	Description
Last Modified By	The user who created this WS-Policy or imported it into the configuration.
Last Modified On	The date and time that the user created this WS-Policy or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this WS-Policy references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this WS-Policy. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this WS-Policy, if one exists.
Policy	The text for this WS-Policy.

4. To make a change to the fields, click **Edit**. See Section 12.3, "Adding Custom WS-Policies" for descriptions of the fields.

   You cannot change the **Resource Name** field.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

12.5 Deleting Custom WS-Policies

If any business service or proxy service is configured to use the WS-Policy, remove the WS-Policy from the business service or proxy service.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **WS-Policies**.

3. Click the **Delete** icon in the **Options** field of the WS-Policy you want to delete.

The WS-Policy is deleted in the current session. If a business service or proxy service has been configured to use a WS-Policy, a Deletion Warning icon indicates that you can delete the WS-Policy with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted WS-Policy.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

13

JCA Bindings

This chapter describes how to create, locate, edit, and delete JCA Binding resources using the Oracle Service Bus Administration Console.

Oracle SOA Suite lets you integrate with back-end Enterprise Information Systems (EIS) using J2EE Connector Architecture (JCA) adapters. Using Oracle JDeveloper and other supported JCA development environments, you configure the adapters and create the JCA adapter resources (.jca files and abstract WSDLs) used for the back-end integration.

You can communicate with JCA-compliant EIS components through Oracle Service Bus (OSB), which lets you consume the JCA resources in the form of JCA Binding resources. With JCA Bindings, you can generate proxy and business services that use the Oracle Service Bus JCA transport to communicate with the EIS applications through the JCA adapters.

This chapter contains the following topics:

- Section 13.1, "Locating JCA Bindings"
- Section 13.2, "Adding JCA Bindings"
- Section 13.3, "Editing JCA Bindings"
- Section 13.4, "Deleting JCA Bindings"

For more information on the JCA transport, see "JCA Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

For more information on the Oracle SOA Suite JCA adapters, see the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

13.1 Locating JCA Bindings

The Summary of JCA Bindings page lists the JCA Bindings that you have added to the Oracle Service Bus domain.

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the JCA Binding.

   - Select **Resource Browser** > **JCA Bindings**. The Summary of JCA Bindings page displays the information shown in Table 13–1. For a more detailed description of the properties, see Section 13.1, "Locating JCA Bindings."

2. To locate a specific JCA Binding:

- To restrict the number of items in the list, you can filter by name and path before you click **Search**.

  Path is the project name and the name of the folder(s) in which the resource resides.

- Click **View All** to remove the search filters and display all JCA Bindings.

Table 13–1   JCA Binding Information

Property	Description
JCA Binding Name	The unique name assigned to the JCA Binding. Click the name to see the View JCA Binding Details page. See Section 13.3, "Editing JCA Bindings."
Path	The project name and the name of the folder in which the JCA Binding resides. Click the name to see the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Action	Contains a Generate icon that lets you generate an Oracle Service Bus WSDL and JCA service from the JCA Binding.
Options	Contains a **Delete** icon. If a business service or proxy service has been configured to use a JCA Binding, a Deletion Warning icon indicates that you can delete the JCA Binding with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted JCA Binding. See Section 13.4, "Deleting JCA Bindings."

13.2 Adding JCA Bindings

The easiest way to create JCA Bindings in Oracle Service Bus is by bulk importing the JCA resources you have created (.jca file, associated WSDL, and TopLink or EclipseLink mapping file). On bulk import, Oracle Service Bus automatically generates a JCA Binding out of .jca files, WSDL resources out of the associated JCA WSDLs, and XML Document resources out of mapping files, and maintains the dependencies among the files.

For more information on importing, see Section 4.15, "Loading Resources from a Zip File.", Section 4.17, "Loading Resources from a URL" and Section 29.1, "Importing Resources."

Once you import your JCA resources, you can generate proxy or business services out of them, as described in Section 20.1.1, "Generating a Proxy Service from a JCA Binding Resource" and Section 19.1.1, "Generating a Business Service from a JCA Binding Resource."

If you do not bulk import your JCA resources, you can manually create JCA Bindings, described in the following steps:

1. In Oracle JDeveloper create a JCA service WSDL, a corresponding JCA (.jca) file, and, if necessary, other resources such as TopLink or EclipseLink mapping files and XSLT files.

   For more information, see the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

2. If you have not already done so, click **Create** to create a new session in the OSB Administration Console, or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

3. In the OSB Administration Console, create or import the JCA resources you created in Oracle JDeveloper, such as the WSDL, XML, and XSLT resources.

4. Create the JCA Binding.

   a. Select **Project Explorer**, then select the project or folder location for the JCA Binding. The Project/Folder View page is displayed.

   b. From the **Create Resource** field, select **JCA Binding** to display the Create a New JCA Binding Resource page.

   c. In the **JCA File** field, click **Browse** to select the JCA file.

      The Resource Name field is automatically populated with the file name (minus the file extension).

      Instead of selecting a file, you can copy and past the contents of a JCA file in the JCA File field text area.

   d. In the **Resource Name** field, enter or change the name of the JCA Binding. JCA Binding names must be unique.

      Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

   e. In the **Resource Description** field, enter a description of this JCA Binding.

   f. Click **Save**. The JCA Binding is saved in the current session.

      A validation warning is displayed in the If the JCA Binding does not yet reference required resources, such as a WSDL and a mapping file.

5. Click the name of the new JCA Binding to display it in the View a JCA Binding page.

6. Click **Edit References**.

7. On the Edit the Reference of a JCA Binding Resource page, **Browse** to select the JCA Binding's reference resources, such as the WSDL, a TopLink/EclipseLink mapping file, and an XSLT file.

8. Click **Save**.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

After you add JCA Binding resources and configure the resource dependencies (associating a WSDL)—or if you have imported JCA resources that maintain their dependencies—you can generate proxy and business services from JCA Bindings. For more information, see Section 20.1.1, "Generating a Proxy Service from a JCA Binding Resource" and Section 19.1.1, "Generating a Business Service from a JCA Binding Resource."

13.3 Editing JCA Bindings

Use the View JCA Binding Details page to view and change details of a JCA Binding.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the JCA Binding, as described in Section 13.1, "Locating JCA Bindings."

3. Click the JCA Binding name. The View a JCA Binding page displays the information shown in Table 13–2.

Table 13–2   JCA Binding Details

Property	Description
Last Modified By	The user who created this JCA Binding or imported it into the configuration.
Last Modified On	The date and time that the user created this JCA Binding or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."]
References	Provides a link to the WSDL resource that the JCA Binding references. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this JCA Binding. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Action icon	With the Change Center in edit mode, Action lets you generate an Oracle Service Bus WSDL and JCA service from the JCA Binding.
JCA File	Displays a read-only view of the JCA Binding.
Abstract WSDL	Displays the JCA Binding's associated WSDL. If the JCA Binding does not reference a WSDL, click **Edit References** to reference the associated WSDL.
TopLink XML	Displays the JCA Binding's required TopLink or EclipseLink mapping file. If the JCA Binding resource does not reference the required mapping file, click **Edit References** to select the required mapping file. This field appears only if the JCA file declares an XML resource.
XSLT	Displays the JCA Binding's required XSLT file. If the JCA Binding resource does not reference a required XSLT resource, click **Edit References** to select the required XSLT file. This field appears only if the JCA file declares an XSLT resource, such as for a socket adapter.

4. To modify the Description or JCA File, click **Edit**.

You cannot change the Resource Name.

5. Click **Save** to commit the updates in the current session.

6. To change the references to the JCA Binding's required resources, or to fix unresolved references, click **Edit References** and select the appropriate resources.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

13.4 Deleting JCA Bindings

If any business service or proxy service is based on the JCA Binding, remove the JCA Binding from the business service or proxy service before deleting the JCA Binding.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **JCA Bindings**.

3. Click the **Delete** icon in the **Options** field of the JCA Binding you want to delete.

The JCA Binding is deleted in the current session. If a business service or proxy service has been configured to use a JCA Binding, a Deletion Warning icon

indicates that you can delete the JCA Binding with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted JCA Binding.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

14

XML Documents

This chapter describes how to create, locate, edit, and delete XML Document resources using the Oracle Service Bus Administration Console.

Use XML Document resources to store XML files for use in proxy or business service configurations. For example, create XML Document resources for TopLink or EclipseLink mapping files needed in JCA proxy or business services that communicate with JCA-compliant systems.

For more information on JCA services, see Chapter 13, "JCA Bindings."

This chapter contains the following sections:

- Section 14.1, "Locating XML Documents"
- Section 14.2, "Adding XML Documents"
- Section 14.3, "Editing XML Documents"
- Section 14.4, "Deleting XML Documents"

14.1 Locating XML Documents

The Summary of XML Documents page lists the XML Documents that you have added to the Oracle Service Bus domain.

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the XML Document.

   - Select **Resource Browser** > **XML Documents**. The Summary of XML Documents page displays the information shown in Table 14–1. For a more detailed description of the properties, see Section 14.3, "Editing XML Documents."

2. To locate a specific XML Document:

   - To restrict the number of items in the list, you can filter by name and path before you click **Search**.

     Path is the project name and the name of the folder(s) in which the resource resides.

   - Click **View All** to remove the search filters and display all XML Documents.

Table 14–1   XML Documents Details

Property	Description
XML Document	The unique name assigned to the XML Document. The name is a link to the View the XML Document page. See Section 14.3, "Editing XML Documents."
Path	The path is the project name and the name of the folder in which the XML Document resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Click the **Delete** icon to delete a specific XML Documents. See Section 14.4, "Deleting XML Documents."

14.2 Adding XML Documents

The easiest way to create XML Documents in Oracle Service Bus is by bulk importing them. For example, if you bulk import JCA resources you have created (.jca file, associated WSDL, and TopLink or EclipseLink mapping file), Oracle Service Bus automatically generates XML Document resources out of mapping files and maintains the dependencies among resource files.

For more information on importing, see Section 4.15, "Loading Resources from a Zip File.", Section 4.17, "Loading Resources from a URL." and Section 29.1, "Importing Resources."

If you do not bulk import, you can manually create XML Documents, described in the following steps:

1. If you have not already done so, click **Create** to create a new session, or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the XML Document. The Project/Folder View page is displayed.

3. From the **Create Resource** field, select **XML Document** to display the Create a New XML Document page.

4. In the **XML File** field, click **Browse** and select the XML file you want to use.

   The Resource Name field is automatically populated with the file name (minus the file extension).

   Instead of selecting a file, you can copy and past the contents of an XML file in the XML File field text area.

5. In the **Resource Name** field, enter or change the name of the XML document. XML Document names must be unique.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

6. In the **Resource Description** field, enter a description for the XML Document.

7. Click **Save**. The new XML Document is saved in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

14.3 Editing XML Documents

Use the Edit an XML Document page to view and change details of a JCA Binding.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the XML Document, as described in Section 14.1, "Locating XML Documents."

3. Click the XML Document name. The View the XML Document page displays the information shown in Table 14–2.

Table 14–2   XML Document Details

Property	Description
Last Modified By	The user who created this XML Document or imported it into the configuration.
Last Modified On	The date and time that the user created this XML Document or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this XML Document references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this XML Document. If such references exist, click the numeric link to view a list of the objects. For example, if the XML file is a TopLink file referenced by a JCA Binding, the JCA Binding is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this XML Document, if one exists.

The a read-only view of the XML Document is displayed by default.

4. To edit the XML Document, click **Edit**. See Section 14.2, "Adding XML Documents" for field descriptions.

5. Make the appropriate edits and click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

14.4 Deleting XML Documents

If any resources such as JCA Bindings reference the XML Document you want to delete, remove those references before deleting the XML Document.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **XML Documents**.

3. Click the **Delete** icon in the **Options** field of the XML Document you want to delete. The XML Document is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** in the Change Center.

15

XSL Transformations

This chapter describes how to create, locate, edit, delete, and fix unresolved references to XSL Transformation resources using the Oracle Service Bus Administration Console.

Transformation maps describe the mapping between two data types. eXtensible Stylesheet Language Transformation (XSLT) maps describe XML-to-XML mappings.

15.1 Locating XSL Transformations

To locate XSL Transformations:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the XSL transformation.

   - Select **Resource Browser** > **XSLTs**. The **Summary of XSLTs** page displays the information shown in Table 15–1.

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, under **Search**, enter the target's name, path, or both, and then click the **Search** button.

   The path is the project name and the name of the folder in which the XSLT resides.

   Wildcard characters * and ? are allowed. Search is case-sensitive.

   Click **View All** to display all XSLTs in the domain. This clears the search parameters from the previous search.

Table 15–1   XSL Transformation Details

Property	Description
XSLT Name	The name assigned to the XSL transformation. Click the name to display the View XSL Transformation Details page.
	See Section 15.3, "Editing XSL Transformations."
Path	The path is the project name and the name of the folder in which the XSL transformation resides, for example, `MyProject/MyFolder/MyResource`.
	Click the path of an XSLT to display the Project/Folder View page that contains it.

Table 15–1   (Cont.) XSL Transformation Details

Property	Description
Actions	Click the **Launch Test Console** icon to invoke the Test Console, which you use to validate and test the design of your services and transformations. For XSL transformations, you can use the Test Console whether you are inside or outside a session. To learn more, see Section 4.2, "Projects View Page."
Options	If you have not already done so, click Create to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."
	Click the **Delete** icon to delete the XSLT. A **Deletion Warning** icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. For more information, see Section 15.4, "Deleting an XSL Transformation."

15.2 Adding an XSL Transformation

To add an XSL Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer** to display the Projects View page or Project/Folder View page.

3. Navigate to the project or a folder to which you want to add the XSL transformation.

4. From the **Create Resource** list, select **XSLT** to display the **Create a New XSLT** page.

5. In the **Resource Name** field, enter a unique name for this XSL transformation.

    Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

6. In the **Resource Description** field, enter a description for the XSL transformation.

7. In the **XSLT** field, do one of the following:

    ■   Enter text for the new XSL transformation.

    ■   Click **Browse** to locate and import an existing XSL transformation.

    ■   Copy and paste text from an existing XSL transformation into this field.

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under **Change Center**.

15.3 Editing XSL Transformations

To edit an XSL Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the XSL transformation, as described in Section 15.1, "Locating XSL Transformations."

3. Click the XSL transformation name. The **View XSL Transformation Details** page displays the structural view of the XSLT by default, which includes the information shown in Table 15–2.

Table 15–2   XSL Transformation Resource Details

Property	Description
Last Modified By	The user who created this XSL transformation or imported it into the configuration.
Last Modified On	The date and time that the user created this XSL transformation or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this XSL transformation references. If such references exist, click the numeric link to view a list of the objects. To learn more, see Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this XSL transformation. If such references exist, click the numeric link to view a list of the objects. For example, if you assigned this XSL transformation to a variable in the Message Flow of a specific proxy service, the proxy service is listed as a reference when you click the link. To learn more, see Section 4.22, "Viewing References to Resources."
Description	A description of this XSL transformation.
Target Namespace	The namespace used to qualify any of the definitions included in the XSL transformation.
XSLT Dependencies	The attributes and groups associated with the XSLT: the XSLT location.
State	The status of the XSL transformation: ■ Valid All of the XSL transformation locations included by the current XSL transformation are specified and are valid. Furthermore, all of the locations for nested XSL transformations (XSL transformations subsequently included by the included XSL transformations) are specified and are valid. ■ Invalid One or more of the locations for XSL transformations included by the current XSL transformation is not specified or is not valid. Additionally, one or more of the nested XSL transformations (XSL transformations subsequently included by the included XSL transformations) may not be specified or may not be valid.To resolve an invalid XSL transformation, click **Edit References**.To learn more, see Section 15.5, "Resolving Unresolved XSL Transformation References."

4. Click **Text view** to display a text view of the XSL transformation details. The text view includes the information shown in Table 15–3.

Table 15–3   XSL Transformation Text Details

Property	Description
Target Namespace	The namespace used to qualify any of the dependencies included in the XSL transformation.

Table 15–3   (Cont.) XSL Transformation Text Details

Property	Description
XSLT	The full text of the XSL transformation.

5. To make a change to the fields, click **Edit**. See Section 15.2, "Adding an XSL Transformation" for a description of the fields.

   You cannot change the **Resource Name** field.

6. Click **Save** to commit the updates in the current session.

15.4 Deleting an XSL Transformation

To delete an XSL Transformation:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **XSLTs** to display the **Summary of XSLTs** page.

3. Click the **Delete** icon in the row of the XSL transformation you want to delete. The XSL transformation is deleted in the current session. A **Deletion Warning** icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under **Change Center**.

15.5 Resolving Unresolved XSL Transformation References

To resolve XSL Transformation references:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **XSLTs** to display the **Summary of XSLTs** page.

3. In the **XSLT Name** column, click the appropriate XSLT name to view details of the unresolved reference. The **View XSL Transformation Details** page is displayed.

4. Click **Edit References** to display the **Edit the References of an XSL Transformation Resource** page.

5. In the **Resource Type** field, select **XSLT**.

6. Click **Browse**. The XSLT Browser is displayed.

7. In the **XSLT Browser**, select an XSLT, then select a dependency from the **Dependencies** pane.

8. Click **Submit**. The XSLT you selected is displayed in the **Resource Name** field.

9. Click **Save**. The **View XSLT Details** page is displayed.

10. Click **OK**. The state of the XSL transformation is displayed as **Valid**.

11. To end the session and deploy the configuration to the runtime, click **Activate** under **Change Center**.

16

MFLs

This chapter describes how to create, locate, edit, and delete MFL resources using the Oracle Service Bus Administration Console.

A Message Format Language (MFL) document is a specialized XML document used to describe the layout of binary data. It is an Oracle proprietary language used to define rules to transform formatted binary data into XML data. An MFL document conforms to the `mfl.dtd`, which includes elements and attributes used to describe each field of data, as well as groupings of fields (groups), repetition, and aggregation.

When you create a business services or proxy services of Messaging Service type, you can select MFL types as the request message type or the response message type of the service.

You use Oracle Format Builder to create MFLs. When you define the hierarchy of a binary record, the layout of fields, and the grouping of fields and groups, the information is saved as an MFL document that can then be used to perform runtime translations. An MFL document can also be used in Format Builder to generate the corresponding DTD that describes its content model. To learn how to use the Format Builder, see "Format Builder" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

16.1 Locating MFLs

To locate MFLs:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the MFL.

   - Select **Resource Browser** > **MFLs**. The **Summary of MFL Files** page displays the information shown in Table 16–1. For a more detailed description of the properties, see Section 16.3, "Editing MFLs."

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, enter the name and path of the search target(s), then click **Search**.

   The path is the project name and the name of the folder in which the MFL file resides.

   Click **View All** to remove the search filters and display all MFL files.

Table 16–1   MFL Details

Property	Description
MFL File Name	The unique name assigned to the MFL. The name is a link to the View MFL Resource Details page. See Section 16.3, "Editing MFLs."
Path	The path is the project name and the name of the folder in which the MFL file resides. It is a link to the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Actions	Click the **Launch Test Console** icon to invoke the Test Console, which you use to validate and test the design of your services and transformations. For MFLs, you can use the Test Console whether you are inside or outside a session. See Section 33.2, "Testing Transformations."
Options	Click the **Delete** icon to delete a specific MFL. See Section 16.4, "Deleting MFLs."

16.2 Adding MFLs

To add an MFL:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the MFL file. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **MFL File** from under **Transformation** to display the Create a New MFL File Resource page.

4. In the **Resource Name** field, enter a unique name for this MFL. This is a required field.

    Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the MFL.

6. In the **MFL Text** field, do one of the following:

    ■   Enter text for the new MFL.

    ■   Click **Browse** to locate and import an existing MFL.

    ■   Copy and paste text from an existing MFL into this field.

    This is a required field.

7. Click **Save**. The MFL file resource is created and saved in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

16.3 Editing MFLs

To edit an MFL:

1. If you have not already done so, click Create to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the MFL file, as described in Section 16.1, "Locating MFLs."

3. Click the MFL file name. The View MFL Resource Details page displays the information shown in Table 16–2.

Table 16–2   MFL Resource Details

Property	Description
Last Modified By	The user who created this MFL file or imported it into the configuration.
Last Modified On	The date and time that the user created this MFL file or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this MFL file references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this MFL file. If such references exist, click the numeric link to view a list of the objects. For example, if you selected this MFL file as the request or response message type for a messaging business service or proxy service, the business service or proxy service is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this MFL file, if one exists.
MFL Text	The text for this MFL.

4. To make a change to the fields, click **Edit**. For descriptions of the fields, see Section 16.2, "Adding MFLs."

   You cannot change the **Resource Name** field.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

16.4 Deleting MFLs

To delete an MFL:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **MFLs** to display the Summary of MFL Files page.

3. Click the **Delete** icon in the Options field of the MFL you want to delete. The MFL file is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

17

Service Accounts

This chapter describes Service Accounts and provides steps on how to create, locate, edit, and delete Service Account resources using the Oracle Service Bus Administration Console.

A service account provides a user name and password that proxy services and business services use for outbound authentication or authentication to a local or remote resource, such as an FTP server or a JMS server. For example, if a business service is required to supply a user name and password for transport-level authentication with a Web Service, you create a service account that specifies the user name and password, then you configure the business service to include the service-account credentials in its outbound requests.

The user names and passwords that you enter in service accounts are used for outbound authentication or for providing credentials to local or remote resources. The user names and passwords that you enter in the Security Configuration module of the Oracle Service Bus Administration Console are used for inbound authentication and for authenticating administrative requests. See Section 17.1, "Specifying Service Accounts."

17.1 Specifying Service Accounts

You can use the same service account for multiple business services and proxy services. To specify the user name and password that a service account provides, you can use any of the following techniques:

- Static

  Requires you to save a user name and password with the service account configuration. The service account encodes this user name and password in the outbound request.

- User name and password pass-through

  Causes the service account to provide the user name and password that it receives from an incoming client request. For example, if an inbound HTTP BASIC request contains "pat" and "patspassword" as the user name and password, the service account encodes "pat" and "patspassword" in the outbound request.

  Because this technique requires that client requests include clear-text user names and passwords, it is applicable only for client requests that use either the HTTP BASIC protocol, a Web Services Security Username Token authentication with a clear-text password, or a custom user name and password token.

  Oracle recommends that you use this technique only when Oracle Service Bus and the endpoint belong to the same authentication domain. For example, use this

technique when you are routing messages within a single organization and both Oracle Service Bus and the message consumer authenticate against a common LDAP server.

The following restrictions apply to this technique:

– It cannot be used in outbound requests that authenticate Oracle Service Bus to a local or remote server or system resource, such as an FTP server or a JMS server.

– It cannot be used with the `fn-bea:lookupBasicCredentials` XQuery function. For more information, see "XQuery Implementation" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus.*

> **Note:** If your proxy is an active WSS intermediary, you can use WS-Security to encrypt a WS-Security Username Token or custom user name/password. In this instance, user name/password pass-through works because the proxy will first decrypt the request and will then have access to the clear-text user name/password.

■ User mapping

Requires you to correlate (map) the user name that is the result of authenticating an inbound request from a client (the local user name) to a user name and password that you specify (the remote user name and password). When the service account receives a request from an authenticated client that has been mapped, it provides the appropriate remote user name and password for the business service or proxy service outbound request.

If the client authenticates at both transport level and message level, the service account maps the message level user name to the remote user name and password.

You can also map an anonymous user name to a remote user name and password.

The following restrictions apply to this technique:

– It cannot be used in outbound requests that authenticate Oracle Service Bus to a local or remote server or system resource, such as an FTP server or a JMS server.

– It cannot be used with the `fn-bea:lookupBasicCredentials` XQuery function. For more information, see "XQuery Implementation" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Related Topics

■ Section 19.1, "Creating and Configuring Business Services"

■ Section 20.2.5, "Email Transport Configuration Page"

17.1.1 Using Service Accounts Data and Sessions

Service accounts and their data participate fully in Oracle Service Bus sessions: you must be in a session to create or modify a service account, and if you discard the session, the service account and its data is also discarded. When you activate a session, Oracle Service Bus saves the user name, password, and other service account data in the user name/password credential mapping provider that is configured for the domain.

17.2 Locating Service Accounts

To locate Service Accounts:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the service account.

   - Select **Resource Browser** > **Service Accounts**. The Summary of Service Accounts displays the information shown in Table 17–1.

2. To search for a service account, enter part or all of the account name in the **Name** field. You can also enter part or all of the account project name and folder in the **Path** fields. Click **Search**.

   Click **View All** to remove the search filters and display all service accounts.

Table 17–1   Service Account Information

Property	Description
Service Account Name	A unique name for the service account. Click the name to see the View Service Account Details page. See Section 17.4, "Editing Service Accounts."
Path	The project name and the name of the folder in which the service account resides. Click the name to see the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."
Options	Contains a **Delete** icon. If a business service or proxy service has been configured to use a service account, a Deletion Warning icon indicates that you can delete the service account with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted service account. See Section 17.5, "Deleting Service Accounts."

17.3 Adding Service Accounts

To add a Service Account:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the service account. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Service Account** to display the Create a New Service Account page.

4. In the **Resource Name** field, enter a unique name for this service account.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Resource Description** field, enter a description for the service account.

6. Under **Resource Type**, do one of the following:

   - To create a service account that provides the user names and passwords that it receives from incoming client requests, select **Pass Through**.

   - To create a service account that provides a user name and password that you save with the service account configuration, select **Static**.

- To create a service account that maps the user name from one or more authenticated clients to user names and passwords that you specify, select **Mapping**.

7. Depending on the resource type you selected in the previous step, do one of the following steps described in Table 17–2.

Table 17–2   Resource Type Options

Selected Resource Type	Complete These Steps
Pass Through	Click **Last**.
Static	1. Click **Next**.
	2. Enter the user name and password in the **User Name** field, **Password**, and **Confirm Password** fields.
	3. Click **Last**.
Mapping	To create a service account that maps the user name from one or more clients to user names and passwords that you specify, do the following:
	1. Click **Next**.
	2. In the Enter Authorized Remote User table, do the following:
	In the **Remote User Name**, **Password**, and **Confirm Password** fields, enter the user name and password that you want to send in outbound requests.
	Click **Add**. The user mapping is added to the Remote Users table.
	(Optional) Add additional remote users in the Enter Authorized Remote User table.
	3. Click **Next**.
	4. To map authorized clients to remote user names and passwords, do the following in the Enter Authorized Local User table:
	In the **Local User Name** field, enter the name that identifies a client that has been authenticated on its inbound request.
	If you have not already added this user in the Security Configuration module of the Oracle Service Bus Administration Console, do so before you use this mapping in a runtime environment. See Section 25.3, "Adding Users." Oracle Service Bus lets you create a mapping for a non-existent local user, but the mapping will never match an authenticated user and will never be used.
	From the **Remote User Name** list, select the user name that you want to send in outbound requests for the authenticated user you specified in the Local User Name field.
	Click **Add**.
	5. Click **Last**.

8. Click **Save**. The service account is created and saved in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

17.4 Editing Service Accounts

To edit a Service Account:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the service account, as described in Section 17.2, "Locating Service Accounts."

3. Click the service account name. The View Service Account Details page displays the information shown in Table 17–3.

Table 17–3   Service Account Details

Property	Description
Last Modified By	The user who created this service account or imported it into the configuration.
Last Modified On	The date and time that the user created this service account or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this service account references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this service account. If such references exist, click the numeric link to view a list of the objects. For example, if you selected this service account as the JMS service account in a proxy service with a JMS transport protocol, the proxy service is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this service account, if one exists.

4. To make a change to the fields, click **Edit**. See Section 17.3, "Adding Service Accounts" for descriptions of the fields.

    You cannot change the **Resource Name** field.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

> **Note:** If the service account that you modified is used to authenticate with a WebLogic JMS server, the JMS server might not recognize your modification for up to 60 seconds. By default, WebLogic Server JMS checks permissions for each destination every 60 seconds. To change this behavior, modify the WebLogic Server startup command so that it sets the following system property to the frequency (in seconds) that you want WebLogic Server JMS to check permissions: `weblogic.jms.securityCheckInterval`.
>
> A value of `0` (zero) for this property ensures that a permissions check is performed for every `send`, `receive`, and `getEnumeration` action on a JMS resource.

Related Topics

See "Ensuring the Security of Your Production Environment" in *Securing a Production Environment*.

17.5 Deleting Service Accounts

When you delete a service account, the user name, password, or local-user to remote-user mapping data that the service account contains is also deleted.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. If any business service or proxy service is configured to use the service account, remove the service account from the business service or proxy service.

   See Section 19.5, "Editing Business Service Configurations" or Section 20.5, "Editing Proxy Service Configurations."

3. Select **Resource Browser** > **Service Accounts** to display the Summary of Service Accounts page.

4. Click the **Delete** icon in the **Options** field of the service account you want to delete.

   The service account is deleted in the current session. If a business service or proxy service has been configured to use a service account, a Deletion Warning icon indicates that you can delete the service account with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted service account.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

18

Service Key Providers

This chapter describes Service Key Providers and provides steps on how to create, locate, edit, and delete Service Key Provider resources using the Oracle Service Bus Administration Console.

A service key provider contains Public Key Infrastructure (PKI) credentials that proxy services use for decrypting inbound SOAP messages and for outbound authentication and digital signatures. A PKI credential is a private key paired with a certificate that can be used for digital signatures and encryption (for Web Service Security) and for outbound SSL authentication. The certificate contains the public key that corresponds to the private key.

> **Note:** To use a service key provider, you must configure a PKI credential mapping provider. See "Configuring the WebLogic Security Framework: Main Steps" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

A single service key provider can contain all of the following PKI credentials:

- A key-pair for digital encryption

  Proxy services use this key-pair to decrypt inbound SOAP messages that have been encrypted to conform with a Web Services Policy statement. If you want the service key provider to support digital encryption, the key store that is associated with the PKI credential mapper must contain at least one X.509 certificate that supports encryption.

- A key-pair for digital signatures

  Proxy services use this key-pair when its endpoint is a Web Service and the Web Service requires clients to sign one or more parts of a SOAP envelope.

- A key-pair for SSL client authentication (two-way SSL)

  Proxy services use this key-pair to authenticate when acting as a client during an outbound TLS/SSL (Secure Sockets Layer) connection; that is, when routing a message to an HTTPS business service or proxy service that requires client-certificate authentication.

You can use the same service key provider for multiple proxy services.

18.1 Locating Service Key Providers

To locate Service Key Providers:

1. Do either of the following:

- Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the service key provider.

- Select **Resource Browser** > **Service Key Providers**. The Summary of Service Key Providers displays the information shown in Table 18–1.

2. To search for a service key provider, enter part or all of the provider name in the **Name** field. You can also enter part or all of the provider project name and folder in the **Path** fields. Click **Search**.

   Click **View All** to remove the search filters and display all service key providers.

Table 18–1   Service Key Provider Information

Property	Description
Name	A unique name for the service key provider. Click the name to see the View Service Key Provider Details page. See Section 18.3, "Editing Service Key Providers."
Path	The project name and the name of the folder in which the service key provider resides. Click the name to see the project or folder that contains this resource. See Section 4.1.1, "Qualifying Resource Names Using Projects and Folders."

18.2 Adding Service Key Providers

To add a Service Key Provider:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select a project or folder in which to add the service key provider. The Project/Folder View page is displayed.

3. From the **Create Resource** list, select **Service Key Provider** to display the Create a New Service Key Provider page.

4. In the **Service Key Provider Name** field, enter a unique name for this service key provider.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Description** field, enter a description for the service key provider.

6. Do any of the following steps, shown in Table 18–2.

Table 18–2 Authentication Options

To Add a Key-Pair for...	Complete These Steps...
Digital encryption	**1.** Next to **Encryption Key**, click **Browse**.
	The Select an alias for Encryption Key window displays the key aliases from the key store that your realm's PKI credential mapper is using.
	2. In the Select an alias for Encryption Key window, enter the password that you use to secure access to the key store. (You set the password when you create the keystore.)
	3. Select a key alias that maps to an X.509 certificate and that supports encryption.
	4. Click **Submit**.
	When you associate this service key provider with a proxy service, Oracle Service Bus embeds the X.509 certificate into the proxy service's WSDL. The proxy service then uses this certificate to encrypt the messages that it sends to its endpoint. The proxy service uses the private key in the PKI credential to decrypt the messages that the endpoint returns.
Digital signatures	**1.** Next to **Digital Signature Key**, click **Browse**.
	The Select an alias for Digital Signature Key window displays the key aliases from the key store that your realm's PKI credential mapper is using.
	2. In the Select an alias for Digital Signature Key window, enter the password that you use to secure access to the key store. (You set the password when you create the keystore.)
	3. Select a key alias.
	4. Click **Submit**.
SSL client authentication (two-way SSL)	**1.** Next to **SSL Client Authentication Key**, click **Browse**.
	The Select an alias for SSL Client Authentication Key window displays the key aliases from the key store that your realm's PKI credential mapper is using.
	2. In the Select an alias for SSL Client Authentication Key window, enter the password that you use to secure access to the key store. (You set the password when you create the keystore.)
	3. Select a key alias.
	4. Click **Submit**.

7. Click **Save**. The service key provider is saved in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

18.3 Editing Service Key Providers

Use the View Service Key Provider Details page to view and change details of a specific service key provider.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the service key provider, as described in Section 18.1, "Locating Service Key Providers."

3. Click the service key provider name. The View Service Key Provider Details page displays the information shown in Table 18–3.

Table 18–3   Service Key Provider Details

Property	Description
Service Key Provider Name	The name of this service key provider.
Last Modified By	The user who created this service key provider or imported it into the configuration.
Last Modified On	The date and time that the user created this service key provider or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this service key provider references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this service key provider. If such references exist, click the numeric link to view a list of the objects. For example, if you selected this service key provider as the service provider for a specific proxy service, the proxy service is listed as a reference when you click the link. See Section 4.22, "Viewing References to Resources."
Description	A description of this service key provider, if one exists.

4. To make a change to the fields, click **Edit**. See Section 18.2, "Adding Service Key Providers" for descriptions of the fields.

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

18.4 Deleting Service Key Providers

When you delete a service key provider, Oracle Service Bus also deletes the associated alias to key-pair bindings from PKI credential mapping provider. Oracle Service Bus does not delete the associated key-certificate pair from the key store.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. If any proxy service is configured to use the service key provider, remove the service key provider from the proxy service. See Section 20.5, "Editing Proxy Service Configurations."

3. Select **Resource Browser** > **Service Key Providers** to display the Summary of Service Key Providers page.

4. Click the **Delete** icon in the Options field of the service key provider you want to delete. The service key provider is deleted in the current session. If a business service or proxy service has been configured to use a service account, a Deletion Warning icon indicates that you can delete the service key provider with a warning confirmation. This might result in conflicts due to unresolved references from the service to the deleted service key provider.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

19

Business Services: Creating and Managing

This chapter describes how to create, configure, and manage business services using the Oracle Service Bus Administration Console.

Business services are Oracle Service Bus definitions of the enterprise services with which you want to exchange messages. You define business services using WSDLs (Web Services Definition Language) just as you would define a proxy service. However, the configuration of business services differs from that of proxy services in that a business service does not have a pipeline. Therefore, a business service is any service not implemented by the Oracle Service Bus pipelines.

For more information, see "Configuring Proxy Services and Business Services" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

19.1 Creating and Configuring Business Services

This section describes how to create and configure business services.

For JCA services, you can generate business services from JCA Bindings, as described in Section 19.1.1, "Generating a Business Service from a JCA Binding Resource", or from Oracle Enterprise Repository, as described in Section 19.1.2, "Generating a Business Service from Oracle Enterprise Repository."

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder to which you want to add the business service.

3. On the Project/Folder View page, select **Business Service** from the **Create Resource** list.

4. On the Create/Edit a Business Service- General Configuration page, provide a name for the service and select the type of service to create.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. Click **Next**. The pages that follow depend on the choices you made on the first page. Enter the appropriate information on each of the subsequent pages, until you reach the summary page, then click **Save** to save the service in the current session.

   For detailed instructions on completing each page, see the following:

   - Section 19.2.1, "General Configuration Page"

   - Section 19.2.2, "Message Type Configuration Page"

   - Section 19.2.3, "Transport Configuration Page"

- Section 19.2.4, "Protocol-Specific Transport Configuration Page"

- Section 19.2.22, "SOAP Binding Configuration Page"

- Section 19.2.23, "Message Handling Configuration Page"

- Section 19.2.24, "Summary Page"

- Section 19.5.2, "Business Service Policies Page"

- Section 19.5.3, "Business Service Security Page"

If a business service is created from a WSDL that includes WS-Policy attachments, the policies (read-only) are displayed on the Protocol-Specific Transport Configuration page. If any of the service's WS-Policies specifies authentication, then you must select a service account. A proxy service that routes to this business service will use this service account to authenticate to the business service.

19.1.1 Generating a Business Service from a JCA Binding Resource

You can generate a JCA business service from an outbound JCA Binding resource in Oracle Service Bus. For more information on JCA Binding resources, see Chapter 13, "JCA Bindings."

To generate a JCA business service from a JCA Binding:

1. In the Oracle Service Bus Administration Console, click **Create** or **Edit** in the Change Center if you area not already in Create or Edit mode.

2. In the **Resource Browser**, click **JCA Bindings**.

3. Locate the JCA Binding from which you want to generate a service, and click the **Action** icon.

4. In the window that appears, confirm the name of the WSDL and the service you want to generate, select the location for these new resources, and click **Generate**.

    Oracle Service Bus generates the service and its corresponding WSDL.

5. Modify any other configuration details on the generated service as appropriate, such as the Endpoint URI.

19.1.2 Generating a Business Service from Oracle Enterprise Repository

You can generate a business service from Oracle Enterprise Repository using the Oracle Service Bus development environment. For more information, see "Generating a Business Service from Oracle Enterprise Repository" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

19.2 Create/Edit a Business Service - Page Reference

Create a business service by selecting **Business Service** from the **Create Resource** list on the Project/Folder View page. When you select that option, it displays the first in a series of pages for configuring and adding business services. The pages displayed vary, depending on the options you choose along the way. The pages are:

- Section 19.2.1, "General Configuration Page"

- Section 19.2.2, "Message Type Configuration Page"

- Section 19.2.3, "Transport Configuration Page"

- Section 19.2.4, "Protocol-Specific Transport Configuration Page"

- Section 19.2.22, "SOAP Binding Configuration Page"

- Section 19.2.23, "Message Handling Configuration Page"

- Section 19.2.24, "Summary Page"

- Section 19.5.2, "Business Service Policies Page"

- Section 19.5.3, "Business Service Security Page"

19.2.1 General Configuration Page

The Create/Edit a Business Service - General Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to modify general configuration settings for a business service.

When you create a business service, this is the first page displayed in a series of pages for configuring the service. The pages displayed after this one differ depending on the choices you make on this page.

Table 19–1 describes how to use the page.

Table 19–1   Create/Edit a Business Service - General Configuration Page

Option	To create or edit...
Service Name	Enter a unique name for the business service.
	Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.
Description	Enter a description.

Table 19–1 (Cont.) Create/Edit a Business Service - General Configuration Page

Option	To create or edit...
Service Type	A service type defines the types and packaging of the messages exchanged by the service. Select the type of business service to create:

- **WSDL Web Service** - Select this option to create a business service based on a WSDL. Then, enter the WSDL name, qualified by its path (for example, *myProject/myFolder*/myWSDL). Alternatively, click **Browse** to select a WSDL from the Select a WSDL page.

  (port or binding) - Enter the name of a port (defined in the WSDL) to describe an actual transport address, or enter the name of a binding (defined in the WSDL) to map to a transport address. If you use **Browse** to select a WSDL, as described above, the Select a WSDL Definition page lists any ports and bindings defined in the WSDL. When you choose a port or a binding on that page, the **(port or binding)** field is filled with the selected name.

- **Transport Typed Service** - Select this option to create a service that uses the EJB or Flow transport.

- **Messaging Service** - Select this option to create a service that exchanges messages of very different content-type. These exchanges can be either request/response or one-way. It can also just have a response with no request when used with the HTTP 'GET' option for the HTTP transport. Unlike Web Services, the content-type of the request and response need not be the same.

- **Any SOAP Service** - Select this option to create a SOAP service that does not have an explicitly defined, concrete interface.

  Keep the default **SOAP 1.1**, or select **SOAP 1.2** from the list.

- **Any XML Service** - Select this option to create an XML service that does not have an explicitly defined, concrete interface.

  HTTP GET is only supported for messaging services and this service type.

- **Business Service** - Select this option to clone an existing business service.

  Enter the path (project/folder) and the name of the business service; or click **Browse** to select the business service from the Summary of Business Services page.

  Since Oracle Service Bus does not accept the same URI for multiple services, you must change the URI for the cloned service.

- **Proxy Service** - Select this option to create a business service based on an existing proxy service.

Note: When a service is created from another service, alert rules are maintained in the following way:

- When a proxy service is created from a business service or a business service is created from a proxy service, the alert rules, if any, are removed.

- When a proxy service is created from another proxy service or a business service is created from another business service, the alert rules, if any, are retained.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.2 Message Type Configuration Page

The Create/Edit a Business Service - Message Type Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure message types for a business service whose type is Messaging Service.

The binding definition for messaging services consists of configuring the content-type of the messages that are exchanged. The content-type for the response does not need to be the same as for the request; therefore, the response is configured separately (for example, the service could accept an MFL message and return an XML acknowledgment receipt).

> **Note:**   Email, File, FTP, or SFTP transport business services whose type is Messaging Service support one-way messaging *only*; the **Response Message Type** should be none. If you select an option other than none, the file, ftp, or sftp protocol will not be available on the Transport Configuration page.

Table 19–2 describes how to use Create/Edit a Business Service - Message Type Configuration page.

Table 19–2   Create/Edit a Business Service - Message Type Configuration Page

Option	To create or edit...
Request Message Type	Select a message type for the request message:
	■ **None** - Select this option if there is no request message (HTTP GET example)
	■ **Binary** - Select this option if the content-type of the message is unknown or not important.
	■ **Text** - Select this option if the message can be restricted to text.
	■ **MFL** - Select this option if the message is a binary document conforming to an MFL definition. You can configure only one MFL file. For MFLs, you can click **Browse** to select an MFL from the MFL Browser, then click **Submit**.
	■ **XML** - Select this option if the message is an XML document. To provide some type information, you can choose to declare the XML schema type of the XML document exchanged.
	■ **Java** - Select this option if a Java Object is being sent in the request. The JMS transport is used for Java Object messages. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–2   (Cont.) Create/Edit a Business Service - Message Type Configuration Page

Option	To create or edit...
Response Message Type	Select a message type for the response message:
	■ **None** - Select this option if there is no response message.
	■ **Binary** - Select this option if the content-type of the message is unknown or not important.
	■ **Text** - Select this option if the message can be restricted to text.
	■ **MFL** - Select this option if the message is a binary document conforming to an MFL definition. You can configure only one MFL file.
	For MFLs, you can click **Browse** to select an MFL from the MFL Browser, then click **Submit**.
	■ **XML** - Select this option if the message is an XML document. To provide some type information, you can choose to declare the XML schema type of the XML document exchanged.
	■ **Java** - Select this option if a Java Object is being received in the response. The JMS transport is used for Java Object messages. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.3 Transport Configuration Page

The Create/Edit a Business Service - Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to select a transport protocol for the business service and to set other general transport configuration settings.

Outbound transport-level security applies to the connections between Oracle Service Bus proxy services and business services. For more information about transport-level security, see "Configuring Transport-Level Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–3 describes how to use this page.

Table 19–3   Create/Edit a Business Service - Transport Configuration Page

Option	To create or edit...
Protocol	Select a transport protocol from the list. The protocols available differ, depending on the service type you are creating: ■ **WSDL Web Service**: soa-direct, bpel-10g, dsp, http, jca, jms, sb, ws ■ **Transport-Typed Service**: ejb, flow, jejb ■ **Messaging Service**: email, file, ftp, http, jms, mq (if available), sftp, tuxedo ■ **Any SOAP Service**: dsp, http, jms, sb ■ **Any XML Service**: dsp, email, file, ftp, http, jms, mq (if available), sb, sftp, tuxedo
Load Balancing Algorithm	Select one of these load-balancing algorithms: ■ **Round-robin** - This algorithm dynamically orders the URLs that you enter in the **Endpoint URI** field for this business service. If the first one fails, it tries the next one, and so on until the retry count is exhausted. For every new message, there is a new order of URLs. ■ **Random** - This algorithm randomly orders the list of URLs that you enter in the **Endpoint URI** field for this business service. If the first one fails, it tries the next one, and so on until the retry count is exhausted. ■ **Random-weighted** - This algorithm randomly orders the list of URLs that you enter in the **Endpoint URI** field for this business service, but some are retried more than others based on the value you enter in the **Weight** field. ■ **None** - This algorithm orders the list of URLs that you enter in the **Endpoint URI** field for this business service from top to bottom.

Table 19–3   (Cont.) Create/Edit a Business Service - Transport Configuration Page

Option	To create or edit...

Endpoint URI   Enter an endpoint URL in the format based on the transport protocol you selected in the **Protocol** field. Following are descriptions of the URI formats for each transport. Optional URI elements are shown in brackets []. For more information on transport URIs, see the respective transport chapters in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

- **bpel-10g** – `protocol://host[:port]`
  `[/protocol-path]/domain/process[/version[/partnerlink/role]]`

- **dsp** - `t3://dsp-ip-address:port/dsp-app-name`

- **ejb** - `ejb:jndi_provider_name:ejb_jndi_name`

  In the URI, `jndi_provider_name` is the name of the Oracle Service Bus JNDI provider resource, and `ejb_jndi_name` is the JNDI name in the JNDI server for the EJB.

  If the JNDI provider is located on the same server, the JNDI provider need not be specified. The URI then would be `ejb::ejb_jndi_name`

  Note: If your EJBs are running on IBM WebSphere, `ejb_jndi_name` must be in the following format:

  `cell/nodes/node_name/servers/server_name/ejb_jndi_name`

  or

  `cell/clusters/cluster_name/ejb_jndi_name`

  as described in the IBM WebSphere documentation at
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.w ebsphere.nd.iseries.doc/info/iseriesnd/ae/rnam_example_prop3.html

- **email** - You can enter multiple endpoint URIs in any combination of the following formats:

  `mailto:`*name@domain_name*`.com`

  `mailto:`*name@domain_name*`.com?smtp=`*smtp_server_resource*

  `mailto:`*name@domain_name*`.com?mailsession=`*jndi_mail_session*

- **file** - `file:///drivename:/somename`

- **flow** - `flow:<reference path to split-join resource>`

  `reference path to split-join resource` is the path (project/folder) and name of the Split-Join; for example, `batchorderProcessing/batchorder`.

- **ftp** - `ftp://host:port/directory`

- **http** - `http://host:port/someService`

  The HTTP transport supports both `HTTP` and `HTTPS` endpoints.

- **jca** - `jca://<resource_adapter_jndi>`

- **jejb** - `jejb:jndi_provider_name:ejb_jndi_name`

  In the URI, `jndi_provider_name` is the name of the Oracle Service Bus JNDI provider resource, and `ejb_jndi_name` is the JNDI name in the JNDI server for the EJB.

  EJB 2.1 example: `jejb:myProvider:osb.jejb.myJejbBiz21`

  EJB 3.0 example: `jejb:myProvider:myBiz31#osb.jejb.myJejbBiz`

  Note: If your EJBs are running on IBM WebSphere, `ejb_jndi_name` must be in the following format:

  `cell/nodes/node_name/servers/server_name/ejb_jndi_name`

  or

  `cell/clusters/cluster_name/ejb_jndi_name`

  as described in the IBM WebSphere documentation at
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.w ebsphere.nd.iseries.doc/info/iseriesnd/ae/rnam_example_prop3.html

Table 19–3   (Cont.) Create/Edit a Business Service - Transport Configuration Page

Option	To create or edit...
Endpoint URI (continued)	■ **jms** - `jms://host:port/factoryJndiName/destJndiName` To target a JMS destination to multiple servers, use the following URI format: `jms://host1:port,host2:port/connection_factory/jndi_destination` You can also omit host and port from the URI to have the lookup performed on the local machine. For example: `jms:///connection_factory/jndi_destination` **In a cluster**: The host names in the JMS URI must exactly match the host names of the cluster servers as they are configured in Oracle WebLogic Server. **Note**: While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the endpoint URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the endpoint URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*. ■ **mq** - `mq://<local-queue-name>?conn=<mq-connection-resource-ref>` `local-queue-name` is the name of the MQ queue from which the business service reads messages. `mq-connection-resource-ref` is the path (project/folder) and name of the MQ connection resource; for example, `default/my_MQconnection`. **Note**: The Endpoint URI cannot contain spaces, so do not create MQ Connection resources or projects/folders with spaces in the names. To make the MQ transport available in Oracle Service Bus, see Section 9, "MQ Connections." ■ **sb** - `sb://[<jndi_provider_name/>]service_name` `jndi_provider_name` (optional) is the name of the Oracle Service Bus JNDI Provider resource. When omitted, the default context is used. `service_name` is a target service and corresponds to the remote proxy service URI. ■ **sftp** - `sftp://hostname:port/directory` ■ **soa-direct** – `[protocol://authority]/default/compositeName[!versionNumber[*label]]/serviceName` The protocol and authority are optional when the SOA services are co-located on the same server as Oracle Service Bus. ■ **tuxedo** - `tuxedo:resourcename[/remotename]` In the URI, `resourcename` corresponds to a WTC Import name and `remotename` corresponds to the service name exported by the remote Tuxedo domain. The URI `resourcename` is required, and the `remotename` is optional. If more than one URI is specified, you must have unique resource names for the endpoints. If no remote name is specified, its value is the value of the resource name. If no remote name is entered or if remote and resource name are the same, only one URI is allowed. In this case resource name and remote name have the same value. This allows already defined WTC Imports to make use of WTC load-balancing and failover. ■ **ws** - `http://host:port/somename` or `https://host:port/somename` Click **Add** to add one or more additional URIs. At runtime, the URLs are selected based on the load balancing algorithm you selected in the **Load Balancing Algorithm** field. If you selected **Random-weighted** in the **Load Balancing Algorithm** field, you can also enter a weight in the **Endpoint URI** field. The default is `1`. If you have multiple endpoint defined, and you selected **None** in the **Load Balancing Algorithm** field, the order of endpoints is significant. You can reorder the endpoints using the **Up** and **Down** arrows in the Options column. Oracle Service Bus does not support duplicate endpoint URIs within the same business service.

Table 19–3 (Cont.) Create/Edit a Business Service - Transport Configuration Page

Option	To create or edit...
Retry Count	In case of delivery failure when sending outbound requests, specify the number of times to retry individual URL endpoints; in other words, the number of failover attempts.
	For example, a business service has one configured URI (U1) and the number of retries is set to 3. If U1 fails on the first attempt, the system retries the U1 endpoint 3 more times.
	If a business service has 2 configured URIs (U1 and U2) and a retry count of 3, if the first attempt (for example, to U1) fails, the system tries (fails over to) the next URI (U2). If that also fails, the system makes two more attempts, once to U1 and once to U2.
Retry Iteration Interval	Specify the number of seconds the system pauses before iterating over all the endpoint URIs in the list again.
	For example, a business service has two configured URIs (U1 and U2) and a retry count of 2 with a retry iteration interval of 5 seconds. If the first attempt (to U1) fails, the system tries U2 right away. But if U2 also fails, the system waits for 5 seconds and retries U1 once more.
Retry Application Errors	Select **Yes** or **No**.
	In case of delivery failure when sending outbound requests, specify whether or not to retry endpoint URIs based on application errors (for example, a SOAP fault).
	This field is available only for these transports: HTTP, EJB, JMS, DSP, Tuxedo, SB, and WS.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.4 Protocol-Specific Transport Configuration Page

The Create/Edit a Business Service - Protocol-Specific Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

- Section 19.2.5, "BPEL-10g Transport Configuration Page"
- Section 19.2.6, "DSP Transport Configuration Page"
- Section 19.2.7, "EJB Transport Configuration Page"
- Section 19.2.8, "Email Transport Configuration Page"
- Section 19.2.9, "File Transport Configuration Page"
- Section 19.2.10, "Flow Transport Configuration Page"
- Section 19.2.11, "FTP Transport Configuration Page"
- Section 19.2.12, "HTTP Transport Configuration Page"
- Section 19.2.13, "JCA Transport Configuration Page"
- Section 19.2.14, "JEJB Transport Configuration Page"
- Section 19.2.15, "JMS Transport Configuration Page"
- Section 19.2.16, "MQ Transport Configuration Page"
- Section 19.2.17, "SB Transport Configuration Page"
- Section 19.2.18, "SFTP Transport Configuration Page"
- Section 19.2.19, "SOA-DIRECT Transport Configuration Page"
- Section 19.2.20, "Tuxedo Transport Configuration Page"

- Section 19.2.21, "WS Transport Configuration Page"

19.2.5 BPEL-10g Transport Configuration Page

Use this page to configure transport settings for a business service using the BPEL-10g (Oracle BPEL Process Manager) transport protocol. For more information on using Oracle Service Bus with Oracle BPEL Process Manager, see "Oracle BPEL Process Manager Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–4   Create/Edit a Business Service - BPEL-10g Transport Configuration Page

Property	Description
Role	The BPEL transport is used to send request messages from Oracle Service Bus to Oracle BPEL Process Manager. The business service can serve one of the following roles: ■ **Synchronous Client** – For synchronous communication with an Oracle Service Bus client, the only location information that is required is the BPEL address. This address is captured statically as the endpoint URI and dynamically through URI rewriting. ■ **Asynchronous Client** – For asynchronous communication with an Oracle Service Bus client, a callback from Oracle BPEL Process Manager to Oracle Service Bus is sent on a different connection than the request, and you must configure Oracle Service Bus to provide the correct callback address. For more information, see "Advanced Use Cases (Asynchronous)" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. ■ **Service Callback** – If the business service is designed to be a service callback to Oracle BPEL Process Manager (where Oracle BPEL Process Manager is calling an external service through Oracle Service Bus), the callback address is known only at runtime. Use an Endpoint URI of bpel://callback. If you configure the business service with the marker URI, configure your pipeline logic to dynamically set the URI on $outbound; for example, using the TransportHeader action. **Note**: A Service Callback business service does not support load balancing or failover.
Callback Proxy	This optional field is available only for the Asynchronous Client role. This field lets you select the proxy service (must be either an SB or HTTP proxy of type Any SOAP) that will be used to route callbacks to the Oracle Service Bus client that made the request. For more information, see "Advanced Use Cases (Asynchronous)" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Service Account	For JNDI context security, used to access the Oracle BPEL Process Manager delivery service. Click Browse and select a service account. If no service account is specified, an anonymous subject is used. There is no restriction on the type of service account that can be configured, such as static or pass-through, but the runtime must be able to access a user name and password.

Table 19–4   (Cont.) Create/Edit a Business Service - BPEL-10g Transport Configuration

Property	Description
Suspend Transaction	Selecting Suspend Transaction makes the business service non-transactional even if the business service is invoked by a transaction.
	If you do not select Suspend Transaction:
	■ If the protocol indicates a WebLogic Server-supported protocol (t3, iiop, http), the transaction is propagated.
	■ If the protocol indicates an OC4J server (ormi, opmn), the BPEL transport throws an exception, since OC4J does not support transaction propagation.
	The BPEL transport ignores the Suspend Transaction option in the following situations:
	■ The business service is called with quality of service (QoS) "best-effort." The BPEL transport automatically suspends any transaction that does not support QoS.
	■ The business service is called with QoS set to "exactly-once" and there is no transaction.
	For a description of transaction propagation, see "Transaction Propagation" in the at *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

19.2.6 DSP Transport Configuration Page

The Create/Edit a Business Service - DSP Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "DSP and Oracle Data Service Integrator Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the DSP (Oracle Data Service Integrator) transport protocol. Table 19–5 describes how to use this page.

Table 19–5   Create/Edit a Business Service - DSP Transport Configuration Page

Option	To create or edit...
Debug Level	Specify one of the following
	■ `0` - for no debug information
	■ `1` - to output information on the request message
	■ `3` - to output information on the request and the response message

Table 19–5 (Cont.) Create/Edit a Business Service - DSP Transport Configuration Page

Option	To create or edit...
Service Account	Click **Browse** and select a service account from the list displayed. If no service account is specified, an anonymous subject is used.
	For more information:
	■ Section 17, "Service Accounts"
	■ Section 17.3, "Adding Service Accounts"
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	The Work Manager is used to post the reply message for response processing.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.7 EJB Transport Configuration Page

The Create/Edit a Business Service - EJB Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure transport settings for a business service using the EJB transport protocol. Table 19–6 describes how to use the page.

Table 19–6 Create/Edit a Business Service - EJB Transport Configuration Page

Option	To create or edit...
Pass Caller's Subject	Select this check box to have Oracle Service Bus pass the authenticated subject from the proxy service when invoking the EJB and no service accounts are configured. The **Service Account** field is disabled when this option is selected.
Service Account	Click **Browse** and select a service account from the list displayed. If no service account is specified, an anonymous subject is used. This option is not available if you use the Pass Caller's Subject option.
	For more information:
	■ Section 17, "Service Accounts"
	■ Section 17.3, "Adding Service Accounts"
Supports Transaction	Select this check box to specify transaction support.
Client Jar	Click **Browse** and select an EJB client JAR resource from the list displayed. To learn about creating JAR resources, see Section 8, "JARs." This is a required field.

Table 19–6 (Cont.) Create/Edit a Business Service - EJB Transport Configuration Page

Option	To create or edit...
Converter Jar	Click **Browse** and select an EJB converter class JAR resource from the list displayed. To learn more about EJB client JAR resources and converter classes, see "EJB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Home Interface	**EJB 2.1 only** – Select the required EJBHome interface from the options populated by the JAR. The JNDI name in this URI sample must be associated with the EJBHome interface you select here. If the EJB is not of the required type or an EJBHome interface is not specified in the client-jar, Oracle Service Bus displays a warning.
Remote Interface	**EJB 2.1 only** – This field is automatically populated depending on the configuration of the Home Interface.
Business Interface	**EJB 3.0 only** – Select the business interface in the client JAR you want to use.
Target Namespace	This field is populated by information picked up from the JAR.
Style	Select **Document Wrapped** or **RPC** according to your requirements. If two or more methods of your stateless session EJB have the same number and data type of parameters, and you want the operations to be document-oriented, you must specify that they be document-wrapped. The style is important because when routing or publishing to the EJB, $body must have content that matches the style. Also when calling out to an EJB, the style affects the parameter contents, especially for document wrapped. Secondly one usage pattern is to define an EJB business service and then create a proxy service with the same WSDL that routes to the EJB. In this case care must be taken on the style of the WSDL because the client tool used to invoke the proxy might have limitations on the style of the WSDL.
Encoding	Select **Encoded** or **Literal**.
Methods	The methods displayed are those of the EJB remote or business interface you selected. Select the required methods (you can select multiple methods). Click + to expand the method. Edit the default parameter values and select a converter if provided (or required). You must exclude the methods with parameters or return types that are not supported by the JAX-RPC stack, or you must associate such arguments with converter classes. You can change the default operation name for a given method. (By default, the operation name is the method name.) If an EJB contains methods with same name, you must change the operation names so that they are unique—WSDLs require unique operation names. **Note**: If the credentials or transaction settings are different between the methods for a given EJB, you can customize the methods for a given business service and create a business service per method. This gives you fine-grained control over transactions and credentials.

Table 19–6   (Cont.) Create/Edit a Business Service - EJB Transport Configuration Page

Option	To create or edit...
Exceptions	This field appears if a method throws a business exception. If an EJB method throws an exception that has data types not supported by Java Web Services (JWS), such as an ArrayList, use the Exceptions field to select a converter class that converts the exception to a type supported by JWS.
	Your converter class must implement com.bea.wli.sb.transports.ejb.ITypeConverter. Converter classes can only be configured for checked exceptions and not for runtime exceptions.
	Package the converter class and the converted exception class in the client or converter JAR.
	For more information, see "EJB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.8 Email Transport Configuration Page

The Create/Edit a Business Service - Email Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure transport settings for a business service using the email transport protocol. Table 19–7 describes how to use the page.

Table 19–7   Create/Edit a Business Service - Email Transport Configuration Page

Option	To create or edit...
SSL Required	Select this option to communicate using Secure Sockets Layer. The Email Transport supports one-way SSL communication. The default SMPT port number when using SSL is 465.
	Notes:
	■ The email server certificate must be present in the Oracle Service Bus truststore in order for SSL to work.
	■ If you are using TLS, you must either place the trust certificate in the default trust store (`$JAVA_HOME/jre/lib/security/cacerts`) or place it in any trust store and add the path in the `$DOMAIN_HOME/bin/setDomainEnv.*` file, as shown below:
	`set EXTRA_JAVA_PROPERTIES=%EXTRA_JAVA_PROPERTIES%` `-Djavax.net.ssl.trustStore=<path to trustStore>`
SMTP Server	Select the default SMTP Server to use for endpoint URI entries of *name@domain_name*.com. If you provide SMTP server parameters in the endpoint URI, as described in Table 19–3, those server resources are used instead of this SMTP Server setting.
	Do not select an SMTP server if you use the Mail Session option.
	For more information on creating SMTP resources, see Section 31.7, "Adding SMTP Servers"

Table 19–7   (Cont.) Create/Edit a Business Service - Email Transport Configuration Page

Option	To create or edit...
Mail Session	Enter the JNDI name of the configured mail session to use for endpoint URI entries of *name@domain_name*.com. If you provide JNDI mail session parameters in the endpoint URI, as described in Table 19–3, those mail sessions are used instead of this Mail Session setting.
	Do not enter a Mail Session if you use the SMTP Server option.
From Name	Enter a display name for the originating email account for this service.
From Address	Enter the originating email account for this service.
Reply To Name	Enter a display name for the reply to email account.
Reply To Address	Enter an email address to reply to.
Connection Timeout	Enter the timeout interval, in milliseconds, before the connection is dropped. If you enter `0`, there is no timeout.
Socket I/O Timeout	Enter the socket I/O timeout interval in milliseconds. If the enter zero (0), there is no timeout.
Request Encoding	Accept the default `ISO-8859-1` as the character set encoding for requests in email transports, or enter a different character set encoding.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.9 File Transport Configuration Page

The Create/Edit a Business Service - File Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure transport settings for a business service using the file transport protocol. Table 19–8 describes how to use this page.

Table 19–8   Create/Edit a Business Service - File Transport Configuration Page

Option	To create or edit...
Prefix	Enter a prefix to be prepended to the file name.
	Do not enter * in this field. This character causes a runtime exception.
Suffix	Enter a suffix to be appended to the file name. This is a required field.
	Do not enter * in this field. This character causes a runtime exception.
Request Encoding	Accept the default `utf-8` as the character set encoding for requests in File transports, or enter a different character set encoding.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.10 Flow Transport Configuration Page

The Create/Edit a Business Service - Flow Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1,

"Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

You use Flow transport-typed business services to access Split-Joins. To learn more about Split-Joins, see "Improving Service Performance with Split-Join" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the Flow transport protocol. Table 19–9 describes how to use this page.

Table 19–9   Create/Edit a Business Service - Flow Transport Configuration Page

Option	To create or edit...
Timeout	Enter the number of seconds to wait for a response. This value is ignored for a request-only (one-way) operation.
	Note: This property is deprecated and will not be available in Oracle Service Bus 12*c*. To avoid additional upgrade steps when moving to 12*c*, use the Wait action on a Split-Join as an alternative to this property.

> **Note:**   For Flow transport business services in Oracle Service Bus 12*c*, the application error retry, iteration, and count properties will also no longer be available for invoking a split-join, and there will be a direct way to invoke a split-join.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.11 FTP Transport Configuration Page

The Create/Edit a Business Service - FTP Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure transport settings for a business service using the ftp transport protocol. Table 19–10 describes how to use this page.

Table 19–10   Create/Edit a Business Service - FTP Transport Configuration Page

Option	To create or edit...
User Authentication	Select **anonymous** if the user of the FTP server is anonymous, or select **external user** if the user of the FTP server is an externally configured account.
Identity (Email id)	This field is available only if the **User Authentication** option is set to **anonymous**.
	Enter the mail ID for the anonymous user.
Service Account	This field is available only if the **User Authentication** option is set to **external user**.
	Enter the service account for the external user.
	For more information:
	■ Section 17, "Service Accounts"
	■ Section 17.3, "Adding Service Accounts".

Table 19–10 (Cont.) Create/Edit a Business Service - FTP Transport Configuration Page

Option	To create or edit...
Timeout	Enter the socket timeout interval, in seconds, before the connection is dropped. The default is 60 seconds.
Prefix for destination File Name	Enter an optional prefix that the transport prepends to the file name on the remote server. Do not enter * in this field. This character causes a runtime exception.
Suffix for the destination File Name	Enter an optional suffix that the transport appends to the file name on the remote server. Do not enter * in this field. This character causes a runtime exception.
Transfer Mode	Select **ASCII** or **binary** as the transfer mode.
Request Encoding	Accept the default `UTF-8` as the character set encoding for requests in ftp transports, or enter a different character set encoding.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.12 HTTP Transport Configuration Page

The HTTP transport supports both HTTP and HTTPS endpoints. The Create/Edit a Business Service - HTTP Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure transport settings for a business service using the HTTP transport protocol. Table 19–11 describes how to use the page.

Table 19–11 Create/Edit a Business Service - HTTP Transport Configuration Page

Option	To create or edit...
Read Timeout	Enter the read timeout interval in seconds. A zero (0) value indicates no timeout.
Connection Timeout	Enter the connection timeout interval in seconds. If the timeout expires before the connection can be established, Oracle Service Bus raises a connection error. A zero (0) value indicates no timeout.

Table 19–11   (Cont.) Create/Edit a Business Service - HTTP Transport Configuration

Option	To create or edit...
HTTP Request Method	This parameter lets you to use one of the following HTTP methods in a request: ■ **POST** – Passes all its data, of unlimited length, directly over the socket connection as part of its HTTP request body. The exchange is invisible to the client, and the URL does not change. For REST-based requests, POST tells the transport to perform a create/replace operation or perform an action with the request. ■ **GET** – You can include as part of the request some of its own information that better describes what to get. This information is passed as a sequence of characters appended to the request URL in a query string. You can use GET in a business service with a Service Type of "Any XML Service," or with a Service Type of "Messaging Service" when the Request Message Type is set to "None." For REST-based requests, GET retrieves a representation of a remote resource. ■ **PUT** – You can use PUT in a business service with a Service Type of "Any XML Service" or "Messaging Service." PUT tells the transport to perform a create/replace operation with a REST-based request, such as uploading a file to a known location. ■ **HEAD** – You can use HEAD in a business service with a Service Type of "Any XML Service," or with a Service Type of "Messaging Service" when the Response Message Type is set to "None." HEAD tells the transport to get header information for a remote resource rather than getting a full representation of the resource in a REST-based request. ■ **DELETE** – You can use DELETE in a business service with a Service Type of "Any XML Service" or "Messaging Service." DELETE tells the transport to perform a delete operation with a REST-based request. **Note**: If a method is already set in the $outbound/transport/request/http:http-method variable, that value takes precedence over any method you select for HTTP Request Method.
Authentication	Select one of the following: ■ **None** - Specifies that authentication is not required to access this service. **Note**: Select this option if you will specify an Oracle WSM policy for the business service. ■ **Basic** - Specifies that basic authentication is required to access this service. Basic authentication instructs WebLogic Server to authenticate the client using a user name and password against the authentication providers configured in the security realm, such as a Lightweight Directory Access Protocol (LDAP) directory service and Windows Active Directory. The client must send its user name and password on the HTTP request header. Basic authentication is strongly discouraged over HTTP because the password is sent in clear text. However, it is safe to send passwords over HTTPS because HTTPS provides an encrypted channel. **Warning**: By default, all users (authorized and anonymous) can access a business service. To limit the users who can access a business service, create a transport-level authorization policy. See Section 25.15, "Editing Transport-Level Access Policies." ■ **Client Certificate** - Specifies encrypted communication and strong client authentication (two-way SSL). To use this option, all endpoint URIs in the service definition must be HTTPS. To learn more, see "Configuring Transport-Level Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–11   (Cont.) Create/Edit a Business Service - HTTP Transport Configuration

Option	To create or edit...
Service Account	Enter a service account. A service account is an alias resource for a user name and password. This is a required field if you selected the **Basic Authentication Required** field. For more information: ■ Section 17, "Service Accounts" ■ Section 17.3, "Adding Service Accounts".
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Request Encoding	Accept the default `iso-8859-1` as the character set encoding for requests in HTTP transports, or enter a different character set encoding.
Response Encoding	Accept the default `iso-8859-1` as the character set encoding for responses in HTTP transports, or enter a different character set encoding.
Proxy Server	Enter a proxy server resource or click Browse to choose an entry from the list of configured proxy server resources.
Follow HTTP redirects	Select this check box to specify that HTTP redirects (which are requests with a response code `3xx`) should be automatically followed. A re-direct occurs when you send an outbound request to the URL of a business service, and that service returns a response code (for example, `302`) that says the URL is no longer valid and this request must be sent to another URL. If you select the **Follow HTTP Redirects** check box, Oracle Service Bus automatically re-sends the request to the new URL without any action on your part. Clear this check box if you do not want the HTTP redirects to be automatically followed.
Use Chunked Streaming Mode	Select this option to use HTTP chunked transfer encoding to send messages. **Note**: Do not use chunked streaming with if you use the Follow HTTP Redirects option. Redirection and authentication cannot be handled automatically in chunked mode.
Session Stickiness	Select this option to use session stickiness (also known as session affinity) for HTTP requests for the business service. For more information, see Section I.4.4.2, "Using WSRP with HTTP Session Stickiness."
Sticky SessionID Name	If **Session Stickiness** is enabled, enter an identifying name for the sticky session.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.13 JCA Transport Configuration Page

Use this page to configure transport settings using the JCA transport protocol. For more information on using the JCA transport, see "JCA Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus* and Section 19.1.1, "Generating a Business Service from a JCA Binding Resource."

Table 19–12   JCA Transport Configuration Page

Option	Description
JCA File	Click **Browse** to select a JCA Binding. The JCA Binding defines different aspects of the service, such as details about the adapter used, a binding to the WSDL and TopLink or EclipseLink mapping file, and the activation/interaction spec properties required by the service. Once you select a valid JCA Binding, the remaining transport configuration fields become available. For more information on JCA Bindings, see Chapter 13, "JCA Bindings."
Adapter Name	A read-only value showing the name of the adapter that the JCA service will use.
Adapter Type	A read-only value showing the adapter type.
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
JNDI Service Account	JNDI Service Account is for JNDI context security, used to access the EIS adapter managed connection factory. Click **Browse** and select a service account. If no service account is specified, an anonymous subject is used. For JCA business services, there is no restriction on the type of JNDI service account that can be configured, such as static or pass-through, but the runtime must be able to access a user name and password. JCA proxy services can use only static JNDI service accounts. For more information on JNDI service accounts, see "Security" in the "JCA Transport" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
EndPoint Properties	This field lets you assign values to endpoint properties such as retries for the type of adapter the service uses. For a list of supported JCA endpoint properties, see "Endpoint Properties" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dynamic EndPoint Properties	This option lets you pass request parameters to JCA-compliant services. For example, you can use a dynamic endpoint property to pass database query parameters to the Oracle JCA Adapter for Database. For more information on querying with parameters, see "Oracle JCA Adapter for Database" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*. Enter a name/value pair for each dynamic endpoint property you want to provide. The endpoint property key matches the query parameter name.

Table 19–12   (Cont.) JCA Transport Configuration Page

Option	Description
Always use configuration from JCA file	This option determines whether or not Activation Spec Properties (proxy services) and Interaction Spec Properties (business services) are always used from the JCA file.
	If this option is selected (default), the JCA transport interacts with the JCA framework using the activation/interaction spec properties in the JCA file.
	If this option is not selected, you can override the Activation/Interaction Spec Properties.
	For the redeployment impact of using this option, see "Endpoint Redeployment" in the "JCA Transport" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Operation Name	Displays a read-only name of the selected WSDL operation. An operation can have its own activation/interaction spec properties, shown in the Activation/Interaction Spec Properties field.
Activation/Interaction Spec Properties	**Activation Spec Properties** is the field name for proxy services; **Interaction Spec Properties** is the field name for business services.
	If this service is an inbound service invoked by an EIS application, this field displays the activation spec properties for the JCA inbound operation shown in Operation Name field.
	You can override the activation/interaction spec properties if you clear the **Always use configuration from JCA file** option.
	Note: For Oracle Adapter Suite adapters, activation/interaction spec properties are displayed as read-only. The Oracle Adapter Suite adapters store their own configurations, which you must change in the Oracle Adapter Suite management tools.
Connection properties (legacy)	For legacy JCA services that use non-managed mode connection properties (deprecated in this release), see the connection configuration options at `http://download.oracle.com/docs/cd/E13159_01/osb/docs10gr3/jcatransport/transport.html#wp1105451`.

For more information on endpoint and activation/interaction spec properties, see the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

19.2.14 JEJB Transport Configuration Page

Use this page to configure transport settings using the JEJB transport protocol. For more information on using the JEJB transport, see "JEJB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–13   JEJB Transport Configuration Page

Option	Description
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"
EJB Spec Version	Select the EJB version of the remote EJB interface.

Table 19–13   (Cont.) JEJB Transport Configuration Page

Option	Description
Pass XMLBeans by value	Select this option if you want the transport to generate an "inlined" XML representation of POJO arguments (an XMLObject) whose parameters you can access and manipulate with XQuery expressions.
	Note: Type information is not available inline for XMLObjects passed by value. If you use this option, you cannot pass the typed XMLObject as the argument in a Java Callout in a proxy service pipeline.
	Do not select this option to pass the POJO by reference, which also results in better performance.
Pass Caller's Subject	As an alternative to selecting a Service Account, select this option to have Oracle Service Bus pass the authenticated subject from the proxy service when invoking the EJB.
Service Account	Click **Browse** and select a JNDI service account from the list displayed. If no service account is specified, an anonymous subject is used. For more information, see Chapter 17, "Service Accounts."
Client JAR	Click **Browse** and select an EJB client JAR resource from the list displayed. The client JAR contains the remote or business interface for the remote service. The Client JAR is registered as a generic Archive Resource.
Home Interface	**EJB 2.1 only** – Select the required EJBHome interface from the options populated by the JAR.
Remote Interface	**EJB 2.1 only** – This field is automatically populated based on the configuration of the Home Interface.
Business Interface	**EJB 3.0 only** – Select the business interface from the client JAR that you want to invoke.
Target Namespace	This field is populated by information picked up from the JAR.
Methods	Select the required methods. Click + to expand the method, which lets you edit the default parameter values.
	You can change the default operation name for a given method. By default, the operation name is the method name. If an EJB contains methods with same name (overloaded), you must change the operation names so that they are unique. WSDLs require unique operation names.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.15 JMS Transport Configuration Page

The Create/Edit a Business Service - JMS Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "JMS Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the JMS transport protocol. Table 19–14 describes how to use the page.

Table 19–14   Create/Edit a Business Service - JMS Transport Configuration Page

Option	To create or edit...
Destination Type	Select a type for the JMS bridge destination: ■ **Queue** (for point-to-point) ■ **Topic** (for publish/subscribe)
Message Type	Select one of the following: ■ **Bytes** (for a stream of uninterpreted bytes) ■ **Text** (for text messages) This option is disabled if you select a Message Type of Java for the response.
Response Queues	This option is available only when Queue is selected for the Destination Type. Select one of the following response options: ■ **None** – No response is expected. Set this option for one-way operations. ■ **One for all Request URIs** – Lets you enter a single URI to handle the response, as well as set other response configuration details such as encoding and timeout, and optionally select a JMS Service Account for passing JMS/JNDI credentials. ■ **One per Request URI** – This option provides response failover, letting you enter a response URI or destination for each request URI. You can optionally select a service account for JMS/JNDI credentials on each request/response pairing.
Response Pattern	This option is available only when you select a response option in the Response Queue field. Select one of the following: ■ Select **JMSCorrelationID** for all services other than JAX-RPC services running on Oracle WebLogic Server. ■ Select **JMSMessageID** for JAX-RPC services running on Oracle WebLogic Server. For more information, see "Message ID and Correlation ID Patterns for JMS Request/Response" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dispatch Policy	Select the instance of Oracle WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For example, if the business service has a JMS transport protocol, the business service endpoint is an MDB (message-driven bean) JAR file that you can associate with the specific dispatch policy. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Request Encoding	Enter the character set for encoding requests. The default is UTF-8.

Table 19–14 (Cont.) Create/Edit a Business Service - JMS Transport Configuration Page

Option	To create or edit...
Response Encoding	This option is available only when one of the response options is selected in the Response Queues field.
	Enter the character set for encoding responses. The default is UTF-8.
Response Timeout	This option is available only when one of the response options is selected in the Response Queues field.
	Enter the amount of time, in seconds, to wait for the response before dropping the connection. The default, zero (0), means the response never times out.
Client Jar	This option is available when the service is a Messaging Service with a response type of Java. Select the client JAR to be used for dequeueing messages that contain Java Objects. Selecting the client JAR ensures it is on the classpath.
	For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Response URI	This option is available when you select the One for all Request URIs response option and the JMSCorrelationID response pattern.
	Enter a response URI in the format:
	jms://host:port/connection_factory/jndi_destination
	To target multiple servers, use the following format:
	jms://host1:port,host2:port/connection_factory/jndi_destination
	You can also omit the host and port in the response URI. For example:
	jms:///connection_factory/jndi_destination
	When you omit host and port, the connection factory/destination lookup occurs on the local server. This is useful, for example, if the request URI goes to a foreign connection factory/destination, but you want the response sent to the local server.
	Note: While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.
Request Responses	This option is available when you select the One per Request URI response option for the JMSCorrelationID pattern to provide response failover.
	For each request URI entered on the generic Transport page, enter a Response URI, as described in the previous Response URI field.
	You can select an optional Service Account for JMS/JNDI credentials that the service uses for both the request and response queues.
Target - Responses	This option is available when you select the One for all Request URIs response option for the JMSMessageID pattern.
	Enter the name of the Target server that is to receive responses, and enter a Response URI, as described in the Response URI field.

Table 19–14 (Cont.) Create/Edit a Business Service - JMS Transport Configuration Page

Option	To create or edit...
Request Connections	This option is available when you select the One per Request URI response option for the JMSMessageID pattern to provide response failover.
	For each request URI, identified sequentially by number in the Seq.No field, optionally enter a JMS **Connection Factory** name. If you do not enter a name, the JMS transport uses the connection factory from the request URI.
	You can select an optional **Service Account** for JMS/JNDI credentials that the service uses for both the request and response queues.
Target - Destinations	This option is available when you select the One per Request URI response option for the JMSMessageID pattern to provide response failover. Use this field in conjunction with Request Connections.
	Each Target server listed (determined by the servers in the current domain), such as Managed Servers in a cluster, lists the request URIs by Seq.No, which correspond to those in the Request Connections field. For each request URI on each target, enter the **Destination** queue on that target server that receives responses.
	Note: Because the Oracle Service Bus development environment supports only a one-server environment, only one Target is listed. To configure this field in a multi-server environment, deploy this service to the runtime environment and complete the service configuration in the Oracle Service Bus Administration Console.
JMS Service Account	This option is available when you use the None or One for all Request URIs options in the Response Queues field.
	Select a service account to use for the JMS resource managed by the JMS server. A service account is an alias resource for a user ID and password, used for both the request and response. The same service account is used for both JMS and JNDI purposes.
	This field is mutually exclusive with the Pass Caller's Subject option. Use one or the other for JMS/JNDI authentication.
	For more information, see Section 17, "Service Accounts."
Use SSL	Select this option only if the requests are made over a TLS/SSL connection.
	TLS/SSL (Secure Sockets Layer) provides secure connections by allowing two applications connecting over a network to authenticate the other's identity and by encrypting the data exchanged between the applications. Authentication allows a server, and optionally a client, to verify the identity of the application on the other end of a network connection. Additionally, if the administrator has restricted access to individual JMS destinations (queues or topics) by setting access control on the JNDI entry for the destination, the service must authenticate when looking up the entry in the JNDI tree. Authenticate using a Service Account or the Pass Caller's Subject option.
	Note: The JMS transport does not support two-way SSL.
Expiration	The time interval in milliseconds after which the message expires. Default value is 0, which means that the message never expires.
Enable Message Persistence	The JMS delivery mode the service uses, which lets you balance reliability with throughput. Select this option (default) for guaranteed message delivery. Deselect this option to improve throughput if an occasional lost message is tolerable.

Table 19–14   (Cont.) Create/Edit a Business Service - JMS Transport Configuration Page

Option	To create or edit...
Unit of Order	Enter a message unit-of-order. Message unit-of-order enables message producers to group messages into a single unit with respect to the processing order. This single unit-of-order requires that all messages in that unit be processed sequentially in the order they were created.
Pass Caller's Subject	Select this check box to have Oracle Service Bus pass the authenticated subject when sending a message.

When you enable this field and the business service targets JMS resources in a different domain, enable global trust on both domains. See Configuring Security for a WebLogic Domain in Securing Oracle WebLogic Server.

This field is mutually exclusive with the Service Account option. Use one or the other for JMS/JNDI authentication.

When you enable this field and the business service targets JMS resources in a different domain, enable global trust on both domains. See "Configuring Security for a WebLogic Domain" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*. |
| JNDI Timeout | The JNDI connection timeout (in seconds) used while looking up the destination or connection factory in the JNDI tree.

The default, zero (0), means the connection never times out. |

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.16 MQ Transport Configuration Page

The Create/Edit a Business Service - MQ Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "MQ Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Before you begin

Configure a MQ Connection resource. See Section 9, "MQ Connections."

To configure the MQ transport

Use this page to configure transport settings for a business service using the native MQ transport protocol. Table 19–15 describes how to use the page.

Table 19–15   Create/Edit a Business Service - MQ Transport Configuration Page

Option	To create or edit...
Message Type	Select one of the following: ■ **Bytes** (for a stream of uninterpreted bytes) ■ **Text** (for text messages)
Is Response Required	Select this option to specify that a response is expected after an outbound message is sent.

Table 19–15   (Cont.) Create/Edit a Business Service - MQ Transport Configuration Page

Option	To create or edit...
Response Correlation Pattern	This option is available only when the **Is Response Required** check box is selected. Specify whether the response correlation pattern should be based on: ■ MessageID ■ CorrelationID ■ Dynamic Queue – Select this option if your WebSphere MQ implementation uses dynamic queues for response correlation. The MQ transport supports only temporary dynamic queues.
Auto-generate Correlation Value	This option is available only when the **Is Response Required** check box is selected. Select this check box to automatically generate a CorrelationID or MessageID.
Model Queue	For Dynamic Queue Response Correlation Pattern only. Enter the name of the model queue used to generate the dynamic queue.
MQ Response URI	This option is available only when the **Is Response Required** check box is selected. The destination to which the response should be published. Enter a response URI in the same format as the endpoint URI: mq://local_queue_name?conn=mq_connection_resource or, if you are using dynamic queues: mq://dynamic_queue_prefix?conn=mq_connection_resource The dynamic_queue_prefix, which is limited to 32 characters, is used to create the dynamic queue on the MQ server. The queue name becomes the prefix plus a unique ID. For example, if the dynamic_queue_prefix is example, the dynamic queue would be named something like example123129083821. You can also use an asterisk (*) as a wildcard in the dynamic queue response URI. For example: mq://dynamic_queue_prefix*?conn=mq_connection_resource mq://dynamic_queue_prefix* mq://* If you do not provide a dynamic_queue_prefix in the URI, the transport uses the dynamic queue name generated by the MQ server. If you do not provide an explicit mq_connection_resource in the URI (best practice), the transport uses the mq_connection_resource from the endpoint URI. For more detailed information, see "MQ Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Response Timeout	This option is available only when the **Is Response Required** check box is selected. Enter the number of seconds to wait for a response before dropping the connection. The default is 300.
Polling Interval	This option is available only when the **Is Response Required** check box is selected. Enter a polling interval, in milliseconds. The default is 1000.
Poller Dispatch Policy	The dispatch policy to use for the poller threads.

Table 19–15   (Cont.) Create/Edit a Business Service - MQ Transport Configuration Page

Option	To create or edit...
Dynamic Queue Pooling	For Dynamic Queue Response Correlation Pattern only. Select this option if you want the service to use pooled connections to dynamic queues.
	To use a separate connection pool for dynamic queues, consider configuring a dedicated MQ Connection resource for the dynamic queues.
	Do not select this option to create a new dynamic queue instance on each request (and destroy the queue after the response).
Endpoint URI 'PUT' options	Enter the MQ PUT message options from among the following:
	■ MQC.MQPMO_ALTERNATE_USER_AUTHORITY
	■ MQC.MQPMO_DEFAULT_CONTEXT
	■ MQC.MQPMO_FAIL_IF_QUIESCING
	■ MQC.MQPMO_LOGICAL_ORDER
	■ MQC.MQPMO_NEW_CORREL_ID
	■ MQC.MQPMO_NEW_MSG_ID
	■ MQC.MQPMO_NO_CONTEXT
	■ MQC.MQPMO_NO_SYNCPOINT
	■ MQC.MQPMO_NONE
	■ MQC.MQPMO_PASS_ALL_CONTEXT
	■ MQC.MQPMO_PASS_IDENTITY_CONTEXT
	■ MQC.MQPMO_RESOLVE_LOCAL_Q
	■ MQC.MQPMO_SET_ALL_CONTEXT
	■ MQC.MQPMO_SET_IDENTITY_CONTEXT
	■ MQC.MQPMO_SYNCPOINT
	■ MQC.MQPMO_VERSION_1
	■ MQC.MQPMO_VERSION_2
	You can use either "|" or "+" to separate multiple options. For example, you can specify the following:
	MQC.MQPMO_LOGICAL_ORDER | MQC.MQPMO_NEW_MSG_ID
	The MQ PUT message options are applied when the message is placed in the outbound queue.
MQ Unrecognized Response URI	Enter the URI representing the queue to which unrecognized response messages should be sent. This setting is enabled only when the **Auto-generate Correlation Value** check box is selected.
	If you do not specify a value for this field, unrecognized response messages are deleted.
Process RFH2 Headers	Select this option to parse WebSphere MQ RFH2 headers from a message payload and automatically generate an RFH2Headers transport header containing the RFH2 data.
	If you do not select this option, the payload is passed through as received.
	For information about how the MQ transport handles RFH2 headers, see "About RFH2 Headers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–15 (Cont.) Create/Edit a Business Service - MQ Transport Configuration Page

Option	To create or edit...
Worker Thread Dispatch Policy	Dispatch policy refers to the instance of Oracle WebLogic Server Work Manager that you want to use for the service endpoint.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.17 SB Transport Configuration Page

The Create/Edit a Business Service - SB Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "SB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the SB (Service Bus) transport protocol. Table 19–16 describes how to use the page.

Table 19–16 Create/Edit a Business Service - SB Configuration Page

Option	To create or edit...
Dispatch policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Timeout	The amount of time in seconds it takes the service to time out.
	Note: The timeout will be ignored when the quality of service is Exactly-Once.
Service Account	Click **Browse** and select a service account from the list displayed. If no service account is specified, an anonymous subject is used.
	For more information:
	■ Section 17, "Service Accounts"
	■ Section 17.3, "Adding Service Accounts"

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.18 SFTP Transport Configuration Page

The Create/Edit a Business Service - SFTP Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "HTTP and Poller Transports" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the sftp transport protocol. Table 19–17 describes how to use this page.

Table 19–17   Create/Edit a Business Service - SFTP Transport Configuration Page

Option	To create or edit...
User Authentication	Select one of the following: ■ **Username Password Authentication** - Specifies that a static service account is associated with this authentication method and the client is authenticated using the provided credentials. ■ **Host Based Authentication** - Specifies that a user name and service key provider is required to use this authentication method. Any user connecting from a known host is authenticated using the private key of the host. ■ **Public Key Authentication** - Specifies that a user name and service key provider is required to use this authentication method. Every user has their own private key. **Note**: The Oracle Service Bus service does not use the service key provider to authenticate any credentials from the SFTP server. It uses only the known_hosts file to authenticate the SFTP server, as described in "Configuring Transport-Level Security for SFTP Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Service Account	Enter the service account for the user, or click **Browse** to select service accounts from a browser.
Service Key Provider	This option is available only when **Host Based** or **Public Key Authentication** is selected. Enter a service key provider in the **Service Key Provider** field. You can click **Browse** to select service key providers from a browser. This is a required field.
Username	This option is available only when **Host Based** or **Public Key Authentication** is selected. Enter the user name.
Timeout	Enter the socket timeout interval, in seconds, before the connection is dropped. If you enter `0`, there is no timeout. The default value is 60.
Prefix for destination File Name	Enter an optional prefix that the transport prepends to the file name on the remote server. Do not enter * in this field. This character causes a runtime exception.
Suffix for the destination File Name	Enter an optional suffix that the transport appends to the file name on the remote server. Do not enter * in this field. This character causes a runtime exception.
Request Encoding	Accept the default `UTF-8` as the character set encoding for requests in SFTP transports.

Table 19–17   (Cont.) Create/Edit a Business Service - SFTP Transport Configuration

Option	To create or edit...
Preferred Cipher Suite	Select the cipher suite to use when communicating with the server.
	If you use the default value, **Use Runtime Default**, the list of supported cipher suites is sent to the server and each is tried until a match is found.
Preferred Data Integrity Algorithm	Select the bulk-hashing algorithm for data integrity checks.
	If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, `hmac-sha1`. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Public Key Algorithm	Select the asymmetric key algorithm for public-key cryptography.
	If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, `ssh-dss`. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Key Exchange Algorithm	Select the default key exchange protocol for negotiating the session key for encrypting the message.
	If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, `diffie-hellman-group1-sha1`. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Compression Algorithm	Select whether to compress in-flight data using zlib. Select **zlib** to compress data; otherwise select **None**. The default is **None**.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.19 SOA-DIRECT Transport Configuration Page

Table 19–18 describes the transport-specific configuration options for the SOA-DIRECT transport.

For more information on the SOA-DIRECT transport, see "SOA-DIRECT Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 19–18   Create/Edit a Business Service – SOA-DIRECT Transport Configuration

Property	Description
JNDI Service Account	Optional. Specifies the security credentials for the JNDI lookup of the target SOA service. The service account must be static. Click **Browse** and select a service account. If you do not specify a service account, an anonymous subject is used.

Table 19–18 (Cont.) Create/Edit a Business Service – SOA-DIRECT Transport

Property	Description
Role	Required. Identifies the communication pattern the service uses. Select one of the following options: ■ **Synchronous Client** (default) – In this role, because there is no asynchronous callback, the Callback Proxy option is disabled. The WS-Addressing Version option is also disabled. ■ **Asynchronous Client** – In this role, because asynchronous callback is usually required, you can identify a Callback Proxy, and you must select a WS-Addressing Version. The asynchronous option is enabled only when the WSDL service is of type SOAP. ■ **Service Callback** – This role is for returning the asynchronous callback to an SOA service after an external service invocation. Load balancing or failover are not available for services in the Service Callback role.
Callback Proxy	Optional. Enabled only for the Asynchronous Client role. This option lets you specify the proxy service that receives callbacks. When you select a callback proxy, if no WS-Addressing is provided by the request or the proxy service pipeline, Oracle Service Bus automatically populates the ReplyTo property in the SOAP header. You must select a WSDL proxy service that uses the SB transport (for RMI), and the callback proxy service must understand WS-Addressing. WS-Addressing properties that are sent in the request or set in the proxy service pipeline are used instead of the callback proxy you set in this option. If you do not specify a Callback Proxy, and the request does not contain ReplyTo properties, you must provide ReplyTo properties in the SOAP header through the proxy service pipeline.
Fault Proxy	This option is not currently supported. You must configure your callback services to handle faults in an asynchronous pattern.
WS-Addressing Version	Required. Enabled only for the Asynchronous Client role. Specify the default WS-Addressing version to use when no WS-Addressing is provided in the request or the proxy service pipeline. WS-Addressing properties that are sent in the request or set in the proxy service pipeline are used instead of the WS-Addressing version you set in this option. For WS-Addressing version mismatches between environments, perform any necessary transformations in the proxy service pipeline. For more information, see the "Transformation Examples" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"

Table 19–18   (Cont.) Create/Edit a Business Service – SOA-DIRECT Transport

Property	Description
Pass Caller's Subject	Optional. Select this option to have Oracle Service Bus pass the authenticated subject from the proxy service when invoking the SOA service. The Invocation Service Account option, an alternative to Pass Caller's Subject, is disabled when you select this option.
	Note: Make sure that domain trust is enabled between client and target server if they are in different domains. For more information, see "Important Information Regarding Cross-Domain Security Support" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.
Invocation Service Account	Optional. Alternative to Pass Caller's Subject, which lets you specify custom security credentials by selecting a service account for RMI invocation. You can specify any type of service account (Pass Through, Static, Mapping).
	Click **Browse** and select a service account. If you do not specify a service account, an anonymous subject is used.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.20 Tuxedo Transport Configuration Page

The Create/Edit a Business Service - Tuxedo Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "Tuxedo Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the Tuxedo transport protocol. Table 19–19 describes how to use the page.

Table 19–19   Create/Edit a Business Service - Tuxedo Transport Configuration Page

Option	To create or edit...
Field Table Classes	Enter the name of the class or classes describing the `FML`/`FML32` buffer received. These are used for the `FML`/`FML32-to-XML` conversion routines to map field names to element names. This is a space separated list of fully qualified class names.
View Classes	Enter the name of the class or classes describing the `VIEW`/`VIEW32` buffer received or sent. These are used for the `VIEW`-to-`XML` or `VIEW32`-to-`XML` conversion routines to map field names to element names. This is a space separated list of fully qualified class names.
Classes Jar	Select a JAR Resource that contains a JAR file with the `FML`/`FML32` or `VIEW`/`VIEW32` classes necessary for this endpoint operation.

Table 19–19 (Cont.) Create/Edit a Business Service - Tuxedo Transport Configuration

Option	To create or edit...
Remote Access Point(s)	Select a remote access point from the available options associated with the Import. The list contains remote access points configured in WTC. A business service cannot be created if there is no associated remote access point.
	If no remote access points exist or to create a new one, select **New**. Enter the corresponding **Access Point Name** and **Network Address** in the adjacent fields. Upon validation of the endpoint, the access point is added to the WTC configuration for each WTC server. If no WTC server exists, one is created.
	If more than one URI has been specified, there will be one remote access point field per URI and the URI displays for informative purposes. If more than one URI exists, each requires a different remote access point. If the URI specified already corresponds to an existing WTC resource, the corresponding remote access point displays, but cannot be modified.
Local Access Point(s)	This field appears only when you select **New** in the **Remote Access Point** field.
	From the list, select a local access point to be associated with the newly created remote access point. If none exist or to create a new one, select **New**. Enter the corresponding **Local Access Point Name** and **Local Network Address** in the adjacent fields.
Request Buffer Type	Select the type of buffer that the remote Tuxedo service will receive.
Request Buffer Subtype	This option is enabled if the previous **Request Buffer Type** value is VIEW or VIEW32. Enter the buffer subtype with which to associate the request buffer.
Response Required?	Select this check box to indicate a bidirectional call. If not checked, the underlying `tpcall` is invoked with `TPNOREPLY` flag, and a null response is posted asynchronously.
Suspend Transaction?	Select this check box to suspend the transaction, if it exists. This is useful when the remote service does not support transactions.
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	This Work Manager is used to asynchronously post a null reply in the case of a one-way invocation.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Request Encoding	Specify a character set encoding for requests in Tuxedo transports.
Response Encoding	Specify a character set encoding for responses in Tuxedo transports.
Timeout	Specify the maximum amount of time (in seconds) that the transport runtime waits for replies; an integer value that is greater than or equal to 0. If not specified or set to zero (default), replies will time out at BLOCKTIME, the maximum number of seconds that the local WTC access point allows for a blocking call.

Table 19–19   (Cont.) Create/Edit a Business Service - Tuxedo Transport Configuration

Option	To create or edit...
Transformation Style	Select one of the following: ■ **None** - (default) The order of fields may not be respected. ■ **Ordered** - The fields are presented with all their occurrences in the correct order. ■ **Ordered and Grouped** - If the fields are logically structured as records, the fields are ordered by occurrence and grouped by record.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.21 WS Transport Configuration Page

The Create/Edit a Business Service - WS Transport Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

For more information, see "WS Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a business service using the WS transport protocol. Table 19–20 describes how to use the page.

Table 19–20   Create/Edit a Business Service - WS Transport Configuration Page

Option	To create or edit...
Response Timeout	Enter the number of seconds to wait for a response. Leaving this field blank indicates that there is no response timeout. The business service will wait for the duration of the sequence timeout configured in the RM policy. If you enter a zero (0) value, there is no timeout; as such, it will never time out.
Service Account	Enter a service account or click **Browse** to select one from the list displayed. The service account specifies the credentials to use when there is an HTTP basic authentication policy on this service.
Queue Error Messages	Select the check box to enable sending error messages to the configured error queue.
Error Queue URI	This option is available only when the **Queue Error Messages** check box is selected. Enter the URI of the JMS queue for storing error messages, in the following format: `jms://host:port/connFactoryJndiName/queueJndiName` **Note**: While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

Table 19–20   (Cont.) Create/Edit a Business Service - WS Transport Configuration Page

Option	To create or edit...
JMS Error Queue Service Account	This option is available only when the **Queue Error Messages** check box is selected.
	Enter a service account or click **Browse** to select one from the list displayed.
	The service account specifies the credentials to use for JNDI lookups and sending error messages to the error queue.
Use SSL for Error Queue	This option is available only when the **Queue Error Messages** check box is selected.
	Select the check box to use SSL for connecting to the error queue.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.22 SOAP Binding Configuration Page

The Create/Edit a Business Service - SOAP Binding Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to configure the SOAP Binding for a business service based on a WSDL.

Select **Enforce WS-I Compliance** to specify whether the service is to conform to the Basic Profile defined by the Web Services Interoperability Organization. This option is available for or SOAP 1.1 services only

When a service is marked WS-I compliant, checks are performed against the messages sent to and from that service.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.23 Message Handling Configuration Page

The Create/Edit a Business Service - Message Handling Configuration page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to specify how Oracle Service Bus handles message content. Table 19–21 describes the message handling options for business services. Different options are available for different transport types.

Table 19–21   Create/Edit a Business Service - Message Handling Configuration Page

Option	To create or edit...
XOP/MTOM Support	This feature lets you encode outbound messages in MTOM/XOP format. SOAP Message Transmission Optimization Mechanism (MTOM) is a method of sending binary data to and from Web services. MTOM uses XML-binary Optimized Packaging (XOP) to transfer the binary data.
	Oracle Service Bus supports XOP/MTOM using the following transports:
	■ HTTP/S
	■ Local
	■ SB
	Select the **Enabled** check box to enable the business service to encode outbound messages in MTOM/XOP format. This option is disabled for imported business services that are based on previous release configurations.
	If XOP/MTOM Support is enabled, select how to handle binary data in the $header and $body message context variables from among the following options:
	■ **Include Binary Data by Reference**: (Default) In an outbound response message, replace xop:Include elements with ctx:binary-content elements when setting up the $body message context variable.
	■ **Include Binary Data by Value**: In an outbound response message, replace xop:Include elements with Base64-encoded text versions of corresponding binary data when setting up the $body message context variable.
	If XOP/MTOM Support is enabled for a business service, it is not required that every outbound message be in the MTOM format. Instead, this setting specifies that the business service is capable of handling an MTOM payload.
	Since Oracle Service Bus does not support a combination of MTOM and SwA, the system issues a runtime error when Oracle Service Bus attempts to dispatch an outbound request to a business service and the business service is both MTOM/XOP-enabled and the $attachments message context variable is not null.
Attachments	Oracle Service Bus supports streaming MIME attachments using the HTTP/S transport.
	This feature lets you store attachments in outbound response messages to a disk file and then process the data in a streaming fashion without buffering the attachment contents in memory. This enables the business service to process large attachments robustly and efficiently.
	Select the **Page Attachments to Disk** check box to enable the business service to stream attachments in outbound response messages.
	If you enable XOP/MTOM Support, the Attachments option is only available if you choose the **Include Binary Data by Reference** option under XOP/MTOM Support. Note also that payloads that contain attachments must conform to RFC 822. Specifically, lines containing Internet headers need to be terminated with CRLF (carriage return line feed).

Table 19–21 (Cont.) Create/Edit a Business Service - Message Handling Configuration

Option	To create or edit...
Result Caching	If you invoke business services whose results seldom change, result caching improves business service performance by returning cached results to the client instead of invoking an external service directly.
	To use result caching for business services, you must first enable it globally, as described in Section 27.2.11, "Enabling Result Caching Globally."
	Configure result caching in the Advanced Settings section using the following guidelines:
	■ Select the **Supported** option. When you do, the remaining fields are enabled.
	■ **Expression Namespaces** – When using expressions to configure result caching, whether with the Cache Token Expression, the Expiration Time, or both, you can enter the namespaces and corresponding prefixes to use in the expressions. This field also lets you view a list of existing namespaces.
	■ **Cache Token Expression** – Oracle Service Bus uses cache keys to identify cached results for retrieval or population, and the cache token portion of the cache key provides the unique identifier. You can use an expression—the Cache Token Expression—to generate the cache token part of the cache key to uniquely identify a cached result for the business service.
	To generate the cache token from a value in the request (in the proxy service message flow that invokes the business service), use an expression that gets the value from the pipeline $body, $header, $operation, or $transportMetaData ($outbound/ctx:transport/ctx:request or $outbound/ctx:transport/ctx:response). For example, you can populate the cache-token from a customer ID in the message $body.
	The Cache Token Expression must resolve to a String or the value of simple content, such as an attribute or an element with no child elements. If the expression evaluates to null or causes an error, results are not cached.
	Note: You can also generate the cache token from the request, without setting a Cache Token Expression in the business service configuration. To do this, include a value in $outbound/ctx:transport/ctx:request/ctx:cache-token in the proxy service pipeline. Any value in that cache-token overrides the Cache Token Expression in the business service configuration.
	■ **Expiration Time** – Expiration Time, or time-to-live (TTL), determines when an entry in the business service's result cache is flushed. A duration of zero (0) means no expiration. A negative duration means do not cache. Select one of the following:

Table 19–21   (Cont.) Create/Edit a Business Service - Message Handling Configuration

Option	To create or edit...
Result Caching (continued)	■ **Use Default** – The default is the expiry-delay value in *DOMAIN_HOME*/config/osb/coherence/osb-coherence-config.xml. The default is 5 minutes.
	■ **Duration** – Set a specific length of time.
	■ **XQuery Expression** – You can enter an XQuery expression that gets an Expiration Time value from the request or response. Select **Request** or **Response**.
	To generate the expiration time from a value in the request or response, use an expression that gets the value from the pipeline $body, $header, $operation, or $transportMetaData ($outbound/ctx:transport/ctx:request or $outbound/ctx:transport/ctx:response). For example, use a value you have set in the Cache-Control HTTP header.
	Expiration Time must resolve to an Integer (representing seconds), an XQuery dayTimeDuration (XSD type), or the integer value of simple content representing seconds, such as an attribute or an element with no child elements. If the expression evaluates to null or causes an error, results are not cached.
	Note: You can also generate the expiration from the request, without setting an Expiration Time in the business service configuration. To do this, include a value in $outbound/ctx:transport/ctx:request/ctx:cache-ttl in the proxy service pipeline. Any value in the cache-ttl element overrides the Expiration Time in the business service configuration.
	For a more detailed description of result caching, see "Improving Performance by Caching Business Service Results" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

After you finish

Click **Next** to review and save this configuration on the Create/Edit a Business Service-Summary page.

19.2.24 Summary Page

The Create/Edit a Business Service - Summary page is one in a series of pages for creating and editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5, "Editing Business Service Configurations."

Use this page to view or modify the configuration settings for a business service before saving it.

To view or modify settings, click **Edit** in the row of the appropriate category (for example, General Configuration, Transport Configuration, and so on). The pages you can edit depend on what pages you configured when creating the business service. The following list shows all pages:

- Section 19.2.1, "General Configuration Page"
- Section 19.2.2, "Message Type Configuration Page"
- Section 19.2.3, "Transport Configuration Page"
- Section 19.2.4, "Protocol-Specific Transport Configuration Page"
- Section 19.5.2, "Business Service Policies Page"

■   Section 19.5.3, "Business Service Security Page"

19.3 Exporting a WSDL Associated with a Business Service

You can export the WSDL of a WSDL-based business service, so you can view or modify the WSDL in an external tool such as an IDE. The WSDL is exported as a JAR file.

This is different than the Export Resources functionality in the System Administration module, which you use to move and stage resources between two domains. See Section 29.2, "Exporting Resources."

Before you begin

You can only export a WSDL when you are outside a session. See Section 3.1, "Using the Change Center."

To export a WSDL

1.   Do either of the following:

   ■   Select **Resource Browser** > **Business Services** to display the Summary of Business Services page.

   ■   Select **Project Explorer**, then select the project or folder containing the business service you want to export as a WSDL. The Project/Folder View page is displayed.

2.   Click the **Export WSDL** icon in the row of the business service whose WSDL you want to export. A dialog box prompts you to open or save the exported JAR file.

3.   In the dialog box, click **Open** to open the file, or click **Save** to save it.

19.4 Locating Business Services

To locate business services:

1.   Do either of the following:

   ■   Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the business service.

   ■   Select **Resource Browser** > **Business Services**. The Summary of Business Services page displays the information shown in Table 19–22. For a more detailed description of the properties, see Section 19.5, "Editing Business Service Configurations."

2.   To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, under **Search**, enter the target's name, path, or both, and then click the **Search** button.

   The path is the project name and the name of the folder in which the business service resides.

   Wildcard characters * and ? are allowed. Search is case-sensitive.

   Click **View All** to display all business services in the domain. This clears the search parameters from the previous search.

Table 19–22   Summary of Business Services Page

Property	Description
Name	A unique name for the business service. Click the name to display the View a Business Service - Configuration Details age.
	See Section 19.5, "Editing Business Service Configurations."
Path	The path is the project name and the name of the folder in which the business service resides, for example, UDDI/BusServices/OSB_services.
	Click the path of a business service to display the Project/Folder View page that contains it.
Actions	Do any of the following:
	■ Click the **Launch Test Console** icon to invoke the Test Console, which you use to validate and test the design of your services and transformations. For business services, you can only use the Test Console at runtime; that is, when the session is activated. For transformations, you can use the Test Console whether you are inside or outside a session. See Section 33.1, "Testing Services" and Section 33.2, "Testing Transformations."
	■ The **Export WSDL** icon is displayed for WSDL-based business services. Click this icon to export a WSDL, which you can then view or modify in an external tools such as an IDE.
	This is different than the Export Resources functionality in the System Administration module, which you use to move and stage resources between two domains. See Section 4.28, "Exporting a WSDL."
	■ The **Generate WSDL** icon is displayed for transport-typed business services, such as EJB and JEJB (but not for Split-Join flow services). Click this icon to generate a WSDL, which you can then view or modify. See Section 4.29, "Generating a WSDL."
Options	Click the **Delete** icon to delete the service. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. For more information, see Section 19.6, "Deleting Business Services."

19.5 Editing Business Service Configurations

To edit a business service configuration:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

    You don't have to enter a session if you only want to view details.

2. Locate the business service you want to view or edit, as explained in Section 19.4, "Locating Business Services."

3. Click the business service name. The View a Business Service - Configuration Details page displays configuration information for the selected business service.

4. To view or modify settings, do either of the following:

    ■ Click the **Edit** icon next to the name of the category whose properties you want to view or edit (for example, General Configuration, Transport Configuration, and so on). The pages you can edit depend on what pages you configured when creating the business service.

    For a list of all those pages, see Section 19.5.1, "View a Business Service - Configuration Details Page."

- Click **Edit** at the bottom of the page to display the Create/Edit a Business Service-General Configuration page, which is the first page in the sequence of pages for configuring this business service.

5. Continue to view or edit, as described in Section 19.1, "Creating and Configuring Business Services."

6. On the Create/Edit a Business Service-Summary page, click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

19.5.1 View a Business Service - Configuration Details Page

The View Business Service - Configuration Details page displays the configuration details of a business service. Table 19–23 describes all the properties that can appear on this page. (Properties vary depending on the details of the business service.)

The categories listed on this page correspond to the Create/Edit a Business Service pages used for creating and editing business service configurations, as described in Section 19.2, "Create/Edit a Business Service - Page Reference."

Click the **Edit** link next to any category name to display the associated configuration page.

Table 19–23   View a Business Service: Configuration Details Page

Properties	Description
Last Modified By	The user who created this business service or imported it into the configuration.
Last Modified On	The date and time that the user created this business service or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this business service references. If such references exist, click the numeric link to view a list of the objects. To learn more, see Section 4.22, "Viewing References to Resources."
Referenced by	The number of objects that reference this business service. If such references exist, click the numeric link to view a list of the objects. To learn more, see Section 4.22, "Viewing References to Resources."
Description	A description of this business service, if one exists.
General Configuration	Shows properties configured on the General Configuration page.
Message Type Configuration	Shows properties configured on the Section 19.2.2, "Message Type Configuration Page."
Transport Configuration	Shows properties configured on the Section 19.2.3, "Transport Configuration Page."
BPEL-10g Transport Configuration	Shows properties configured on the Section 19.2.5, "BPEL-10g Transport Configuration Page."
DSP Transport Configuration	Shows properties configured on the Section 19.2.6, "DSP Transport Configuration Page."
EJB Transport Configuration	Shows properties configured on the Section 19.2.7, "EJB Transport Configuration Page."

Table 19–23   (Cont.) View a Business Service: Configuration Details Page

Properties	Description
Email Transport Configuration	Shows properties configured on the Section 19.2.8, "Email Transport Configuration Page."
File Transport Configuration	Shows properties configured on the Section 19.2.9, "File Transport Configuration Page."
Flow Transport Configuration	Shows properties configured on the Section 19.2.10, "Flow Transport Configuration Page."
FTP Transport Configuration	Shows properties configured on the Section 19.2.11, "FTP Transport Configuration Page"
HTTP Transport Configuration	Shows properties configured on the Section 19.2.12, "HTTP Transport Configuration Page."
JCA Transport Configuration	Shows properties configured on the Section 19.2.13, "JCA Transport Configuration Page."
JEJB Transport Configuration	Shows properties configured on the Section 19.2.14, "JEJB Transport Configuration Page."
JMS Transport Configuration	Shows properties configured on the Section 19.2.15, "JMS Transport Configuration Page."
MQ Transport Configuration	Shows properties configured on the Section 19.2.16, "MQ Transport Configuration Page."
SB Transport Configuration	Shows properties configured on the Section 19.2.17, "SB Transport Configuration Page."
SFTP Transport Configuration	Shows properties configured on the Section 19.2.18, "SFTP Transport Configuration Page."
SOA-DIRECT Transport Configuration	Shows properties configured on the Section 19.2.19, "SOA-DIRECT Transport Configuration Page."
Tuxedo Transport Configuration	Shows properties configured on the Section 19.2.20, "Tuxedo Transport Configuration Page."
WS Transport Configuration	Shows properties configured on the Section 19.2.21, "WS Transport Configuration Page."
Message Handling Configuration	Shows properties configured on the Section 19.2.23, "Message Handling Configuration Page."

19.5.2 Business Service Policies Page

The View a Business Service - Policies Configuration page is one in a series of pages for editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5.1, "View a Business Service - Configuration Details Page."

Use this page to configure policy settings for a business service with a service type of WSDL-based XML Web Service, Messaging Service, Any SOAP Service, or Any XML Service. Table 19–24 describes how to use the page. In order for Oracle WSM policies to be used with WSDL-based XML Web Service, Messaging Service, or Any XML Service proxy services, the protocol must be set to HTTP.

For WSDL-based services, WLS 9.2 policies bound to the service are exposed (inlined) in the effective WSDL. Abstract policies are pre-processed before they are inlined. Oracle WSM policies are bound by reference, not inlined in the effective WSDL.

Table 19–24   View a Business Service - Policies Page

Option	To edit...
Service Policy Configuration	Following are the options in the Service Policy Configuration field: **Oracle WSM Policies** ■ **From OWSM Policy Store** – Policies are defined in the central Oracle Web Services Manager Policy Store managed by Oracle Enterprise Manager. These policies support WS-Security 1.0/1.1, SAML 1.1/2.0, KerberosToken Profile, and other industry standards. With this option, you can add polices from a predefined list. Click **Add** under Service Level Policies to display the list. Use filtering if necessary to locate the policies you want. For business services, only client-side policies are displayed. If a WSDL used to create a business service contains embedded standard WS-Security policies, Oracle Service Bus throws an error and displays a conflict. To resolve this conflict, replace the embedded WSDL policies with compatible Oracle Web Services Manager policies. To do this, select **From OWSM Policy Store** and click **Add Compatible**. Oracle Service Bus makes a best-effort attempt at finding the closest matching policy from the Oracle Web Services Manager policy store based on the standard policy embedded in the WSDL. The algorithm may return zero, one, or multiple matching policies. If after you click **Add Compatible** Oracle Service Bus finds no compatible policies, click **Add** and select the appropriate Oracle Web Services Manager policies. If Oracle Service Bus returns more than one compatible policy, delete the policies you do not want to use. **Note**: This option is activated only if you have configured your domain to use Oracle Web Services Manager. For more information, see "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. **WLS 9 Policies** ■ **From WSDL** – Select this option if the service policy is associated with the WSDL upon which the service is based. These policies support WS-Security 1.0, SAML 1.1, and other industry standards. With this option you can view (read-only) request and response policies from the WSDL. **Note**: If you receive an error in the business service configuration about WS-Security Policies not supported by Oracle Service Bus, use the **From OWSM Policy Store** option to replace the embedded WSDL policies with compatible Oracle WSM policies. ■ **From Pre-defined Policy or WS-Policy Resource** - Select this option to add service-level policies, operation-level policies (in which case the policy applies to both the request and response messages), request policies, and response policies from the Administration Console. Policies are either pre-defined in Oracle WebLogic Server or user-defined in Oracle Service Bus with a WS-Policy resource. These policies support WS-Security 1.0, SAML 1.1, and other industry standards. With this option you can add policies at the service, operation, request, and response levels. Click **Add** under the appropriate section to select policies from a list. **Note**: The policy binding models are mutually exclusive. You must use only one type of policy in a service. If you bind policies directly to the service, all WSDL-based policies are ignored. After adding policies, you can provide overrides on the Security page.

After you finish

Click **Update** to save this configuration; or click **Reset** to undo your changes.

19.5.3 Business Service Security Page

The View a Business Service - Security Configuration page is one in a series of pages for editing business services, as described in Section 19.1, "Creating and Configuring Business Services" and Section 19.5.1, "View a Business Service - Configuration Details Page."

The fields available on this page depend on the use of policies in the business service. For example, if the service uses Oracle WSM policies (recommended), Policy Overrides appear on the Security page. For more information, see "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

WLS 9.2 policies bound to the service are exposed (inlined) in the effective WSDL. Abstract policies are pre-processed before they are inlined. Oracle WSM policies are bound by reference, not inlined in the effective WSDL.

Use this page to configure security settings for a business service that has a WSDL-based policy or that uses pre-defined policy bindings. Table 19–25 describes how to use the page.

Table 19–25   View a Business Service - Security Page

Option	To edit...
Service Account	Click **Browse** and select a service account from the list displayed. If no service account is specified, an anonymous subject is used.
	Note: This option is not available for services that use Oracle Web Services Manager Policies, because user credentials are provided by override keys.
	For more information:
	■ Section 17.1, "Specifying Service Accounts"
	■ Section 17.3, "Adding Service Accounts"
Policy Overrides	For Oracle WSM policies, provide any desired overrides that are allowed.
	For more information, see Table 19–24 and "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

After you finish

Click **Update** to save this configuration; or click **Reset** to undo your changes.

19.6 Deleting Business Services

To delete a business service:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Business Services** to display the Summary of Business Services page.

3. Click the **Delete** icon in the row of the business service you want to delete. The business service is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

20

Proxy Services: Creating and Managing

This chapter describes how to create, configure, and manage proxy services using the Oracle Service Bus Administration Console.

Proxy services are Oracle Service Bus definitions of services implemented locally on WebLogic Server.

For more information, see "Configuring Proxy Services and Business Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

20.1 Creating and Configuring Proxy Services

This section describes how to create and configure proxy services.

For JCA services, you can generate proxy services from JCA Bindings, as described in Section 20.1.1, "Generating a Proxy Service from a JCA Binding Resource."

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Project Explorer**, then select the project or folder to which you want to add the proxy service.

3. On the Project/Folder View page, select **Proxy Service** from the **Create Resource** list.

4. On the General Configuration page, provide a name for the service and select the type of service to create.

    Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. Click **Next**. The pages that follow depend on the choices you made on the first page. Enter the appropriate information on each of the subsequent pages, until you reach the summary page, then click **Save** to save the service in the current session.

    For detailed instructions on completing each page, see the following:

    - Section 20.2.1, "General Configuration Page"

    - Section 20.2.2, "Message Type Configuration Page"

    - Section 20.2.3, "Transport Configuration Page"

    - Section 20.2.4, "Protocol-Specific Transport Configuration Pages"

    - Section 20.2.17, "Operation Selection Configuration Page"

    - Section 20.2.18, "Message Handling Page"

    - Section 20.2.19, "Summary Page"

- Section 20.5.2, "Proxy Service Policy Page"

- Section 20.5.3, "Proxy Service Security Page"

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

7. After you create a proxy service, the next step is to configure its message flow. The message flow defines the implementation of a proxy service. To learn more, see Section 22.1, "Viewing and Editing Message Flows."

20.1.1 Generating a Proxy Service from a JCA Binding Resource

You can generate a JCA proxy service from an inbound JCA Binding resource in Oracle Service Bus. For more information on JCA Binding resources, see Chapter 13, "JCA Bindings."

To generate a JCA proxy service from a JCA Binding:

1. In the Oracle Service Bus Administration Console, click **Create** or **Edit** in the Change Center if you area not already in Create or Edit mode.

2. In the Resource Browser, click **JCA Bindings**.

3. Locate the JCA Binding from which you want to generate a service, and click the **Action** icon.

4. In the window that appears, confirm the name of the WSDL and the service you want to generate, select the location for these new resources, and click **Generate**.

   Oracle Service Bus generates the service and its corresponding WSDL.

5. Modify any other configuration details on the generated service as appropriate, such as the Endpoint URI.

20.2 Create/Edit a Proxy Service - Page Reference

Create a proxy service by selecting **Proxy Service** from the **Create Resource** list on the Project/Folder View page. When you select that option, it displays the first in a series of pages for configuring and adding proxy services. The pages displayed vary, depending on the options you choose along the way. The pages are:

- Section 20.2.1, "General Configuration Page"

- Section 20.2.2, "Message Type Configuration Page"

- Section 20.2.3, "Transport Configuration Page"

- Section 20.2.4, "Protocol-Specific Transport Configuration Pages"

- Section 20.2.17, "Operation Selection Configuration Page"

- Section 20.2.18, "Message Handling Page"

- Section 20.2.19, "Summary Page"

- Section 20.5.2, "Proxy Service Policy Page"

- Section 20.5.3, "Proxy Service Security Page"

20.2.1 General Configuration Page

The Create/Edit a Proxy Service - General Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating

and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to modify general configuration settings for a proxy service.

When you create a proxy service, this is the first page displayed in a series of pages for configuring the service. The pages displayed after this one differ depending on the choices you make on this page.

Table 20–1 describes how to use the page.

Table 20–1   Create/Edit a Proxy Service - General Configuration Page

Option	To create or edit...
Service Name	Enter a unique name for the proxy service.
	Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.
Description	Enter a description.
Service Type	Select the type of proxy service to create:
	■ **WSDL Web Service** - Select this option to create a proxy service based on a WSDL. Then enter the WSDL name, qualified by its path (for example, myProject/myFolder/myWSDL). Alternatively, click **Browse** to select a WSDL from the Select a WSDL Definition page.
	(port or binding) - Enter the name of a port (defined in the WSDL) to describe an actual transport address, or enter the name of a binding (defined in the WSDL) to map to a transport address. If you use **Browse** to select a WSDL, the Select a WSDL Definition page lists any defined ports and bindings.
	Note: If you are going to use the SOAP Body Type for operations, ensure that the WSDL does not have two operations with the same input message. The SOAP Body Type operation cannot be uniquely identified by inspecting the input message.
	■ **Transport Typed Service** - Select this option to create a JEJB proxy service.
	■ **Messaging Service** - Select this option to create a service that can receive messages of one data type and respond with messages of a different data type. These exchanges can be either request/response or one-way.
	(HTTP GET is supported only in the Any XML Service and Messaging Service service types.)
	■ **Any SOAP Service** - Select this option to create a SOAP service that does not have an explicitly defined, concrete interface.
	Select **SOAP 1.1** or **SOAP 1.2** from the list to specify the SOAP version to be used.
	■ **Any XML Service** - Select this option to create an XML service that does not have an explicitly defined, concrete interface.
	(HTTP GET is supported only in the Any XML Service and Messaging Service service types.)
	■ **Business Service** - Select this option to create a proxy service with a route node that routes to the business service you select. You cannot create a proxy service from a transport typed business service. If you create a proxy service from a DSP transport business service, Oracle Service Bus switches the transport type of the proxy service to HTTP, because the DSP transport cannot be used for proxy services. You can change the transport type of the proxy service to any other available transport.
	Enter the path (project/folder) and the name of the business service; or click **Browse** to select the business service from the Select Business Service page.

Table 20–1 (Cont.) Create/Edit a Proxy Service - General Configuration Page

Option	To create or edit...
Service Type (continued)	■ **Proxy Service** - Select this option to clone an existing proxy service. Since Oracle Service Bus does not accept the same URI for multiple services, you must change the URI for the cloned service.
	Note: When a service is created from another service, alert rules are maintained in the following way:
	■ When a proxy service is created from a business service or a business service is created from a proxy service, the alert rules, if any, are removed.
	■ When a proxy service is created from another proxy service or a business service is created from another business service, the alert rules, if any, are retained.
Service Key Provider	If needed, enter the path (project/folder) and name of a service key provider, or click **Browse** to select one from the Select Service Key Provider page.
	A service key provider is only required in certain cases:
	■ Outbound two-way TLS/SSL, where the proxy service routes messages to HTTPS services that require client-certificate authentication.
	■ In some Web Service security scenarios, for example, if the proxy service requires messages to be encrypted.
	To add a Web service security-enabled proxy service, you must create the proxy service from a WSDL (port or binding) with WS-Policy attachments.
	For more information, see Chapter 18, "Service Key Providers." To learn how to create a service key provider, see Section 18.2, "Adding Service Key Providers."
Publish to Registry	This option appears only when a default UDDI registry exists.
	Select this option to publish the proxy service to the default registry automatically. For more information, see Section 30.8, "Using Auto-Publish."

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.2 Message Type Configuration Page

The Create/Edit a Proxy Service - Message Type Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure message types for a proxy service whose type is Messaging Service.

The binding definition for messaging services consists of configuring the content-type of the messages that are exchanged. The content-type for the response does not have to be the same as for the request; therefore, the response is configured separately (for example, the service could accept an MFL message and return an XML acknowledgment receipt).

> **Note:** Email, File, FTP, or SFTP transport proxy services whose type is Messaging Service support one-way messaging *only*; the **Response Message Type** should be none. If you select an option other than none, the Email, File, FTP, or SFTP protocols will not be available on the Transport Configuration page.

Table 20–2 describes how to use Create/Edit a Proxy Service - Message Type Configuration page.

Table 20–2   Create/Edit a Proxy Service - Message Type Configuration Page

Option	To create or edit...
Request Message Type	Select a message type for the request message:
	■ **None** - Select this option if there is no request message.
	■ **Binary** - Select this option if the content-type of the message is unknown or not important.
	■ **Text** - Select this option if the message can be restricted to text.
	■ **MFL** - Select this option if the message is a binary document conforming to an MFL definition. Enter the MFL file name (qualified by its path), or click **Browse** to select a file.
	You can configure only one MFL file.
	Note: To support multiple MFL files, define the content as binary or text and use the MFL action in the message flow to convert to XML.
	■ **XML** - Select this option if the message is an XML document. Enter the XML file name (qualified by its path), or click **Browse** to select a file.
	Optionally provide some type information by declaring (in the **element or type** field) the XML schema type of the XML document exchanged.
	■ **Java** - Select this option if a Java Object is being sent in the request. The JMS transport is used for Java Object messages. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

Table 20–2   (Cont.) Create/Edit a Proxy Service - Message Type Configuration Page

Option	To create or edit...
Response Message Type	Select a message type for the response message:

- **None** - Select this option if there is no response message.

  Note: If you are using a Reply action in the pipeline to propagate success or failure messages from a service to the calling client, select an option other than None. The None option blocks the reply.

- **Binary** - Select this option if the content-type of the message is unknown or not important.

- **Text** - Select this option if the message can be restricted to text.

- **MFL** - Select this option if the message is a binary document conforming to an MFL definition. Enter the MFL file name (qualified by its path), or click **Browse** to select a file.

  You can configure only one MFL file.

  Note: To support multiple MFL files, define the content as binary or text and use the MFL action in the message flow to convert to XML.

- **XML** - Select this option if the message is an XML document. Enter the XML file name (qualified by its path), or click **Browse** to select a file.

  Optionally provide some type information by declaring (in the **element or type** field) the XML schema type of the XML document exchanged.

- **Java** - Select this option if a Java Object is being received in the response. The JMS transport is used for Java Object messages. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.3 Transport Configuration Page

The Create/Edit a Proxy Service - Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to select a transport protocol for the proxy service and to set other general transport configuration settings. Table 20–3 describes how to use the page.

> **Note:** Inbound transport-level security applies to the client applications and Oracle Service Bus proxy services. Outbound transport-level security applies to the connections between Oracle Service Bus proxy services and business services. To learn more about transport-level security, see "Configuring Transport-Level Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 20–3   Create/Edit a Proxy Service - Transport Configuration Page

Option	To create or edit...
Protocol	Select a transport protocol from the list. The protocols available differ, depending on the service type you are creating:
	■ **WSDL Web Service**: http, jca, jms, local, sb, ws
	■ **Transport Typed**: jejb
	■ **Messaging Service**: email, file, ftp, http, jms, local, mq (if available), sftp, tuxedo
	■ **Any SOAP Service**: http, jms, local, sb
	■ **Any XML Service**: email, file, ftp, http, jms, local, mq (if available), sb, sftp, tuxedo

Table 20–3   (Cont.) Create/Edit a Proxy Service - Transport Configuration Page

Option	To create or edit...
Endpoint URI	Enter an endpoint URI in the format based on the transport protocol you selected in the **Protocol** field, above: The formats are:

- **email** - `mailfrom:mail-server-hostname:mail-server-port`

- **file** - `file:///drivename:/somename`

- **ftp** - `ftp://hostname:port/directory`

- **http** - `someName`

  The HTTP transport supports both `HTTP` and `HTTPS` endpoints.

- **jca** - `jca://<resource_adapter_jndi>`

- **jejb** – The URL format is `ejb_jndi_name`. The URI configured for a JEJB proxy service becomes the global JNDI name for locating the stateless session bean generated by the JEJB transport from the remote/business interface in the client JAR.

  Note: For EJB 3.0, `ejb_jndi_name` is the mappedName attribute of the @javax.ejb.Stateless annotation in the generated bean. The lookup JNDI name for the generated EJB service is suffixed with `#interface_class`, which is the fully qualified name of the business interface.

  You can access the JEJB proxy service as:

  EJB 2.1: `protocol://host:port/ejb_jndi_name`

  EJB 3.0: `protocol://host:port/ejb_jndi_name#interface_class`

  Note: If your EJBs are running on IBM WebSphere, `ejb_jndi_name` must be in the following format:

  `cell/nodes/node_name/servers/server_name/ejb_jndi_name`

  or

  `cell/clusters/cluster_name/ejb_jndi_name`

  as described in the IBM WebSphere documentation at
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ib
  m.websphere.nd.iseries.doc/info/iseriesnd/ae/rnam_example_prop3.html

- **jms** - `jms://host:port/factoryJndiName/destJndiName`

  To target a target a JMS destination to multiple servers, use the following URI format:

  `jms://host1:port,host2:port/QueueConnectionFactory/DestName`

  You can also omit host and port from the URI to have the lookup performed on the local machine. For example:

  `jms:///connection_factory/jndi_destination`

  When you create a proxy service, you can configure a JMS endpoint URI even if the server at that endpoint if not available. However, in the case of JMS, when you activate the session, the endpoint must be available. To learn more, see Section 3.2, "Activating Sessions."

  Note: While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the endpoint URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the endpoint URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

  In a cluster: The host names in the JMS URI must exactly match the host names of the cluster servers as they are configured in WebLogic Server.

- **local**

  This transport does not require an endpoint URI.

Table 20–3   (Cont.) Create/Edit a Proxy Service - Transport Configuration Page

Option	To create or edit...
Endpoint URI (continued)	■ **mq** - `mq://<local-queue-name>?conn=<mq-connection-resource-ref>` `local-queue-name` is the name of the MQ queue from which the proxy service reads messages. `mq-connection-resource-ref` is the path (project/folder) and name of the MQ connection resource; for example, `default/my_MQconnection`. **Note**: The Endpoint URI cannot contain spaces, so do not create MQ Connection resources or projects/folders with spaces in the names. To make the MQ transport available in Oracle Service Bus, see Chapter 9, "MQ Connections." ■ **sb** - `service_name` `service_name` is the unique identifier for the proxy service. By default, this name will be the proxy service name. `service_name` must only contain characters permitted in URIs (as described in RFC2396 at `http://www.ietf.org/rfc/rfc2396.txt`), except it cannot contain forward slash (/) or colon (:) characters. ■ **sftp** - `sftp://hostname:port/directory` ■ **tuxedo** - `exportname` The URI `exportname` corresponds to a WTC Export that the remote Tuxedo domain identifies as a Tuxedo service. If more than one URI is specified, you must have unique resource names for the endpoints. If no remote name is specified, its value is the value of the resource name. If no remote name is entered or if remote and resource name are the same, only one URI is allowed. In this case resource name and remote name will have the same value. This allows users using already defined WTC Imports to make use of WTC load-balancing and failover. **Note:** If you configure two identical URIs, an error indicates that the service name already exists. ■ **ws** - `/contextPath` `contextPath` must be unique for proxy services that use either HTTP or WS transport.
Get All Headers	Select **Yes** to retrieve all the headers from the transport. Select **No** to retrieve a defined set of headers. If you select **No**, enter a set of headers in the **Header** field, then click **Add**. (This step does not apply to Local transport.) **Note**: Oracle Service Bus does not pass the HTTP Authorization header from the request to the pipeline because it opens a security vulnerability. You could inadvertently create a log action that writes the user name and unencrypted password to a log file. If your design pattern requires the HTTP Authorization header to be in the pipeline, do the following: a. In the startup command for Oracle Service Bus, set the following system property to true: `com.bea.wli.sb.transports.http.GetHttpAuthorizationHeaderAllowed` b. In the Oracle Service Bus Administration Console, on the Transport Configuration page, select **Get All Headers** or select **User-specified Headers** and specify Authorization. c. Restart Oracle Service Bus. Oracle Service Bus will pass the Authorization header to the pipeline.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.4 Protocol-Specific Transport Configuration Pages

The Create/Edit a Proxy Service-*Transport-Type* Configuration pages are each one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

This page differs, depending on what kind of transport you chose to configure. It can be:

- Section 20.2.5, "Email Transport Configuration Page"
- Section 20.2.6, "File Transport Configuration Page"
- Section 20.2.7, "FTP Transport Configuration Page"
- Section 20.2.8, "HTTP Transport Configuration Page"
- Section 20.2.9, "JCA Transport Configuration Page"
- Section 20.2.10, "JEJB Transport Configuration Page"
- Section 20.2.11, "JMS Transport Configuration Page"
- Section 20.2.12, "MQ Transport Configuration Page"
- Section 20.2.13, "SB Transport Configuration Page"
- Section 20.2.14, "SFTP Transport Configuration Page"
- Section 20.2.15, "Tuxedo Transport Configuration Page"
- Section 20.2.16, "WS Transport Configuration Page"

20.2.5 Email Transport Configuration Page

The Create/Edit a Proxy Service - Email Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure transport settings for a proxy service using the email transport protocol. Table 20–4 describes how to use the page.

Table 20–4   Create/Edit a Proxy Service - Email Transport Configuration Page

Option	To create or edit...
SSL Required	Select this option to communicate using Secure Sockets Layer. The default SSL port numbers are: - POP3(S): 995 - IMAP: 993 **Note**: The email server certificate must be present in the Oracle Service Bus truststore in order for SSL to work. The Email Transport supports one-way SSL communication.
Service Account	Enter a service account name, or click **Browse** to select service accounts from a browser.
Managed Server	This field is available only in a clustered domain. Select the Managed Server to act as the polling server. All of the Managed Servers can process the message, but only one can poll for the message.
Polling Interval	Enter a polling interval, in seconds.
Email Protocol	Select POP3 or IMAP as the server type for the email account.

Table 20–4   (Cont.) Create/Edit a Proxy Service - Email Transport Configuration Page

Option	To create or edit...
Read Limit	Specify the maximum number of messages to read per polling sweep. Enter 0 to specify no limit.
Pass By Reference	Select this check box to stage the file in the archive directory and pass it as a reference in the headers.
	By default when you create a new service, the **Pass By Reference** option is selected and you must specify the archive directory location.
Pass Attachments by Reference	Select this check box to stage the attachments in the archive directory and pass them as a reference in the headers.
	By default, when the **Pass By Reference** option is selected, the **Pass Attachments By Reference** option is implicitly true and you must specify the archive directory location.
Post Read Action	Select what happens to a message after it has been read:
	■ **Archive** - The message is archived.
	■ **Delete** - The message is deleted.
	■ **Move** - The message is moved. Move is only available with the IMAP protocol.
Attachments	Select how attachments are handled:
	■ **Archive** - Attachments are saved to the archive directory.
	■ **Ignore** - Attachments are ignored.
	Note: If attachments are archived, the attachment files are passed as a reference in the message headers irrespective of the settings for the Pass By Reference parameter.
IMAP Move Folder	Enter the folder to which the message is moved if the **Post Read Action** field is set to **Move**.
Download Directory	Enter a temporary location for downloading emails.
Archive Directory	Specify the path to the archive location if the **Post Read Action** field is set to **Archive**.
	This field is required if the **Pass By Reference** or **Pass Attachments By Reference** option is selected.
Error Directory	Enter the file system directory path to write the message and any attachments if there is a problem.
Request Encoding	Accept the default ISO-8859-1 as the character set encoding for requests in Email transports, or enter a different character set encoding.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.6 File Transport Configuration Page

The Create/Edit a Proxy Service - File Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure transport settings for a proxy service using the file transport protocol. Table 20–5 describes how to use this page.

Table 20–5   Create/Edit a Proxy Service - File Transport Configuration Page

Option	To create or edit...
File Mask	Specifies the files that should be polled by the proxy service. If the URI is a directory and *.* is specified, then the service will poll for all the files in the directory. Only the wildcard characters * and ? are allowed in the File Mask. Regular expressions are not supported.
Managed Server	This field is available only in a clustered domain. Select the Managed Server to act as the polling server. All of the Managed Servers can process the message, but only one can poll for the message.
Polling Interval	Enter a polling interval, in seconds. The default is 60.
Read Limit	Specify the maximum number of messages to read per polling sweep. Enter 0 to specify no limit. The default is 10.
Sort By Arrival	Select this check box to specify that events are delivered in the order of arrival. When this option is selected for a proxy service that is executed in a clustered environment, messages are always sent to the same server. In other words, load balancing across servers is ignored when this option is selected.
Scan Subdirectories	Select this check box to recursively scan all the directories.
Pass By Reference	Select this check box to stage the file in the archive directory and pass it as a reference in the headers.
Post Read Action	Select what happens to a message after it has been read: ■ **Archive** - The message is archived. ■ **Delete** - The message is deleted.
Stage Directory	Enter an intermediate directory to temporarily stage the files while processing them. Do not put the stage directory inside of the polling directory (the directory identified in the URL of the file transport proxy service; for example, file:///c:/dir1/dir2).
Archive Directory	Specify the path to the archive location if the **Post Read Action** option is set to **Archive**. The Archive Directory field is also a required field if you have selected the Pass By Reference field. Do not put the archive directory inside of the polling directory.
Error Directory	Enter the location where messages and attachments are posted if there is a problem. Do not put the error directory inside of the polling directory.
Request Encoding	Accept the default UTF-8 as the character set encoding for requests in file transports, or enter a different character set encoding.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.7 FTP Transport Configuration Page

The Create/Edit a Proxy Service - FTP Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure transport settings for a proxy service using the ftp transport protocol. Table 20–6 describes how to use this page.

Table 20–6   Create/Edit a Proxy Service - FTP Transport Configuration Page

Option	To create or edit...
User Authentication	Select **anonymous** if the user of the FTP server is anonymous, or select **external user** if the user of the FTP server is an externally configured account.
Identity (Email ID)	This field is available only if the **User Authentication** option is set to **anonymous**. Enter the mail ID for the anonymous user.
Service Account	This field is available only if the **User Authentication** option is set to **external user**. Enter the service account for the user. This is a required field when the **User Authentication** option is set to **external user**.
Pass By Reference	Select this check box to stage the file in the archive directory and pass it as a reference in the headers.
Remote Streaming	Select this check box to stream the FTP files directly from the remote server at the time of processing. When you select this option, the archive directory is the remote directory on the remote FTP server computer. Therefore, you should specify the archive directory as relative to the FTP user directory.
File Mask	Enter the regular expression for the files to be picked. The default is *.*.
Managed Server	This field is available only in a clustered domain. Select the Managed Server to act as the polling server. All of the Managed Servers can process the message, but only one can poll for the message.
Polling Interval	Enter a polling interval, in seconds. The default is 60.
Read Limit	Specify the maximum number of messages to read per polling sweep. Enter 0 to specify no limit. The default is 10.
Post Read Action	Select what happens to a message after it has been read. ■ **Archive** - The message is archived. ■ **Delete** - The message is deleted.
Transfer Mode	Select **ASCII** or **binary** as the transfer mode.
Archive Directory	Specify the path to the archive location if the **Post Read Action** option is set to **Archive**. This field is required if the **Pass By Reference** option is selected. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Download Directory	Enter the directory on your local computer where files are downloaded during the file transfer. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Error Directory	Enter the location where messages are posted if there is a problem. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Request Encoding	Accept the default UTF-8 as the character set encoding for requests in FTP transports.

Table 20–6   (Cont.) Create/Edit a Proxy Service - FTP Transport Configuration Page

Option	To create or edit...
Scan Subdirectories	Select this check box to recursively scan all directories
Sort By Arrival	Select this check box to deliver events in the order of arrival.
Timeout	Enter the socket timeout interval, in seconds, before the connection is dropped. If you enter 0, there is no timeout.
Retry Count	Specify the number of retries for FTP connection failures.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.8 HTTP Transport Configuration Page

The HTTP transport now supports both HTTP and HTTPS endpoints.

The Create/Edit a Proxy Service - HTTP Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure transport settings for a proxy service using the HTTP transport protocol. Table 20–7 describes how to use the page.

Table 20–7   Create/Edit a Proxy Service - HTTP Transport Configuration Page

Option	To create or edit...
HTTPS required	Select this check box for inbound HTTPS endpoints.
	To learn more, see "Configuring Transport-Level Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 20–7   (Cont.) Create/Edit a Proxy Service - HTTP Transport Configuration Page

Option	To create or edit...
Authentication	Select one of the following: ■ **None** - Specifies that authentication is not required. **Note**: Select this option if you will specify an Oracle WSM policy for the proxy service. ■ **Basic** - Specifies that basic authentication is required to access this service. Basic authentication instructs WebLogic Server to authenticate the client using a user name and password against the authentication providers configured in the security realm, such as a Lightweight Directory Access Protocol (LDAP) directory service and Windows Active Directory. The client must send its user name and password on the HTTP request header. Basic authentication is strongly discouraged over HTTP because the password is sent in clear text. However, it is safe to send passwords over HTTPS because HTTPS provides an encrypted channel. **Warning**: By default, all users (authorized and anonymous) can access a proxy service. To limit the users who can access a proxy service, create a transport-level authorization policy. See Section 25.15, "Editing Transport-Level Access Policies." ■ **Client Certificate** - Specifies encrypted communication and strong client authentication (two-way SSL). To learn more, see "Configuring Transport-Level Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. ■ **Custom Authentication** - Specifies that an authentication token is contained in an HTTP header. The client's identity is established through the use of this client-supplied token. You must configure an Identity Assertion provider that maps the token to an Oracle Service Bus user. The custom authentication token can be of any active token type supported by a configured WebLogic Server Identity Assertion provider.
Dispatch Policy	Select a dispatch policy for this endpoint. Leave blank to use the default dispatch policy. Dispatch policy refers to the instance of Oracle WebLogic Server Work Manager that you want to use for the service endpoint. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Request Encoding	■ For HTTP inbound transports: If the character set encoding parameter of the `Content-Type` header is not specified in Client Request, enter a character set encoding parameter. If you do not enter a value, the field defaults to `ISO-8859-1`. ■ For HTTP outbound transports: If you have not configured a request encoding, the Oracle Service Bus runtime decides the most appropriate encoding while it makes a request to the business service. In the case of a non-pass-through scenario, the default character encoding is `UTF-8` at runtime. However if it is a pass-through scenario, the runtime will pass through the encoding received with the outbound response.

Table 20–7 (Cont.) Create/Edit a Proxy Service - HTTP Transport Configuration Page

Option	To create or edit...
Response Encoding	Accept the default ISO-8859-1 as the character set encoding for responses in HTTP transports, or enter a different character set encoding.
Authentication Header	Enter the HTTP header (any except Authorization) from which Oracle Service Bus is to extract the token. This field is available only if you selected the **Custom Authentication** check box.
	For example, client-xyz-token.
Authentication Token Type	Select an authentication token type. Only the active token types configured for an Identity Assertion provider are available. (See Configuring Identity Assertion Providers at http://download.oracle.com/docs/cd/E13159_01/osb/docs10gr3/security/message_level_cust_auth.html#wp1066076 for Custom Tokens for more information.) This field is available only if you selected the **Custom Authentication** check box.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.9 JCA Transport Configuration Page

Use this page to configure transport settings using the JCA transport protocol. For more information on using the JCA transport, see "JCA Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus* and Section 20.1.1, "Generating a Proxy Service from a JCA Binding Resource."

Table 20–8 JCA Transport Configuration Page

Option	Description
JCA File	Click **Browse** to select a JCA Binding. The JCA Binding defines different aspects of the service, such as details about the adapter used, a binding to the WSDL and TopLink or EclipseLink mapping file, and the activation/interaction spec properties required by the service.
	Once you select a valid JCA Binding, the remaining transport configuration fields become available.
	For more information on JCA Bindings, see Chapter 13, "JCA Bindings."
Adapter Name	A read-only value showing the name of the adapter that the JCA service will use.
Adapter Type	A read-only value showing the adapter type.
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

Table 20–8   (Cont.) JCA Transport Configuration Page

Option	Description
JNDI Service Account	JNDI Service Account is for JNDI context security, used to access the EIS adapter managed connection factory. Click **Browse** and select a service account. If no service account is specified, an anonymous subject is used.
	For JCA business services, there is no restriction on the type of JNDI service account that can be configured, such as static or pass-through, but the runtime must be able to access a user name and password. JCA proxy services can use only static JNDI service accounts.
	For more information on JNDI service accounts, see "Security" in the "JCA Transport" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
EndPoint Properties	This field lets you assign values to endpoint properties such as retries for the type of adapter the service uses.
	For a list of supported JCA endpoint properties, see "Endpoint Properties" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dynamic EndPoint Properties	This option lets you pass request parameters to JCA-compliant services. For example, you can use a dynamic endpoint property to pass database query parameters to the Oracle JCA Adapter for Database.
	For more information on querying with parameters, see "Oracle JCA Adapter for Database" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.
	Enter a name/value pair for each dynamic endpoint property you want to provide. The endpoint property key matches the query parameter name.
Always use configuration from JCA file	This option determines whether Activation Spec Properties (proxy services) and Interaction Spec Properties (business services) are always used from the JCA file.
	If this option is selected (default), the JCA transport interacts with the JCA framework using the activation/interaction spec properties in the JCA file.
	If this option is not selected, you can override the Activation/Interaction Spec Properties.
	For the redeployment impact of using this option, see "Endpoint Redeployment" in the "JCA Transport" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Operation Name	Displays a read-only name of the selected WSDL operation. An operation can have its own activation/interaction spec properties, shown in the Activation/Interaction Spec Properties field.
Activation/Interaction Spec Properties	**Activation Spec Properties** is the field name for proxy services; **Interaction Spec Properties** is the field name for business services.
	If this service is an inbound service invoked by an EIS application, this field displays the activation spec properties for the JCA inbound operation shown in Operation Name field.
	You can override the activation/interaction spec properties if you clear the **Always use configuration from JCA file** option.
	Note: For Oracle Adapter Suite adapters, activation/interaction spec properties are displayed as read-only. The Oracle Adapter Suite adapters store their own configurations, which you must change in the Oracle Adapter Suite management tools.

Table 20–8 (Cont.) JCA Transport Configuration Page

Option	Description
Connection properties (legacy)	For legacy JCA services that use non-managed mode connection properties (deprecated in this release), see the connection configuration options at `http://download.oracle.com/docs/cd/E13159_01/osb/docs10gr3/jcatransport/transport.html#wp1105451`.

For more information on endpoint and activation/interaction spec properties, see the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

20.2.10 JEJB Transport Configuration Page

Use this page to configure transport settings using the JEJB transport protocol. For more information on using the JEJB transport, see "JEJB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Table 20–9   JEJB Transport Configuration Page

Option	Description
Dispatch Policy	Select the instance of WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
EJB Spec Version	Select the EJB version of the remote EJB interface.
Pass XMLBeans by value	Select this option if you want the transport to generate an "inlined" XML representation of POJO arguments (an XMLObject) whose parameters you can access and manipulate with XQuery expressions. **Note**: Type information is not available inline for XMLObjects passed by value. If you use this option, you cannot pass the typed XMLObject as the argument in a Java Callout in a proxy service pipeline. Do not select this option to pass the POJO by reference, which also results in better performance.
Transaction Attribute	Select one of the following options for handling transactions: **Supports** – The transport accepts an incoming transaction. Quality of service is exactly-once if the operation is invoked in a transaction and best-effort if the operation is invoked outside of a transaction. **Required** – The transport accepts an incoming transaction. If no ongoing transaction exists, the transport starts one. Quality of service is exactly-once. **RequiresNew** – The transport always starts a new transaction, suspending an ongoing transaction. Quality of service is exactly-once. **Mandatory** – The transport invokes the method in the existing transaction. Quality of service is exactly-once. **NotSupported** – The transport suspends an existing transaction and resumes it on invocation. Quality of service is best-effort. **Never** – The transport does not invoke the method in a transaction. Quality of service is best-effort.

Table 20–9   (Cont.) JEJB Transport Configuration Page

Option	Description
Client JAR	Click **Browse** and select an EJB client JAR resource from the list displayed. The client JAR contains the remote or business interface for the remote EJB. The Client JAR is registered as a generic Archive Resource.
Home Interface	**EJB 2.1 only** – Select the required EJBHome interface from the options populated by the client JAR.
Remote Interface	**EJB 2.1 only** – This field is automatically populated based on the configuration of the Home Interface.
Business Interface	**EJB 3.0 only** – Select the business interface from the client JAR that you want to invoke.
Target Namespace	This field is populated by information picked up from the JAR.
Methods	Select the required methods. Click + to expand the method, which lets you edit the default parameter values.
	You can change the default operation name for a given method. By default, the operation name is the method name. If an EJB contains methods with same name (overloaded), you must change the operation names so that they are unique. WSDLs require unique operation names.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.11 JMS Transport Configuration Page

The Create/Edit a Proxy Service - JMS Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "JMS Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a proxy service using the JMS transport protocol. Table 20–10 describes how to use the page.

Table 20–10   Create/Edit a Proxy Service - JMS Transport Configuration Page

Option	To create or edit...
Destination Type	Select one of the following:
	■ **Queue** (for a point-to-point destination type)
	■ **Topic** (for a publish/subscribe destination)
Is Response Required	This option is available only when **Queue** is selected for the **Destination Type**.
	Select this option to specify that a response is expected after an outbound message is sent.

Table 20–10 (Cont.) Create/Edit a Proxy Service - JMS Transport Configuration Page

Option	To create or edit...
Response Pattern	This option is available only when the Is Response Required check box is selected. Select one of the following: ■ Select **JMSMessageID** for JAX-RPC services running on Oracle WebLogic Server. ■ Select **JMSCorrelationID** for all other services. When you select this option, you must also enter a **Response URI**.
Response Message Type	This option is available only when the **Is Response Required** check box is selected. Select one of the following: ■ **Bytes** (for a stream of uninterpreted bytes) ■ **Text** (for text messages) This option is disabled if you select a Message Type of Java for the response.
Client Jar	This option is available when the service is a Messaging Service with a request type of Java. Select the client JAR to be used for dequeueing messages that contain Java Objects. Selecting the client JAR ensures it is on the classpath. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Dispatch Policy	Select the instance of Oracle WebLogic Server Work Manager that you want to use for the dispatch policy for this endpoint. The default Work Manager is used if no other Work Manager exists. For example, if the business service has a JMS transport protocol, the business service endpoint is an MDB (message-driven bean) JAR file that you can associate with the specific dispatch policy. For information about Work Managers, see: ■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server* ■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Request Encoding	Enter the character set for encoding requests. The default is UTF-8.
Response Encoding	This option is available only when the Is Response Required check box is selected. Enter the character set for encoding responses. The default is UTF-8.

Table 20–10   (Cont.) Create/Edit a Proxy Service - JMS Transport Configuration Page

Option	To create or edit...
Response URI	This option is available only when JMSCorrelationID is selected for the Response Correlation Pattern.
	Enter a response URI in the format:
	jms://host:port/connection_factory/jndi_destination
	To target multiple servers, use the following format:
	jms://host1:port,host2:port/connection_factory/jndi_destination
	You can also omit the host and port in the response URI. For example:
	jms:///connection_factory/jndi_destination
	When you omit host and port, the connection factory/destination lookup occurs on the local server. This is useful, for example, if the request URI goes to a foreign connection factory/destination, but you want the response sent to the local server.
	Note: While Oracle WebLogic Server allows forward slashes in JNDI names, such as "myqueues/myqueue", JNDI names with forward slashes interfere with the URI format required by Oracle Service Bus, and you cannot use those names. To work around this issue, define a JMS foreign server and reference that foreign server in the URI. For more information, see "Configure foreign servers" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.
Response Connection Factory	This option is available only when JMSMessageID is selected for the Response Correlation Pattern.
	Enter a response connection factory URI.
	If a connection factory is not specified, the connection factory for the request is used for the response.
JMS Service Account	Select a service account to use for the JMS resource managed by the JMS server. A service account is an alias resource for a user ID and password, used for both the request and response. The same service account is used for both JMS and JNDI purposes.
	For more information, see Chapter 17, "Service Accounts."
Use SSL	Select this option only if the requests are made over a TLS/SSL connection.
	TLS/SSL (Secure Sockets Layer) provides secure connections by allowing two applications connecting over a network to authenticate the other's identity and by encrypting the data exchanged between the applications. Authentication allows a server, and optionally a client, to verify the identity of the application on the other end of a network connection. Additionally, if the administrator has restricted access to individual JMS destinations (queues or topics) by setting access control on the JNDI entry for the destination, the service must authenticate when looking up the entry in the JNDI tree.
	Note: The JMS transport does not support two-way SSL.
Message Selector	Enter a message selector expression.
	Only messages with properties matching the expression are processed

Table 20–10 (Cont.) Create/Edit a Proxy Service - JMS Transport Configuration Page

Option	To create or edit...
JMS Client ID	For topics only, enter the client ID to use for the subscriber. If no value is entered, a client ID is automatically generated. Assigning a custom ID helps you identify this component when monitoring the service. This is only effective for durable subscribers.
	The JMS Client ID is one of several MDB descriptors that are used to generate subscriber names and the ClientID value for the topic's subscribers. Defining a JMS client ID makes it easier to identify and view subscribers for a specific topic by looking at the subscriber name or client ID.
	For more information, see "Specifying the JMS Client ID in Proxy Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Durable Subscription	This option is available only if **Topic** is selected for the **Destination Type**.
	Select this check box if the subscription is durable or leave it blank if the subscription is not durable
Retry Count	Enter the number of delivery retries a message can have before it is moved to the error destination. This field only applies to Oracle WebLogic Server JMS destinations.
Retry Interval	Enter the amount of time, in milliseconds, before rolled back or recovered messages are redelivered. This field only applies to Oracle WebLogic Server JMS destinations.
Error Destination	Enter the name of the target destination for messages that have reached their redelivery limit. This field only applies to Oracle WebLogic Server JMS destinations.
Expiration Policy	Select an Expiration Policy to use when an expired message is encountered on a destination. This field only applies to Oracle WebLogic Server JMS destinations.
Is XA Required	Select this check box if your connection factory is XA.
	This value is taken into account when the remote connection factory is unavailable. If your connection factory is available and this value is true, make sure that the connection factory is defined as transactional.
Transaction Timeout	Enter the amount of time in seconds for the JMS proxy service to wait for a transaction to be processed. This option only applies to services using an XA connection factory. The default is 600 seconds.

Table 20–10   (Cont.) Create/Edit a Proxy Service - JMS Transport Configuration Page

Option	To create or edit...
Topic Messages Distribution	This option is available when you select Topic for the Destination Type.
	Topic Messages Distribution provides different options for handling inbound JMS messages and providing high availability and failover.
	Select one of the following properties to determine how message-driven beans handle incoming JMS messages:
	■ **One Copy Per Application** (default) – If the JMS proxy service is deployed to a cluster, this option provides high availability and scalability by ensuring that an inbound JMS message is processed on only one of the available servers in a cluster.
	■ **One Copy Per Server** – Select this option if you want inbound JMS messages published to a topic to be received by the proxy service on every member of the cluster.
	■ **Compatibility** – Select this option if you want inbound JMS messages to be processed on a specific Managed Server or all Managed Servers in a cluster. If you select this property, select a Target server.
	Note: For topic destinations deployed on Oracle WebLogic Server prior to version 10.3.4, the only valid option is Compatibility.
	The "One Copy" options you select for Topic Messages Distribution override the Subscription Sharing Policy and Client ID Policy configured on the JMS Connection Factory.
Target	This option is available only in an Oracle Service Bus cluster when you select Compatibility for the Topic Messages Distribution option. Select the target server that will handle incoming JMS messages. If you select one of the "One Copy" options for Topic Messages Distribution, this field displays the name of the cluster.
	If you do not set a target, the JMS proxy service instance (reading messages off the Topic) on each Managed Server in the cluster gets a copy of the message.
JNDI Timeout	The JNDI connection timeout (in seconds) used while looking up the destination or connection factory in the JNDI tree.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.12 MQ Transport Configuration Page

The Create/Edit a Proxy Service - MQ Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "MQ Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Before you begin

Configure a MQ Connection resource. See Chapter 9, "MQ Connections."

To configure the MQ transport

Use this page to configure transport settings for a proxy service using the native MQ transport protocol. Table 20–11 describes how to use the page.

Table 20–11   Create/Edit a Proxy Service - MQ Transport Configuration Page

Option	To create or edit...
Is Response Required	Select this option to specify that a response is expected after an outbound message is sent.
Response Correlation Pattern	This option is available only when the **Is Response Required** check box is selected.
	Specify whether the response correlation pattern should be based on **MessageID** or **CorrelationID**.
MQ Response URI	This option is available only when the **Is Response Required** check box is selected.
	The destination to which the response should be published. Enter a response URI in the same format as the endpoint URI: `mq://<local-queue-name>?conn=<mq-connection-resource-ref>`
Response Message Type	This option is available only when the **Is Response Required** check box is selected.
	Select one of the following:
	■ **Bytes** (for a stream of uninterpreted bytes)
	■ **Text** (for text messages)
Transaction Timeout	Enter the amount of time in seconds to wait before timing out an XA transaction initiated by the proxy service. This property only applies to services using an MQ connection resource that is configured for XA transactions (that is, the **XA Enabled** option is selected for the MQ connection resource).
	The default is `300` seconds.
Polling Interval	Enter a polling interval, in milliseconds. The default is `1000`.
Poller Dispatch Policy	The dispatch policy to use for the poller threads.
Backout Threshold	Enter a value representing the number of times the pipeline should retry a message before redirecting the message to the queue specified in the **Dead Letter URI** field.
	If you do not specify a value for this field, the message is redirected to the dead letter queue without attempting any retries.
MQ Dead Letter URI	Enter the URI of the dead letter queue to which request messages should be redirected after attempting the number of retries specified in the **Backout Threshold** field.
	If you do not specify a value for this field, the message is returned to the queue and ignored by the MQ transport for each poll after retrying the number of times specified in the **Backout Threshold** field. The Dead Letter URI uses the same format as the EndPoint URI.

Table 20–11   (Cont.) Create/Edit a Proxy Service - MQ Transport Configuration Page

Option	To create or edit...
Endpoint URI 'GET' options	Enter the MQ GET message options from among the following:
	■ MQC.MQGMO_ACCEPT_TRUNCATED_MSG
	■ MQC.MQGMO_ALL_MSGS_AVAILABLE
	■ MQC.MQGMO_BROWSE_FIRST
	■ MQC.MQGMO_BROWSE_NEXT
	■ MQC.MQGMO_COMPLETE_MSG
	■ MQC.MQGMO_CONVERT
	■ MQC.MQGMO_FAIL_IF_QUIESCING
	■ MQC.MQGMO_LOCK
	■ MQC.MQGMO_LOGICAL_ORDER
	■ MQC.MQGMO_MARK_BROWSE_CO_OP
	■ MQC.MQGMO_MARK_SKIP_BACKOUT
	■ MQC.MQGMO_NO_SYNCPOINT
	■ MQC.MQGMO_NONE
	■ MQC.MQGMO_NO_WAIT
	■ MQC.MQGMO_SYNCPOINT
	■ MQC.MQGMO_SYNCPOINT_IF_PERSISTENT
	■ MQC.MQGMO_UNLOCK
	■ MQC.MQGMO_UNMARK_BROWSE_CO_OP
	■ MQC.MQGMO_UNMARK_BROWSE_HANDLE
	■ MQC.MQGMO_UNMARKED_BROWSE_MSG
	■ MQC.MQGMO_VERSION_1
	■ MQC.MQGMO_VERSION_2
	■ MQC.MQGMO_VERSION_3
	■ MQC.MQGMO_WAIT
	You can use either "|" or "+" to separate multiple options. For example, you can specify the following:
	MQC.MQGMO_ACCEPT_TRUNCATED_MSG | MQC.MQGMO_LOCK
	The MQ GET message options are applied when reading a message from the inbound queue.
Process RFH2 Headers	Select this option to parse WebSphere MQ RFH2 headers from a message payload and automatically generate an RFH2Headers transport header containing the RFH2 data.
	If you do not select this option, the payload is passed through as received.
	For information about how the MQ transport handles RFH2 headers, see "About RFH2 Headers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Ignore Reply To Headers	Select this option to ignore the reply-to headers for the queue manager.

Table 20–11   (Cont.) Create/Edit a Proxy Service - MQ Transport Configuration Page

Option	To create or edit...
Dispatch Policy	Select a dispatch policy for this endpoint.
	Dispatch policy refers to the instance of Oracle WebLogic Server Work Manager that you want to use for the service endpoint.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.13 SB Transport Configuration Page

The Create/Edit a Proxy Service - SB Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "SB Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a proxy service using the SB (Service Bus) transport protocol. Table 20–12 describes how to use the page.

Table 20–12   Create/Edit a Proxy Service - SB Transport Configuration Page

Option	To create or edit...
Dispatch Policy	Select a dispatch policy for this endpoint or use the default dispatch policy.
	Dispatch policy refers to the instance of Oracle WebLogic Server Work Manager that you want to use for the service endpoint to process the request.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Use SSL	When specified, requests must be sent over an SSL connection.
	However, unsecured connections are not forbidden. The administrator must close all unsecured protocols on the server (for example, t3 or http) to strictly enforce secured client connections.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.14 SFTP Transport Configuration Page

The Create/Edit a Proxy Service - SFTP Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "HTTP and Poller Transports" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a proxy service using the sftp transport protocol. Table 20–13 describes how to use this page.

Table 20–13   Create/Edit a Proxy Service - SFTP Transport Configuration Page

Option	To create or edit...
User Authentication	Select one of the following: ■ **Username Password Authentication** - Specifies that a static service account is associated with this authentication method and the client is authenticated using the provided credentials. ■ **Host Based Authentication** - Specifies that a user name and service key provider is required to use this authentication method. Any user connecting from a known host is authenticated using the private key of the host. ■ **Public Key Authentication** - Specifies that a user name and service key provider is required to use this authentication method. Every user has their own private key. **Note**: The Oracle Service Bus service does not use the service key provider to authenticate any credentials from the SFTP server. It uses only the known_hosts file to authenticate the SFTP server, as described in "Configuring Transport-Level Security for SFTP Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Service Account	Enter the service account for the user, or click **Browse** to select service accounts from a browser.
Service Key Provider	This option is available only when **Host Based** or **Public Key Authentication** is selected. Enter a service key provider in the **Service Key Provider** field. You can click **Browse** to select service key providers from a browser. This is a required field.
Username	This option is available only when **Host Based** or **Public Key Authentication** is selected. Enter the user name.
Pass By Reference	Select this check box to stage the file in the archive directory and pass it as a reference in the headers.
Remote Streaming	Select this check box to stream the SFTP files directly from the remote server at the time of processing. When you select this option, the archive directory is the remote directory on the remote SFTP server computer. Therefore, you should specify the archive directory as relative to the SFTP user directory.
File Mask	Enter the regular expression for the files to be picked. The default is *.*.
Managed Server	This field is available only in a clustered domain. Select the Managed Server to act as the polling server. All of the Managed Servers can process the message, but only one can poll for the message.

Table 20–13   (Cont.) Create/Edit a Proxy Service - SFTP Transport Configuration Page

Option	To create or edit...
Polling Interval	Enter the interval in seconds at which the file is polled from the specified location. The default is 60.
Read Limit	Specify the maximum number of messages to read per polling sweep. Enter 0 to specify no limit. The default is 10.
Post Read Action	Select what happens to a message after it has been read. ■ **Archive** - The message is archived. ■ **Delete** - The message is deleted.
Archive Directory	Specify the path to the archive location if the **Post Read Action** option is set to **Archive**. This field is required if the **Pass By Reference** option is selected. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Download Directory	Enter the directory on your local computer where files are downloaded during the file transfer. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Error Directory	Enter the location where messages are posted if there is a problem. **Note**: The Archive, Download, and Error directories are absolute paths, and they are automatically created. If you specify a relative path, the files are created relative to the Java process that starts the WebLogic Server.
Request Encoding	Accept the default UTF-8 as the character set encoding for requests in SFTP transports.
Preferred Cipher Suite	Select the cipher suite to use when communicating with the server. If you use the default value, **Use Runtime Default**, the list of supported cipher suites is sent to the server and each is tried until a match is found.
Preferred Data Integrity Algorithm	Select the bulk-hashing algorithm for data integrity checks. If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, hmac-sha1. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Public Key Algorithm	Select the asymmetric key algorithm for public-key cryptography. If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, ssh-dss. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Key Exchange Algorithm	Select the default key exchange protocol for negotiating the session key for encrypting the message. If you use the default value, **Use Runtime Default**, Oracle Service Bus sends the preferred algorithm, diffie-hellman-group1-sha1. If that is not supported by the server, the list of supported algorithms is sent to the server and each is tried until a match is found.
Preferred Compression Algorithm	Select whether to compress in-flight data using zlib. Select **zlib** to compress data; otherwise select **None**. The default is **None**.
Scan Subdirectories	Select this check box to recursively scan all directories

Table 20–13   (Cont.) Create/Edit a Proxy Service - SFTP Transport Configuration Page

Option	To create or edit...
Sort By Arrival	Select this check box to deliver events in the order of arrival.
Timeout	Enter the socket timeout interval, in seconds, before the connection is dropped. If you enter `0`, there is no timeout. The default value is 60.
Retry Count	Specify the number of retries for SFTP connection failures.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.15 Tuxedo Transport Configuration Page

The Create/Edit a Proxy Service - Tuxedo Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "Tuxedo Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a proxy service using the Tuxedo transport protocol. Table 20–14 describes how to use the page.

Table 20–14   Create/Edit a Proxy Service - Tuxedo Transport Configuration Page

Option	To create or edit...
Field Table Classes	Enter the name of the class or classes describing the `FML/FML32` buffer received. These are used for the `FML/FML32-to-XML` conversion routines to map field names to element names. This is a space separated list of fully qualified class names.
View Classes	Enter the name of the class or classes describing the `VIEW/VIEW32` buffer received or sent. These are used for the `VIEW-to-XML` or `VIEW32-to-XML` conversion routines to map field names to element names. This is a space separated list of fully qualified class names.
	`X_C_TYPE` and `X_COMMON` Tuxedo buffer types are handled in the same manner as `VIEW/VIEW32` buffers.
	If an incoming request contains a `VIEW`, then the corresponding `VIEW` class should be specified in the Oracle Service Bus `CLASSPATH`.
Classes Jar	Select a JAR resource that contains a JAR file with the `FML/FML32` or `VIEW/VIEW32` classes necessary for this endpoint operation.
Local Access Point	Select a local access point from the list that is associated with the export. The list contains local access points configured in WTC. A proxy service cannot be created if there is not an associated local access point.
	If no local access points exist or to create a new one, select **New**. Enter the corresponding **Local Access Point Name** and **Local Network Address** in the adjacent fields. Upon validation of the endpoint, the access point is added to the WTC configuration for each WTC server. If no WTC server exists, one is created.
	You can enter an existing access point name after selecting the **New** option. This causes the existing information to be updated with the new parameters. You can change only the host name and port number.

Table 20–14   (Cont.) Create/Edit a Proxy Service - Tuxedo Transport Configuration Page

Option	To create or edit...
Remote Access Point	This field appears only when you select **New** in the **Local Access Point** field.
	From the list, select a remote access point to be associated with the newly created local access point. If none exist or to create a new one, select **New**. Enter the corresponding **Access Point Name** and **Network Address** in the adjacent fields.
	You can enter an existing access point name after selecting the **New** option. This causes the existing information to be updated with the new parameters. You can change only the host name and port number.
	The remote access point will also be the authentication principal for the WTC connection for inbound requests. Optionally, you can create a user with the same access point ID in the default security realm to allow incoming calls. To do so, select Yes from the Create User? list. The password will be randomly generated using a temporary variable to avoid security issues.
Reply Buffer Type	This option is available only if the **Response Required** option is selected.
	Select the type of buffer that the remote Tuxedo client will receive.
Reply Buffer Subtype	This option is available only when the **Response Required** option is selected and the **Reply Buffer Type** value is **VIEW** or **VIEW32**.
	Enter the buffer subtype with which to associate the reply buffer.
Response Required?	Select this check box if this service is expected to send a response.
	The default status is that this option is selected.
	This option is cleared and the unavailable if the service type is **Messaging Service** and the response message type is **None**.
Request Encoding	Specify a character set encoding for requests in Tuxedo transports.
Response Encoding	Specify a character set encoding for responses in Tuxedo transports.
Transformation Style	Select one of the following:
	■ **None** - (default) The order of fields may not be respected.
	■ **Ordered** - The fields are presented with all their occurrences in the correct order.
	■ **Ordered and Grouped** - If the fields are logically structured as records, the fields are ordered by occurrence and grouped by record.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.16 WS Transport Configuration Page

The Create/Edit a Proxy Service - WS Transport Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

For more information, see "WS Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Use this page to configure transport settings for a proxy service using the WS transport protocol. Table 20–15 describes how to use the page.

Table 20–15   Create/Edit a Proxy Service - WS Transport Configuration Page

Option	To create or edit...
Dispatch Policy	Select a dispatch policy for this endpoint or use the default dispatch policy.
	Dispatch policy refers to the instance of Oracle WebLogic Server Work Manager that you want to use for the service endpoint.
	For information about Work Managers, see:
	■ "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*
	■ Section 37.13, "Using Work Managers with Oracle Service Bus"
Retry Count	The number of times to retry delivery of a message to the pipeline.
	If an unhandled exception occurs in the request pipeline of a proxy service, the incoming WS transport message will be redelivered to the pipeline up to the number of times specified by the retry count. This value is important for reliably processing WS transport messages.
Retry Delay	The number of seconds the system pauses before retrying to send a message to the pipeline after an error.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.17 Operation Selection Configuration Page

The Create/Edit a Proxy Service - Operation Selection Configuration page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations." This page is displayed only if the service you are creating has operations.

Use this page to enforce WS-I compliance (for SOAP 1.1 services only) and select the selection algorithm to use to determine the operation called by this proxy service. This option is only available for SOAP or XML services defined from a WSDL.

The WSDL specification defines a default algorithm to compute which operation is called based on the type of the SOAP message received. However, there are cases (for example, performance issues, signature/encryption issues, or the default algorithm is not applicable) when you may need to select the operation based on other means.

Oracle Service Bus provides additional algorithms. Each of them follows the same pattern and are based on the evaluation of an expression to get a value that is then used to lookup the corresponding operation in a static table.

Oracle Service Bus is generally very forgiving if an inbound message is either missing data such that the operation cannot be determined, or has data that does not correspond to a valid operation. Both of these conditions result in $operation being empty. Rather than reject all such messages, Oracle Service Bus does not initialize the operation variable in the context but otherwise continues to process the message.

However, security requirements are enforced if the proxy service is WSDL-based and at least one of the following conditions is true:

- The WSDL has a WS-Security policy and the proxy is an active intermediary.

- The proxy has message-level custom authentication (either custom token or username/password).

If these conditions are met, then there is a check at runtime to make sure the operation selection algorithm returns a valid operation name. If the operation selection returns null or an operation that is not in the WSDL, then the message is rejected and an error is raised.

Table 20–16 describes how to use the Create/Edit a Proxy Service - Operation Selection Configuration page.

Table 20–16   Create/Edit a Proxy Service - Operation Selection Configuration Page

Option	To create or edit...
Enforce WS-I Compliance	For SOAP 1.1 services only:
	Select this check box to specify whether the service is to conform to the Basic Profile defined by the Web Services Interoperability Organization.
	When a service is marked WS-I compliant, checks are performed against the messages sent to and from that service. For proxies, checks are performed against request messages received by the proxy. For invoked services (such as services invoked by a proxy through a service callout action or route node), checks are performed against the response messages received from those services. It is the WS-I compliance property of the invoked service and not the proxy that determines whether checks are performed against messages received from the invoked service. If you specify WS-I compliance testing for an invoked service, the message flow generates a fault for response errors.

Table 20–16   (Cont.) Create/Edit a Proxy Service - Operation Selection Configuration

Option	To create or edit...
Selection Algorithm	Select one of the following and perform any required additional steps: **Transport Header** - Select this algorithm to define the transport header that contains the lookup value. Then: In the **Header Name** field, enter the transport header that extracts the value used as a key to select the operation being invoked. Under the **Operation Mapping** field, specify the value for each operation in the **Value** field. The value is used as the key of the operation. **SOAPAction Header** - Select this algorithm to specify that operation mapping be done automatically from the WSDL associated with this proxy service. **WS-Addressing** - Select this algorithm to specify that the lookup value is contained by the WS-Addressing `Action` tag located in the SOAP headers of the SOAP message. Then, under the **Operation Mapping** field, specify the value for each operation in the **Value** field. The value is used as the key of the operation. **SOAP Header** - Select this algorithm to define an XPath expression to be evaluated against the SOAP headers. This lets you get the lookup value. Then: In the **XPath Expression** field, specify the XPath expression that extracts the value used as a key to select the operation being invoked. Under the **Operation Mapping** field, specify the value for each operation in the **Value** field. The value is used as the key of the operation. **SOAP Body Type** - This is the default algorithm defined by the WSDL specification to compute which operation is called based on the type of the SOAP message received. If the proxy service is configured for a Web Service security pass-through scenario with an encrypted body, you cannot select this algorithm. A similar caveat applies to pass-through encrypted SOAP headers. If you have a WSDL that has two operations with the same input message, do not select this algorithm for operations, because the operation cannot be uniquely identified by inspecting the input message. **Payload Type** - Available only for XML services based on a WSDL port or WSDL binding.
Header Name	This option is available only when the **Selection Algorithm** option is set to **Transport Header**. Enter the transport header that extracts the value used as a key to select the operation being invoked.
XPath Expression	This option is available only when the **Selection Algorithm** option is set to **SOAPHeader**. Specify the XPath expression that extracts the value used as a key to select the operation being invoked.
Operation Mapping	This option is available only when the **Selection Algorithm** option is set to **Transport Header**, **WS-Addressing**, or **SOAP Body Type**. Specify the value for each operation in the **Value** field. The value is used as the key of the operation.

After you finish

Click **Next** to continue configuring this service on the next page; or click **Last** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.18 Message Handling Page

The Create/Edit a Proxy Service - Message Handling page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to specify whether the proxy service should stream message content, how the service is to decode request messages received and encode response messages sent, and whether the proxy service should stream MIME attachments instead of buffering the attachment contents in memory.

Using this page, you can enable the proxy service to stream message content rather than storing it in memory. You can also enable the proxy service to decode and parse inbound messages in MTOM/XOP format and to send responses using the MTOM/XOP format, when appropriate. SOAP Message Transmission Optimization Mechanism (MTOM) is a method of sending binary data to and from Web services. MTOM uses XML-binary Optimized Packaging (XOP) to transfer the binary data.

Using this page, you can also enable the proxy service to store MIME attachment content to a disk file and then process the data in a streaming fashion without buffering the attachment contents in memory. This enables the proxy service to process large attachments robustly and efficiently.

Table 20–17 describes how to use the Create/Edit a Proxy Service - Message Handling page.

Table 20–17   Create/Edit a Proxy Service - Message Handling Page

Option	To create or edit...
Transaction Required	Select this option to ensure Oracle Service Bus executes the proxy service message flow in the context of a transaction. If a global transaction already exists, the transport provider propagates it in the request (even if you do not select this option). If no global transaction exists, the message flow runtime starts a transaction. If the message flow runtime starts a transaction, the transaction context begins before the service configuration is validated or run (for example, security checking or WS-I validation), and before message flow execution, ensuring that all processing and execution occurs in the transaction context.
	If the message flow runtime starts a transaction, quality of service is exactly-once. However, if Service Callouts or Publish actions have the outbound quality of service parameter set to best-effort (the default), Oracle Service Bus executes those actions outside of the transaction context. To have Oracle Service Bus execute those actions in the same request transaction context, set quality of service on those actions to exactly-once.
	The service maintains its messaging pattern (synchronous, asynchronous, one-way) regardless of the setting on this option.
	For transaction timeouts, the global transaction timeout value configured in the Oracle WebLogic Server Console applies.
	Exceptions in Transactions
	Oracle Service Bus invokes the system error handler for failed transactions. You cannot catch failed transaction exceptions in a user-configured error handler. For synchronous patterns, a transaction exception is returned through the response. For asynchronous patterns, where the transaction is designed to be committed in the request, the exception is sent back on the request thread.
	In asynchronous patterns, an error in the response that occurs after transaction committal in the request does not affect the transaction.
Same Transaction for Response	This option applies only to one-way and asynchronous messaging patterns.
	If you select this option, Oracle Service Bus propagates the transaction context from the request thread to the response thread.
	If you select this option, the message pattern becomes synchronous automatically, regardless of the initial message pattern setting (such as asynchronous or one-way).
	You would not use this option, for example, if the business service in the request required a transaction committal before sending the response, such as in a one-way pattern.
	For transaction timeouts, the global transaction timeout value configured in the Oracle WebLogic Server Console applies.
Content Streaming	Select this option to stream message content rather than store it in memory.
	Select the **Enabled** check box and choose the following:
	■ Whether to buffer the intermediate content in memory (**Memory Buffer**) or to a disk file (**Disk Buffer**)
	■ Whether to enable **Compression**
	For more information, see "Streaming body Content" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Table 20–17   (Cont.) Create/Edit a Proxy Service - Message Handling Page

Option	To create or edit...
XOP/MTOM Support	Oracle Service Bus supports XOP/MTOM using the following transports: ■ HTTP/S ■ Local ■ SB Select the **Enabled** check box to enable the proxy service to decode and parse inbound messages in MTOM/XOP format and to send responses using the MTOM/XOP format, when appropriate. This option is disabled for imported proxy services that are based on previous release configurations. If XOP/MTOM Support is enabled, select how to handle binary data in the $header and $body message context variables from among the following options: ■ **Include Binary Data by Reference**: (Default) In an inbound request message, replace xop:Include elements with ctx:binary-content elements when setting up the $header and $body message context variables. ■ **Include Binary Data by Value**: In an inbound request message, replace xop:Include elements with Base64-encoded text versions of corresponding binary data when setting up the $header and $body message context variables. Use Include Binary Data by Reference when you need direct access to binary data, for example to pass data to a Java callout or Message Format Language (MFL) transform. Use Include Binary Data by Value in the following cases: ■ To bridge between MTOM and non-MTOM services. For example, consider an MTOM-enabled proxy service that receives a request that is then routed to a non-MTOM-enabled service. You could use this option to comply with existing standards for sending binary data in XML in Base64-encoded form. ■ To validate the contents of the message against an XML schema that requires a base64binary element to be used in place of binary data If XOP/MTOM Support is enabled for a proxy service, it is not required that every inbound message be in the MTOM format. Instead, this setting specifies that when an MTOM-formatted message arrives, the proxy service should handle it accordingly. Note also that when proxy services not enabled for XOP/MTOM Support receive an MTOM-formatted message, the service rejects the message and issues a runtime error.
Attachments	Oracle Service Bus supports streaming MIME attachments using the following transports: ■ HTTP/S ■ Local (when chained through an HTTP proxy with streaming attachments enabled) Select the **Page Attachments to Disk** check box to enable the proxy service to stream MIME attachments. When enabled for HTTP proxy services, the option applies to proxy service inbound request messages. If you select XOP/MTOM Support, the Attachments option is only available if you choose the **Include Binary Data by Reference** option under XOP/MTOM Support. Note also that payloads that contain attachments must conform to RFC 822. Specifically, lines containing Internet headers need to be terminated with CRLF (carriage return line feed).

After you finish

Click **Next** to review and save this configuration on the Create/Edit a Proxy Service-Summary page.

20.2.19 Summary Page

The Create/Edit a Proxy Service - Summary page is one in a series of pages for creating and editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to view or modify the configuration settings for a proxy service before saving it.

To view or modify settings, click **Edit** in the row of the appropriate category (for example, General Configuration, Transport Configuration, and so on). The pages you can edit depend on what pages you configured when creating the proxy service. The following list shows all pages:

- Section 20.2.1, "General Configuration Page"
- Section 20.2.2, "Message Type Configuration Page"
- Section 20.2.3, "Transport Configuration Page"
- Section 20.2.4, "Protocol-Specific Transport Configuration Pages"
- Section 20.2.5, "Email Transport Configuration Page"
- Section 20.2.6, "File Transport Configuration Page"
- Section 20.2.7, "FTP Transport Configuration Page"
- Section 20.2.8, "HTTP Transport Configuration Page"
- Section 20.2.9, "JCA Transport Configuration Page"
- Section 20.2.10, "JEJB Transport Configuration Page"
- Section 20.2.11, "JMS Transport Configuration Page"
- Section 20.2.12, "MQ Transport Configuration Page"
- Section 20.2.13, "SB Transport Configuration Page"
- Section 20.2.14, "SFTP Transport Configuration Page"
- Section 20.2.15, "Tuxedo Transport Configuration Page"
- Section 20.2.16, "WS Transport Configuration Page"
- Section 20.2.17, "Operation Selection Configuration Page"
- Section 20.2.18, "Message Handling Page"

20.3 Exporting a WSDL Associated with a Proxy Service

You can export the WSDL of a WSDL-based proxy service, so you can view or modify the WSDL in an external tool such as an IDE. The WSDL is exported as a JAR file. This is different than the Export Resources functionality in the System Administration module, which you use to move and stage resources between two domains. See Section 29.2, "Exporting Resources."

Before you begin

You can only export a WSDL when you are outside a session. See Section 3.1, "Using the Change Center."

To export a WSDL

1. Do either of the following:

   - Select **Resource Browser** > **Proxy Services** to display the Summary of Proxy Services page.

   - Select **Project Explorer**, then select the project or folder containing the proxy service you want to export as a WSDL. The Project/Folder View page is displayed.

2. Click the **Export WSDL** icon in the row of the proxy service whose WSDL you want to export. A dialog box prompts you to open or save the exported JAR file.

3. In the dialog box, click **Open** to open the file, or click **Save** to save it.

20.4 Locating Proxy Services

To locate proxy services:

1. Do either of the following:

   - Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the proxy service.

   - Select **Resource Browser** > **Proxy Services**. The Summary of Proxy Services page displays the information shown in Table 20–18. For a more detailed description of the properties, see Section 20.5, "Editing Proxy Service Configurations."

2. To restrict the number of items in the list, you can filter by name, path, or both. In the **Name** and **Path** fields, under **Search**, enter the target's name, path, or both, and then click the **Search** button.

   The path is the project name and the name of the folder in which the proxy service resides.

   Wildcard characters * and ? are allowed. Search is case-sensitive.

   Click **View All** to display all proxy services in the domain. This clears the search parameters from the previous search.

Table 20–18   Summary of Proxy Services Page

Property	Description
Name	A unique name for the proxy service. Click the name to display the View a Proxy Service - Configuration Details page.
	See Section 20.5, "Editing Proxy Service Configurations."
Path	The path is the project name and the name of the folder in which the proxy service resides, for example, UDDI/Proxies/OSB_services.
	Click the path of a proxy service to display the Project/Folder View page that contains it.

Table 20–18 (Cont.) Summary of Proxy Services Page

Property	Description
Actions	Do any of the following:
	■ Click the **Edit Message Flow** icon to edit the message flow of the proxy service. See Section 22.1, "Viewing and Editing Message Flows."
	■ Click the **Launch Test Console** icon to invoke the Test Console, which you use to validate and test the design of your services and transformations. For business services, you can only use the Test Console at runtime; that is, when the session is activated. For transformations, you can use the Test Console whether you are inside or outside a session. See Section 33.1, "Testing Services" and Section 33.2, "Testing Transformations."
	■ The Export WSDL icon is displayed for WSDL-based business services. Click this icon to export a WSDL, which you can then view or modify in an external tools such as an IDE.
	This is different than the Export Resources functionality in the System Administration module, which you use to move and stage resources between two domains. See Section 4.28, "Exporting a WSDL."
Options	Click the **Delete** icon to delete the service. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. For more information, see Section 20.6, "Deleting Proxy Services."

20.5 Editing Proxy Service Configurations

This section describes how to edit proxy service configurations.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the proxy service you want to view or edit. See Section 20.4, "Locating Proxy Services."

3. Click the proxy service name. The View a Proxy Service - Configuration Details page displays configuration information for the selected proxy service.

4. To view or modify settings, do either of the following:

   ■ Click the **Edit** icon next to the name of the category whose properties you want to view or edit (for example, General Configuration, Transport Configuration, and so on). The pages you can edit depend on what pages you configured when creating the proxy service.

   For a list of all those pages, see Section 20.5.1, "View a Proxy Service - Configuration Details Page."

   ■ Click **Edit** at the bottom of the page to display the Create/Edit a Proxy Service-General Configuration page, which is the first page in the sequence of pages for configuring this proxy service.

5. Continue to view or edit, as described in Section 20.1, "Creating and Configuring Proxy Services."

6. On the Create/Edit a Proxy Service-Summary page, click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

20.5.1 View a Proxy Service - Configuration Details Page

The View Proxy Service - Configuration Details page displays the configuration details of a proxy service. Table 20–19 describes all the properties that can appear on this page. (Properties vary depending on the details of the proxy service.)

The categories listed on this page correspond to the Create/Edit a Proxy Service pages used for creating and editing proxy service configurations, as described in Section 20.2, "Create/Edit a Proxy Service - Page Reference."

Click the **Edit** link next to any category name to display the associated configuration page.

Table 20–19   View a Proxy Service: Configuration Details Page

Properties	Description
Last Modified By	The user who created this proxy service or imported it into the configuration.
Last Modified On	The date and time that the user created this proxy service or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this proxy service references. If such references exist, click the numeric link to view a list of the objects.
Referenced by	The number of objects that reference this proxy service. If such references exist, click the numeric link to view a list of the objects.
Description	A description of this proxy service, if one exists.
General Configuration	Shows properties configured on the Section 20.2.1, "General Configuration Page."
Message Type Configuration	Shows properties configured on the Section 20.2.2, "Message Type Configuration Page."
Transport Configuration	Shows properties configured on the Section 20.2.3, "Transport Configuration Page."
Email Transport Configuration	Shows properties configured on the Section 20.2.5, "Email Transport Configuration Page."
File Transport Configuration	Shows properties configured on the Section 20.2.6, "File Transport Configuration Page."
FTP Transport Configuration	Shows properties configured on the Section 20.2.7, "FTP Transport Configuration Page."
HTTP Transport Configuration	Shows properties configured on the Section 20.2.8, "HTTP Transport Configuration Page."
JCA Transport Configuration	Shows properties configured on the Section 20.2.9, "JCA Transport Configuration Page."
JMS Transport Configuration	Shows properties configured on the Section 20.2.11, "JMS Transport Configuration Page."
MQ Transport Configuration	Shows properties configured on the Section 20.2.12, "MQ Transport Configuration Page."
SB Transport Configuration	Shows properties configured on the Section 20.2.13, "SB Transport Configuration Page."

Table 20–19   (Cont.) View a Proxy Service: Configuration Details Page

Properties	Description
SFTP Transport Configuration	Shows properties configured on the Section 20.2.14, "SFTP Transport Configuration Page."
Tuxedo Transport Configuration	Shows properties configured on the Section 20.2.15, "Tuxedo Transport Configuration Page."
WS Transport Configuration	Shows properties configured on the Section 20.2.16, "WS Transport Configuration Page."
Message Level Security Configuration	Shows properties configured on the Section 20.5.3, "Proxy Service Security Page."
Operation Selection Configuration	Shows properties configured on the Section 20.2.17, "Operation Selection Configuration Page."
Message Handling Configuration	Shows properties configured on the Section 20.2.18, "Message Handling Page."

20.5.2 Proxy Service Policy Page

The View a Proxy Service - Policies page is one in a series of pages for editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure policy settings for a proxy service with a service type of WSDL-based XML Web Service, Messaging Service, Any SOAP Service, or Any XML Service. Table 20–20 describes how to use the page. In order for Oracle WSM policies to be used with WSDL-based XML Web Service, Messaging Service, or Any XML Service proxy services, the protocol must be set to HTTP.

For WSDL-based services, WLS 9.2 policies bound to the service are exposed (inlined) in the effective WSDL. Abstract policies are pre-processed before they are inlined. Oracle WSM policies are bound by reference, not inlined in the effective WSDL.

Table 20–20   Proxy Service - Policy Page

Option	To edit...
Service Policy Configuration	Following are the options in the Service Policy Configuration field:
	▪ **From OWSM Policy Store** – Policies are defined in the central Oracle Web Services Manager Policy Store managed by Oracle Enterprise Manager. These policies support WS-Security 1.0/1.1, SAML 1.1/2.0, KerberosToken Profile, and other industry standards.
	With this option, you can add polices from a predefined list. Click **Add** under Service Level Policies to display the list. Use filtering if necessary to locate the policies you want. For proxy services, only service policies are displayed.
	For more information, see "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
	▪ **From WSDL** – Select this option if the service policy is associated with the WSDL upon which the service is based. These policies support WS-Security 1.0, SAML 1.1, and other industry standards.
	With this option you can view (read-only) request and response policies from the WSDL.
	▪ **From Pre-defined Policy or WS-Policy Resource** - Select this option to add service-level policies, operation-level policies (in which case the policy applies to both the request and response messages), request policies, and response policies from the Administration Console.
	Policies are either pre-defined in Oracle WebLogic Server or user-defined in Oracle Service Bus with a WS-Policy resource. These policies support WS-Security 1.0, SAML 1.1, and other industry standards.
	With this option you can add policies at the service, operation, request, and response levels. Click **Add** under the appropriate section to select policies from a list.
	Note: The policy binding models are mutually exclusive. You must use only one type of policy in a service. If you bind policies directly to the service, all WSDL-based policies are ignored.

After you finish

Click **Update** to save this configuration; or click **Reset** to undo your changes.

20.5.3 Proxy Service Security Page

The View a Proxy Service - Security page is one in a series of pages for editing proxy services, as described in Section 20.1, "Creating and Configuring Proxy Services" and Section 20.5, "Editing Proxy Service Configurations."

Use this page to configure security settings for a proxy service.

The fields available on this page depend on the use of policies in the proxy service. For example, if the proxy service use Oracle WSM policies (recommended), Policy Overrides appear on the Security page. For more information, see "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

For WLS 9.2 policies, the configuration for both custom user name/password and custom token is similar. In both cases, you specify XPath expressions that enable

Oracle Service Bus to locate the necessary information. The root of these XPath expressions is as follows:

- Use `soap-env:Envelope/soap-env:Header` if the service binding is AnySOAP or WSDL-SOAP.

- Use `soap-env:Body` if the service binding is not SOAP based.

All XPath expressions must be in a valid XPath format. The XPath expressions must use the XPath "declare namespace" syntax to declare any namespaces used, as follows:

```
declare namespace
ns='http://webservices.example.com/MyExampleService';)
```

Table 20–21 describes the View a Proxy Service - Security Configuration page.

Table 20–21   View a Proxy Service - Security Configuration Page

Option	To edit...
Service Key Provider	If needed, enter the path (project/folder) and name of a service key provider, or click **Browse** to select one from the Select Service Key Provider page.
	For more information, see Chapter 18, "Service Key Providers." To learn how to create a service key provider, see Section 18.2, "Adding Service Key Providers."
Policy Overrides	For Oracle WSM policies, provide any desired overrides that are allowed.
	For more information, see Table 20–20 and "Securing Oracle Service Bus with Oracle Web Services Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Process WS-Security Header	If a client request includes a WS-Security security header, decide whether to process this header:
	■ **Select Yes**: In an active intermediary scenario, the client applies WS-Security to the request and response messages. The proxy service processes the security header and enforces the WS-Security policy.
	This option is automatically set to Yes when a proxy service is based on a WSDL that has a security policy or when you attach a policy to the service on the Service Policy Configuration page.
	■ **Select No**: In a WS-Security pass-through scenario, the client applies WS-Security to the request and response messages. The proxy service does not process the security header. Instead, it passes the secured request message untouched to a business service.
Transport Access Control	For all proxy services, you can create a transport-level policy, which applies a security check when a client attempts to establish a connection with the proxy service. Only requests from users who are listed in the transport-level policy are allowed to proceed.

Table 20–21   (Cont.) View a Proxy Service - Security Configuration Page

Option	To edit...
Message Access Control	A message-level access control policy applies a security check when a client attempts to invoke a proxy service with message-level security. You can create a message-level access control policy in the following cases:
	■ For proxy services that are active Web Service security intermediaries; that is, you process the WS-Security header.
	■ For proxy services for which you set custom authentication on this page.
	Only users who are listed in the message-level policy are allowed to invoke the operation.
Authentication Type	Select one of the following:
	■ Select **None** if the proxy service will not use custom authentication.
	■ **Custom User Name and Password**
	When you select this option, you must also enter values in the **User Name XPath** field and the **User Password XPath** field.
	■ **Custom Token**
	When you select this option, you must also select a **Token Type** and enter values for the **Token XPath** field.
User Name XPath	This option is available only when the **Custom Authentication Settings** option is set to **Custom User Name and Password**. When available, this option is required.
	Enter the user name as an XPath expression.
	The XPath expression is evaluated against the message headers or payload, as appropriate, which allows Oracle Service Bus to obtain the user name and for custom authentication.
User Password XPath	This option is available only when the **Custom Authentication Settings** option is set to **Custom User Name and Password**. When available, this option is required.
	Enter the password as an XPath expression.
	The XPath expression is evaluated against the message headers or payload, as appropriate, which allows Oracle Service Bus to obtain the password values for custom authentication.
Token Type	This option is available only when the **Custom Authentication Settings** option is set to **Custom Token**. When available, this option is required.
	Select the token type from the list. Only the active token types configured for a WebLogic Server Identity Assertion provider are available. See "Configuring Identity Assertion Providers for Custom Tokens" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
Token XPath	This option is available only when the **Custom Authentication Settings** option is set to **Custom Token**. When available, this option is required.
	Enter an XPath expression to specify a path to the custom token. Oracle Service Bus evaluates the Token XPath expression against the message headers or payload, as appropriate, to obtain the token for custom authentication.

Table 20–21   (Cont.) View a Proxy Service - Security Configuration Page

Option	To edit...
Context Properties	Optionally, specify one or more context properties to pass additional context information to the Authentication (Custom User Name and Password) or Identity Assertion (Custom Token) security provider.
	Context Properties provide a way (the `ContextHandler` interface) to pass additional information to the WebLogic Security Framework so that a security provider can obtain contextual information. See "Context Properties Are Passed to Security Providers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
	Enter the **Property Name** as a literal string, and the **Value Selector** as a valid XPath expression. (XPath expressions can also be literal strings.)
	The XPath expressions are evaluated against the same message-part that is used for the custom token or custom user name/password. That is, the **Value Selector** XPath expressions for SOAP-based proxy services evaluate against the header and against the payload for non-SOAP-based proxy services.
	The XPath expression is evaluated at runtime to produce the property's value. A ContextHandler is essentially a name/value list and, as such, it requires that a security provider know what names to look for. Therefore, the XPath expressions are evaluated only if a security provider asks for the value of one of these user-defined properties.
	Click **Add Property** to add this context property. You can add multiple context properties.

After you finish

Click **Update** to save this configuration; or click **Reset** to undo your changes.

20.6 Deleting Proxy Services

Deleting a proxy service deletes all of the ACLs referenced by the proxy from the repository controlled by Oracle Service Bus, as well as from the appropriate authorization provider.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See

2. Select **Resource Browser** > **Proxy Services** to display the Summary of Proxy Services page.

3. Click the **Delete** icon in the row of the of the proxy service you want to delete. The proxy service is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

21

Proxy Services: Actions

This chapter describes how to add different types of actions to message flows using the Oracle Service Bus Administration Console, such as route, publish, service callout, transport headers, conditional actions, error actions, and message transformation actions.

Actions are the elements of pipeline stages, error handler stages, route nodes, and branch nodes that define how messages are to be defined as they flow through a pipeline.

This chapter includes the following sections:

- Section 21.23, "Adding Replace Actions"

- Section 21.24, "Adding Validate Actions"

- Section 21.25, "Adding Alert Actions"

- Section 21.26, "Adding Log Actions"

- Section 21.27, "Adding Report Actions"

21.1 Adding and Editing Actions in Message Flows

Actions are the elements of pipeline stages, error handler stages, route nodes, and branch nodes that define how messages are to be defined as they flow through a proxy service.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

They also assume you have already added a pipeline stage, a route node, and optionally an error handler stage. See:

- Section 22.4, "Adding Pipeline Pair Nodes to Message Flows"

- Section 22.7, "Adding Stages to Pipelines"

- Section 24.2, "Adding Proxy Service Error Handlers"

To add an action to a message flow

1. Select the component to which you want to add an action. For example, click the **Stage** icon, then click **Edit Stage**, or click the **Route Node** icon, then click **Edit Route**.

2. Depending on whether actions have already been added to the stage or to the route node, do one of the following:

   - If no actions have yet been added, the Edit Stage Configuration page displays only the Add an Action icon. Click that icon, then select an action type.

   - If one or more actions have already been added, the Edit Stage Configuration page displays one or more icons representing those actions, for example, a Publish icon or a Routing icon, and so on. Click the appropriate icon, click **Add an Action**, then select an action type.

   - Some actions, such as request and response actions in publish actions, include an **Add an Action** link where an action is appropriate. Click that icon, then select an action type.

   There are no restrictions on what actions may be chained together in a message flow.

   Table 21–1 through Table 21–4 list the actions you can configure for message flows.

Table 21–1   Message Flow - Communication Actions

Action	Description	More Information
  Dynamic Publish	Publish a message to a service identified by an Xquery expression	Section 21.4, "Adding Dynamic Publish Actions"

Table 21–1   (Cont.) Message Flow - Communication Actions

Action	Description	More Information
Publish	Publish a message to a statically specified service.	Section 21.2, "Adding Publish Actions"
Publish Table	Publish a message to zero or more statically specified services. Switch-style condition logic is used to determine at runtime which services will be used for the publish.	Section 21.3, "Adding Publish Table Actions"
Routing Options	Modify any or all of the following properties in the outbound request: URI, Quality of Service, Mode, Retry parameters, Message Priority.	Section 21.5, "Adding Routing Options Actions"
Service Callout	Configure a synchronous (blocking) callout to an Oracle Service Bus-registered proxy or business service.	Section 21.6, "Adding Service Callout Actions"
Transport Headers	Set the transport header values in messages	Section 21.7, "Adding Transport Header Actions"
Dynamic Routing	Assign a route for a message based on routing information available in an XQuery resource.	Section 21.8, "Adding Dynamic Routing to Route Nodes"
Routing	Identify a target service for the message and configure how the message is routed to that service:	Section 21.9, "Adding Routing Actions to Route Nodes"
Routing Table	Assign a set of routes wrapped in a switch-style condition table.Different routes are selected based upon the results of a single XQuery expression.	Section 21.10, "Adding Routing Tables to Route Nodes"

Table 21–2   Message Flow - Flow Control Actions

Action	Description	More Information
For each	Iterate over a sequence of values and execute a block of actions	Section 21.11, "Adding For Each Actions"
If...then...	Perform an action or set of actions conditionally, based on the Boolean result of an XQuery expression.	Section 21.12, "Adding If... Then... Actions"
Raise error	Raise an exception with a specified error code (a string) and description.	Section 21.13, "Adding Raise Error Actions"
Reply	Specify that an immediate reply be sent to the invoker.	Section 21.14, "Adding Reply Actions"

Table 21–2   (Cont.) Message Flow - Flow Control Actions

Action	Description	More Information
▶ Resume	Resume message flow after an error is handled by an error handler.	Section 21.15, "Adding Resume Actions"
▶❘ Skip	Specify that at runtime, the execution of the current stage is skipped and the processing proceeds to the next stage in the message flow.	Section 21.16, "Adding Skip Actions"

Table 21–3   Message Flow - Message Processing Actions

Action	Description	More Information
Assign	Assign the result of an XQuery expression to a context variable.	Section 21.17, "Adding Assign Actions"
Delete	Delete a context variable or a set of nodes specified by an XPath expression.	Section 21.18, "Adding Delete Actions"
Insert	Insert the result of an XQuery expression at an identified place relative to nodes selected by an XPath expression.	Section 21.19, "Adding Insert Actions"
Java callout	Invoke a Java method from the pipeline.	Section 21.20, "Adding Java Callout Actions"
MFL transform	Convert non-XML to XML or XML to non-XML in the pipeline.	Section 21.21, "Adding MFL Transform Actions"
Rename	Rename elements selected by an XPath expression without modifying the contents of the element.	Section 21.22, "Adding Rename Actions"
Replace	Replace a node or the contents of a node specified by an XPath expression.	Section 21.23, "Adding Replace Actions"
Validate	Validate elements selected by an XPath expression against an XML schema element or a WSDL resource.	Section 21.24, "Adding Validate Actions"

Table 21–4   Message Flow - Reporting Actions

Action	Description	More Information
Alert	Send an alert notification based on pipeline message context.	Section 21.25, "Adding Alert Actions"

Table 21–4   (Cont.) Message Flow - Reporting Actions

Action	Description	More Information
![] Log	Construct a message to be logged.	Section 21.26, "Adding Log Actions"
![] Report	Enable message reporting for a proxy service.	Section 21.27, "Adding Report Actions"

3. When you have finished adding actions, you can further configure the actions in stage or route node, as described in Table 21–5.

Table 21–5   Edit Stage Configuration Tasks

To...	Complete This Step...
Delete an action	Click the appropriate icon, then click **Delete this Action**.
Move an action down (demote)	Click the appropriate icon, then click **Move Action Down**. The action is moved below the next action contained in this stage. This option is displayed only when a stage contains two or more actions.
Move an action up (promote)	Click the appropriate icon, then click **Move Action Up**. The action is moved above the previous action contained in this stage. This option is displayed only when the stage contains two or more actions.
Cut an action	Click the appropriate icon, then click **Cut**.
Copy an action	Click the appropriate icon, then click **Copy**.
Paste an action that you have cut or copied	Click the appropriate icon, then click **Paste Action**. You can copy and paste actions across stages. However, in the case of Assign, Replace or Insert actions, note the following: - All variable-related and user-defined namespaces from the source (copied) stage are added as user-defined namespaces in the target (pasted) stage. - Duplicate namespaces (identical namespaces in both source and target stage) are not copied. - Conflicting namespaces (namespace declarations that use the same prefix but different URIs) are copied. Users will be able to save the configuration, but will not be able activate it until the conflicting namespace declarations in stage B are removed.
Validate a stage	In the Edit Stage Configuration page, click **Validate** to validate all the actions configured in that stage.

4. Click **Save** to commit the updates in the current session.

5. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

6. Click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

21.2 Adding Publish Actions

Use a publish action to identify a statically specified target service for a message and to configure how the message is packaged and sent to that service.

For more information on publish behavior, see Section 37.4, "Performing Transformations in Message Flows."

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Select **Add an Action** > **Communication** > **Publish**.

3. Click **Service**. The Select Service page is displayed.

4. Select a service from the list, then click **Submit**. This is the target service for the message.

5. If the service has operations defined, you can specify an operation to be invoked by selecting it from the **Operation** list.

6. To make the outbound operation the same as the inbound operation, select the **Use inbound operation for outbound** check box.

7. To configure how the message is packaged and sent to the service, in the **Request Actions** field, click **Add an Action**. Then select an action to associate with the service. You can add more than one action. See Section 21.1, "Adding and Editing Actions in Message Flows."

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.3 Adding Publish Table Actions

Use a publish table action to publish a message to zero or more statically specified services. Switch-style condition logic is used to determine at runtime which services will be used for the publish.

For more information on publish behavior, see Section 37.4, "Performing Transformations in Message Flows."

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Select **Add an Action** > **Communication** > **Publish Table**.

3. Click **Expression**. The XQuery Expression Editor page is displayed. Create an XQuery expression, which at runtime returns the value upon which the routing decision will be made. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. From the **Operator** list, select a comparison operator. Then, in the adjacent field, enter a value against which the value returned from the XQuery expression will be evaluated.

5. Click **Service** to select a service to which messages are to be published if the expression evaluates true for the value you specified. The Select Service page is displayed.

6. Select a service from the list, then click **Submit**. This is the target service for the message.

7. If the service has operations defined, you can specify the operation to be invoked by selecting it from the **invoking** list.

8. If you want the outbound operation to be the same as the inbound operation, select the **Use inbound operation for outbound** check box.

9. In the **Request Actions** field, to configure how the message is packaged and sent to the service, click **Add an Action**, then select one or more actions that you want to associate with the service. To learn more about the type of action you want to add, see Section 21.1, "Adding and Editing Actions in Message Flows."

10. To insert a new case, click the **Case** icon, then select **Insert New Case**.

11. Repeat steps 4-8 for the new case.

12. Add additional cases as dictated by your business logic.

13. Click the **Case** icon of the last case you define in the sequence, then select **Insert Default Case** to add a default case at the end.

14. Configure the default case—the configuration of this case specifies the routing behavior in the event that none of the preceding cases is satisfied.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.4 Adding Dynamic Publish Actions

Use a dynamic publish action to publish a message to a service specified by an XQuery expression.

For more information on publish behavior, see Section 37.4, "Performing Transformations in Message Flows."

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Communication** > **Dynamic Publish**.

3. Click **Expression**.

4. In the XQuery Expression Editor, enter an Xquery expression or select an XQuery resource that provides a result similar to:

```
<ctx:route>
<ctx:service isProxy="false">project/folder/businessservicename</ctx:service>
<ctx:operation>foo</ctx:operation>
</ctx:route>
```

> **Notes:** If a proxy service is being invoked, set `isProxy` to true. If a business service is being invoked, set `isProxy` to false.
>
> The element `operation` is optional.

5. Click **Save**.

6. In the **Request Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action. To learn more about the type of actions you can add, see the table of actions in Section 21.1, "Adding and Editing Actions in Message Flows."

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.5 Adding Routing Options Actions

Use the Routing Options action to modify any or all of the following properties for the outbound request in `$outbound`: URI, Quality of Service, Mode, Retry parameters. Although these properties can be modified using Assign, Insert, Replace, or Delete actions on `$outbound`, using Routing options provides a simpler way to perform this task, without requiring knowledge of XPath, XQuery, or the structure of the `$outbound` context variable.

The Routing Options action can only be used where the context variable `$outbound` is valid. It can be added to the following actions:

- Publish

- Dynamic Publish

- Publish Table

- Service Callout

- Routing

- Dynamic Routing

- Routing Table

For more information on routing, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

To configure a routing options action:

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Communication** > **Routing Options**.

3. Complete any or all of the following steps:

   - To set the URI for the outbound message: Select **URI**, and click the **XQuery Expression Editor**. Enter an expression that returns a URI. This overrides the URI for the invoked service.

> **Note:** When routing to another proxy service, the URI override has no effect.

- To set the Quality of Service element: Select **Quality of Service**, and select the Quality of Service option from the list. This overrides the default that is auto computed.

- To set the Mode: Select **Mode**, and select either request, or request-response from the list.

> **Note:** This is normally already automatically set, based on the interface of the service invoked. However, in some cases like Any Soap or Any XML services, this is not so.

- To set the Retry Interval: Select **Retry Interval**, and specify the number of seconds between retries. This overrides the default configured with the invoked service.

- To set the Retry Count: Select **Retry Count**, and specify the number of retries the system must attempt before discontinuing the action. This overrides the default configured with the invoked service.

- To set the Message Priority: Select **Priority**, and click the **XQuery Expression Editor**. Enter an expression that returns a positive integer.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.6 Adding Service Callout Actions

Use a service callout action to configure a synchronous (blocking) callout to an Oracle Service Bus-registered proxy or business service.

For more information on service callout actions, see Section 37.5, "Constructing Service Callout Messages."

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Communication** > **Service Callout**.

3. Click **Service**. The Service Browser is displayed.

4. Select a service from the list of registered proxy or business services, then click **Submit**.

5. If the service you chose in step 3, above, is WSDL-based and has operations that can be invoked on the service, those operations are listed in the **invoking Operation** list. Select an operation to be invoked on the service.

> **Note:** Selecting an operation, which Oracle Service Bus requires for many reasons, does not guarantee that only the selected operation is invoked. For example, if you select OperationA, but a message also contains an invocation for Operation B, then OperationB will be invoked as well.

6. Specify how you want to configure the request and response messages by selecting one of the following options:

   ■ Select **Configure SOAP Body** to configure the SOAP Body. Selecting this option lets you use `$body` directly.

   > **Note:** This option supports SOAP-RPC encoded, which is not supported when configuring payload parameters or document.

   ■ Select **Configure Payload Parameters** or **Configure Payload Document** to configure the payload.

7. Subsequent configuration options depend on the kind of service you selected and on the kind of configuration options you chose for that service.

   Table 21–6 provides instructions for each option.

Table 21–6   Service Callout Configuration Options

For These Options...	Follow These Steps...
SOAP Request Body and SOAP Response Body	To configure these options, ■ In the **SOAP Request Body** field, enter the name of a variable to hold the XML of the SOAP Body element for the callout request. ■ In the **SOAP Response Body** field, enter the name of a variable to which the XML of the SOAP Body element on the response will be bound.
SOAP Request Header and SOAP Response Header	To configure these options, ■ In the **SOAP Request Header** field, enter the name of a variable to hold the XML of the SOAP Header element for the callout request You must wrap the input document for the SOAP Request Header with `<soap-env:Header>...</soap-env:Header>`. ■ In the **SOAP Response Header** field, enter the name of a variable to which the XML of the SOAP Headers on the response, if any, will be bound.

Table 21–6   (Cont.) Service Callout Configuration Options

For These Options...	Follow These Steps...
Request Parameters and Response Parameters	To configure options, ■ In the **Request Parameters** fields, enter names for the variables that will be evaluated at runtime to provide values for the request parameters. You must provide only the core payload documents in the input variable—the SOAP package is created for you by Oracle Service Bus. In other words, do not wrap the input document with `<soap-env:Body>...</soap-env:Body>`. For example, when creating a body input variable that is used for this request parameter, you would define that variable's contents using the XPath statement `body/*` (to remove the wrapper `soap-env:Body`), not `$body` (which results in keeping the `soap-env:Body` wrapper). ■ In the **Response Parameters** fields, enter the names of the variables to which the responses will be assigned at runtime.
Request Document and Response Document	To configure these options, ■ In the **Request Document Variable** field, enter the name of a variable to assign a request document to. For SOAP Document-type services, the variable is evaluated at runtime to form the body of the SOAP message sent to the service. For *Any XML* services, the variable is evaluated at runtime to form the body of the XML message sent to the service. For SOAP Document-type services and for Any XML services, you provide only the core payload documents in the input variable—the SOAP package is created for you by Oracle Service Bus. In other words, do not wrap the input document with `<soap-env:Body>...</soap-env:Body>`. For example, when creating a body input variable that is used for this request parameter, you would define that variable's contents using the XPath statement `body/*` (to remove the wrapper `soap-env:Body`), not `$body` (which results in keeping the soap-env:Body wrapper). For *Messaging* services, the variable is evaluated to form the body of the message, based on the type of data expected by the service. The following restrictions apply to variables used with Messaging services: - For services that expect binary data, the variables must have a `ctx:binary-content` element. - For services that expect MFL data, the variable must have the XML equivalent. - For services that expect text data, the variable is a string. ■ In the **Response Document Variable** field, enter the name of the variable to which a response document will be assigned at runtime.

8. Optionally, add one or more transport headers. For more information, see Section 21.7, "Adding Transport Header Actions."

> **Note:** In addition to the transport headers you specify, headers are added by the Oracle Service Bus binding layer. For more information, see "Configuring Transport Headers in Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.7 Adding Transport Header Actions

Use a transport header action to set the header values in messages.

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Communication** > **Transport Headers**.

3. From the **Set Transport Headers for** list, select one of the following, to specify to the runtime which of the message context locations are to be modified:

   - **Outbound Request** - Select this option to set header values for outbound requests (the messages sent out by a proxy service in route, publish, or service callout actions). This header element is located in the message context as follows:

     ```
     $outbound/ctx:transport/ctx:request/tp:headers
     ```

   - **Inbound Response** - Select this option to set header values for inbound responses (the response messages a proxy service sends back to clients). This header element is located in the message context as follows:

     ```
     $inbound/ctx:transport/ctx:response/tp:headers
     ```

4. Optionally, select **Pass all Headers through Pipeline** to pass all headers through from the inbound message to the outbound message or vice versa. Every header in the source set of headers will be copied to the target header set, overwriting any existing values in the target header set.

   For information about using this option in conjunction with the header-specific pass through option, "Configuring Transport Headers in Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

5. Complete the following steps for each Header you want to add:

   a. In the Transport Headers table, click **Add Header** to display fields for configuring the header.

   b. Specify a header by doing either of the following:

      - From the list in the Name column, select a header name. The list contains all of the predefined header names for the target transport (for example, **Content-Type** for HTTP transports, **JMSCorrelationID** for JMS transports, and so on).

–   Enter a header name in the **Other** field. If that header name is not one of the predefined headers for this service's transport, it becomes a user-header, as defined by the transport specification.

c. Select one of the options in the Action column to specify how to set the headers value:

Set Header to Expression

Selecting this option lets you use an XQuery or XSLT expression to set the value of the header. The expression can be simple (for example, "text/xml") or a complex XQuery or XSLT expression.

Because the Oracle Service Bus transport layer defines the XML representation of all headers as string values, the result of any expression is converted to a string before the header value is set. Expressions that return nothing result in the header value being set to the empty string. You cannot delete a header using an expression.

Caution: Not all of the header settings you can specify in this action are honored at runtime.For information about which of the headers for a given transport you can set and which of those set are honored at runtime, "Configuring Transport Headers in Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Delete Header

Specifies that the header is removed from the request or response metadata.

Copy Header from Inbound Request (if you are setting transport headers for the Outbound Request)

or

Copy Header from Outbound Response (if you are setting transport headers for the Inbound Response)

Specifies that this header is copied directly from the corresponding header of the same name from the inbound message to the outbound message and vice versa. For example, to set the SOAPAction header for an outbound request, selecting **Copy Header from Inbound Request** causes the runtime to copy the value from the SOAPAction request header of `$inbound`. In the case of inbound response headers, the source of the header to copy is the response headers of `$outbound`.

If the **Copy Header** option is selected for a header that does not exist in the source, this option is ignored and no action is performed on the target for this header.

For information about using this option in conjunction with the global **Pass all Headers through Pipeline** option, "Configuring Transport Headers in Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

6. To add additional Headers to the table, click the **Header** icon, then click **Add Header**.

The table is expanded to include an additional row, which includes a new set of options that you can use to configure another transport header. You can add as many headers as necessary to this table. You do not have to order the headers in

the table, because the runtime declares namespaces and places header elements in their proper order when generating the corresponding XML.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.7.1 Setting Cookies in Outbound HTTP Transport Headers

In an HTTP message flow you can set cookies on the transport header in the following ways:

- With a complex XML expression (Oracle Service Bus default)
- With a string expression (requires server startup option)

21.7.1.1 Setting a Cookie as a Complex XML Expression

To set a cookie using a complex XML expression, which is the Oracle Service Bus default format, configure the value of the HTTP Cookie header in the outbound request using the following expression syntax:

```
<cookie-values xmlns="http://www.bea.com/wli/sb/transports/http">
    <value>{fn:concat("cookie_name", "=", "cookie_value")}</value>
</cookie-values>
```

21.7.1.2 Setting a Cookie with a String Expression

To set the Cookie header with a string in the outbound request, you must add the following option to your domain startWebLogic command:

```
-Dcom.bea.osb.http.cookieAsNoComplexElement=true
```

After you restart the server with this option, you can set an HTTP Cookie header with a string expression. For example:

$cookie_name = "cookie_value"

21.8 Adding Dynamic Routing to Route Nodes

Assign a route for a message based on routing information available in an XQuery resource.

This is a terminal action, which means you cannot add another action after this one. However, this action can contain request and response actions.

For more information on routing, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add Dynamic Routing to a Route Node

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the **Route Node** icon, then click **Edit Route**. The Edit Stage Configuration page is displayed.

3. Click the **Add an Action** icon, then select **Communication** > **Dynamic Routing**.

4. Click **Expression**. The XQuery Expression Editor is displayed.

5. In the XQuery Expression Editor, enter an Xquery expression, the result of which is similar to:

```
<ctx:route>
    <ctx:service isProxy='true'>{$service}</ctx:service>
    <ctx:operation>{$operation}</ctx:operation>
</ctx:route>
```

> **Note:** If a proxy service is being invoked, set `isProxy` to true. If a business service is being invoked, set `isProxy` to false.
>
> - The service name is the fully qualified service name.
>
> - The operation element is optional.

6. Click **Save**.

7. In the **Request Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action. To learn more about the type of actions you want to add, see the table of actions in Section 21.1, "Adding and Editing Actions in Message Flows."

8. In the **Response Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action. To learn more about the type of actions you want to add, see the table of actions in Section 21.1, "Adding and Editing Actions in Message Flows."

9. Click **Save**.

10. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

11. Click **Save** to commit the updates in the current session.

12. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

21.9 Adding Routing Actions to Route Nodes

Identify a target service for the message and configure how the message is routed to that service.

This is a terminal action, which means you cannot add another action after this one. However, this action can contain request and response actions.

For more information on routing, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Routing Action to a Route Node

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the **Route Node** icon, then click **Edit Route**. The Edit Stage Configuration page is displayed.

3. Click the **Add an Action** icon, then select **Communication** > **Routing**.

4. Click **Service**. The Service Browser is displayed.

5. Select a service from the list, then click **Submit**. The service is displayed instead of the default link.

6. If you want the outbound operation to be the same as the inbound operation, select the **Use inbound operation for outbound** check box.

7. In the **Request Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action. To learn more about the type of actions you can add, see the table of actions in Section 21.1, "Adding and Editing Actions in Message Flows."

8. In the **Response Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action. To learn more about the type of actions you can add, see the table of actions in Section 21.1, "Adding and Editing Actions in Message Flows."

9. Click **Save**.

10. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

11. Click **Save** to commit the updates in the current session.

12. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

21.10 Adding Routing Tables to Route Nodes

A routing table is a set of routes wrapped in a switch-style condition table. It is a short-hand construct that allows different routes to be selected based upon the results of a single XQuery expression. You can nest multiple levels of branching actions in the stage editor.

Identify target services for messages and configure how the messages are routed to these services.

This is a terminal action, which means you cannot add another action after this one. However, this action can contain request and response actions.

For more information on routing, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Routing Table to a Route Node

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the **Route Node** icon, then click **Edit Route**. The Edit Stage Configuration page is displayed.

3. Click the **Add an Action** icon, then select **Communication** > **Routing Table**. The routing table action is displayed.

4. From the **Operator** list, select a comparison operator, then enter a value expression in the adjacent field.

5. Click **Service**. The Select Service page is displayed.

6. Select a service from the list, then click **Submit**.

7. To invoke an operation on the service, select an operation from the **Operation** list

8. If you want the outbound operation to be the same as the inbound operation, select the **Use inbound operation for outbound** check box.

9. In the **Request Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action.

10. In the **Response Actions** field, click **Add an Action** to add an action, then select an action that you want to associate with the service. You can add more than one action.

    To learn more about the types of request and response actions you can add, see Section 21.1, "Adding and Editing Actions in Message Flows."

11. To insert a new case, click the **Case** icon, then select **Insert New Case**.

12. Repeat steps 2-7 for the new case. You can click the **Case** icon, then select **Insert Default Case** to add a default case at the end whose routes are selected if none of the preceding cases is satisfied.

13. Click **Save**.

14. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

15. Click **Save** to commit the updates in the current session.

16. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

21.11 Adding For Each Actions

Use the for each action to iterate over a sequence of values and execute a block of actions.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a For Each action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **For Each**.

3. Enter variable names in the **variable** fields, click **XPath** to open the XPath editor to create an XPath expression, and configure the actions in the **Do ()** loop.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.12 Adding If... Then... Actions

Use an if...then... action to perform an action or set of actions conditionally, based on the Boolean result of an XQuery expression.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an If...Then action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **If...Then**.

3. Click **Condition** to display the XQuery Condition Editor page.

   The condition you create is used as the test that is executed before the `then()` clause is entered, per standard if...then logic. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. When you finish editing the XQuery condition, click **Add an Action**, then select an action that you want to associate with the condition. To learn more about the type of action you want to add, see Section 21.1, "Adding and Editing Actions in Message Flows."

   In the route node, you can select only the routing, dynamic routing, or routing table actions. However, these actions can contain request and response actions inside of them.

5. As your logic requires, click the **If...Then...** icon, then click **Add else-if Condition** or **Add else Condition** to add `else-if` conditions or `else` conditions. Click **Add an Action** to associate actions with these conditions.

   Condition actions can be nested.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.13 Adding Raise Error Actions

Use the raise error action to raise an exception with a specified error code (a string) and description.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Raise Error action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **Raise Error**.

3. In the **error code** field, enter the error code you want to raise.

4. In the **error message** field, enter a description of the error code.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.13.1 Transactions

If a service is transactional, a triggered Raise Error action aborts the transaction in the request (asynchronous) or in either the request or response (synchronous). For example, you may introspect messages and determine conditions under which a Raise Error action should occur even if no SOAP fault occurs, and Raise Error causes the transaction to be aborted.

21.14 Adding Reply Actions

Use the reply action to specify that an immediate reply be sent to the invoker.

The reply action can be used in the request, response or error pipeline. You can configure it to result in a reply with success or failure. In the case of reply with failure where the inbound transport is HTTP, the reply action specifies that an immediate reply is sent to the invoker.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Reply action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **Reply**.

3. Select **With Success** to reply that the message was successful, or select **With Failure** to reply that the message has a fault.

    Reply With Failure will cause a transaction, if started by Oracle Service Bus, to be aborted.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.15 Adding Resume Actions

Use the resume action to resume message flow after an error is handled by an error handler. This action has no parameters and can only be used in error pipelines.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Resume action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **Resume**.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.16 Adding Skip Actions

Use the skip action to specify that at runtime, the execution of this stage is skipped and the processing proceeds to the next stage in the message flow. This action has no parameters and can be used in the request, response or error pipelines.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Skip action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Flow Control** > **Skip**.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.17 Adding Assign Actions

Use the assign action to assign the result of an XQuery expression to a context variable.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an Assign action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Assign**.

3. Click **Expression**. The XQuery Expression Editor page is displayed. The XQuery expression is used to create the data that will be assigned to the named variable. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. When you finish editing the expression, enter a context variable in the variable field. To learn more about context variables, see "Inbound and Outbound Variables" and "Constructing Messages to Dispatch" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.18 Adding Delete Actions

Use the delete action to delete a context variable or a set of nodes specified by an XPath expression.The delete action is one of a set of update actions.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Delete action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. To delete a context variable, select the **Variable** option, then enter the name of a context variable in the **Variable** field.

   Alternatively, to delete all nodes selected by an XPath expression, select the **XPath** option, then click **XPath**. The XPath Expression Editor page is displayed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions." After you save the expression, enter a context variable in the **variable** field.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.19 Adding Insert Actions

Use the insert action to insert the result of an XQuery expression at an identified place relative to nodes selected by an XPath expression. The insert action is one of a set of update actions.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an Insert action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Insert**.

3. Click **Expression** to edit an XQuery expression. The XQuery expression is used to create the data that will be inserted at a specified location in a named variable. The XQuery Expression Editor page is displayed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. When you finish editing the expression, select the relative location from the list. The relative location is used to control where the insert is performed relative to the result of the XPath expression:

   - **Before**—as sibling before each element or attribute selected by the XPath expression

   - **After**—as sibling after each element or attribute selected by the XPath expression

- **As first child of**—as first child of each element identified by the XPath expression. An error occurs if the result of the XPath returns attributes.

- **As last child of**—as last child of each element identified by the XPath expression. An error occurs if the XPath returns attributes.

5. Click **XPath**. The XPath Expression Editor page is displayed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

Valid configurations include those in which:

- XQuery and XPath expressions both return elements.

- The XQuery and XPath expressions both return attributes—in which case, the XQuery expression must return attributes.

6. When you finish editing the XPath expression, enter a context variable in the in variable field. The XPath evaluates the contents of this variable.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.20 Adding Java Callout Actions

Use the Java callout action to invoke a Java method, or EJB business service, from within the message flow.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Java Callout action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Java Callout**.

3. Click **Method**. The Select a JAR page is displayed. Select a JAR resource from the list. The Select a Class and Method page is displayed.

4. From the list of Java classes listed, click the + beside the appropriate class, to display a list of methods. Select a method and click **Submit**. The Java callout action is displayed on the Edit Stage page, as follows:

   - **Method** is replaced by the name of the Java method you selected in steps 2 and 3. This name is a link to the Select a Class and Method page. You can click this link to change your selection of Java method.

     The method must be a static method.

   - **Parameters**: An **Expression** link to the XQuery Expression Editor page is provided for each argument the Java method requires. A label for each link indicates the data type for the argument, which will be one of the following:

     – `Java.lang.String`

     – Primitive types, and their corresponding class types (for example, `int` vs. `java.lang.Integer`)

- – `java.lang.BigDecimal`, and `java.lang.BigInteger` (these types are used in financial calculations where round-off errors or overflows are not tolerable)

  - – only `org.apache.xbeans.XmlObject` and no typed xml beans.

  - – `byte[]`

  - – `java.lang.String[]` (INPUT ONLY)

  - – `XmlObject[]` (INPUT ONLY)

  - – `javax.activation.DataSource`

- ■ **Result**: A **Result** field in which you enter the variable to which the result is to be assigned. The label for the field indicates the data type of the result.

Note: If the result is a byte array (the only possible array returned), the binary-content XML element is returned.

- ■ **Return Parameter as Reference**: This option makes the return value of a Java Callout invocation a <java-content ref="jcid"> reference element regardless of its actual type, where jcid is the key to the object in the pipeline object repository. In the Result value field, enter the name of the variable to contain the java-content reference. This option lets you work with a referenced object in the message flow in addition to the pipeline XML for providing passthrough, performing message enrichment with Java Callout and inline actions, or performing message transformation between Java and non-Java transports. For more information, see "Sending and Receiving Java Objects in Messages" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

- ■ **Attach a Service Account**: A **Service Account** link lets you specify an optional Service Account if there is a security context for this Java method. To learn more about security contexts and service accounts, see Chapter 17, "Service Accounts."

  In the case of fixed and mapped service accounts, the userid/password from the service account is authenticated in the local system and the security context propagated to the Java callout. In the case of `passthru`, the security context is propagated to the Java callout. This context is the message level context if defined (with WS-Security). Else it is the transport level context.

5. Under Parameters, click **Expression**. The XQuery Expression Editor page is displayed. Use the XQuery Expression Editor to provide the arguments required by the Java method. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

   If the type of the input value you enter does not match the declared input argument type, Oracle Service Bus tries to automatically typecast input values to the declared type of the input argument. For example a string value of `"123"` will be converted to integer `123` if the declared type of the input argument is java primitive `int`.

6. In the **Result** field, assign a variable for the result returned by the Java method.

7. If there is a security context for the Java method, select the check box and click **Service Account**. The Select Service Account page is displayed. Select the required service account from the list, and click **Submit**.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.21 Adding MFL Transform Actions

Use the MFL (Message Format Language) transform action to convert message content from XML to non-XML, or vice versa, in the message pipeline. An MFL is a specialized XML document used to describe the layout of binary data. It is an Oracle proprietary language used to define rules to transform formatted binary data into XML data, or vice versa. See Section 16, "MFLs."

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an MFL Transform action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **MFL Transform**.

3. From the **Apply MFL Transformation** list, select **XML to Non-XML** or **Non-XML to XML**, according to your requirement.

4. Click **Expression**. Using the XQuery Expression Editor, specify the variable on which the MFL transformation action is to be performed. This input must be text or binary when transforming to XML, and must be XML when transforming to non-XML. Binary content in the message context is represented by the binary-content XML element. This XML should be the result of the Xquery expression when the input must be binary. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

5. Select one of the following options:

    ■   **MFL Resource**: click the **resource** link. The Select MFL page is displayed. Select the static MFL resource that will perform the MFL transform action.

    ■   **MFL Resource from**: click the **Expression** link. The XQuery Expression Editor page is displayed. Using the XQuery Expression Editor, create or edit an XQuery expression to dynamically specify an MFL resource that will perform the transform action, in the format project/folder/MFLresourcename. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

6. In the **Assign to Variable** field, enter the name of the variable to which the result of this transform action is to be assigned. The result will be a binary-content XML element.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.22 Adding Rename Actions

Use the rename action to rename elements selected by an XPath expression without modifying the contents of the element. The rename action is one of a set of update actions.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Rename action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Rename**.

3. Click **XPath**. The XPath Expression Editor page is displayed. The XPath expression is used to specify the data (in the named variable) that will be renamed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. In **variable** field, enter the context variable that holds the element you want to rename.

5. Do one of the following:

    ■   To rename selected elements using a localname, select the first **localname** option, then enter a local name in the **localname** field.

    ■   To rename selected elements using a namespace, select the first **namespace** option, then enter a namespace in the **namespace** field.

    ■   To rename selected elements using a local name and namespace, select the **localname and namespace** option, then enter a local name and namespace in the **localname and namespace** fields.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.23 Adding Replace Actions

Use a replace action to replace a node or the contents of a node specified by an XPath expression. The node or its contents are replaced with the value returned by an XQuery expression.

A replace action can be used to replace simple values, elements and even attributes. An XQuery expression that returns nothing is equivalent to deleting the identified nodes or making them empty, depending upon whether the action is replacing entire nodes or just node contents.

The replace action is one of a set of update actions.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Replace action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Replace**.

3. Click **XPath**. The XPath Expression Editor page is displayed. The XPath expression is used to specify the data (in the named variable) that will be replaced. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

4. When you finish editing the XPath expression, enter a context variable in the in **variable** field.

5. Click **Expression**. The XQuery Expression Editor page is displayed. The XQuery expression is used to create the data that replaces the data specified by the XPath in the named variable.See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

6. When you finish editing the XQuery expression, select one of the options:

   ■ **Replace entire node**—to specify that the nodes selected by the XPath expression you defined are replaced along with all of its contents

   ■ **Replace node contents**—to specify that the node is not replaced; only the contents are replaced.

   > **Note:** Selecting the **Replace node contents** option and leaving the **XPath** field blank is more efficient than selecting the **Replace entire node** option and setting the XPath to `./*`

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.24 Adding Validate Actions

Use a validate action to validate elements selected by an XPath expression against an XML schema element or a WSDL resource. You can validate global elements only; Oracle Service Bus does not support validation against local elements.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Validate action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Message Processing** > **Validate**.

3. Click **XPath**. to construct an XPath expression that specifies the elements to be validated. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions." When you are finished constructing the expression in the XPath Expression Editor, click **Save** to insert the expression on the Edit Stage Configuration page.

4. In the **in variable** field, enter the name of the variable to hold the element to be validated.

5. Click **resource**, then select **WSDL** or **Schema**.

6. From the WSDL Browser or XML Schema Browser, do the following:

   a. Select the WSDL or XML schema

   b. Select the WSDL or XML schema type or element

   c. Click **Submit**.

7. To save the result of this validation (a boolean result), select **Save result of validation in variable** and enter the name of the variable in which you want to save the result.

   Alternatively, to raise an error if the element fails validation against the WSDL or XML schema element, select **Raise Error** on validation failure.

After you are finished

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.25 Adding Alert Actions

Use the alert action to generate alerts based on message context in a pipeline, to send to an alert destination. Unlike SLA alerts, notifications generated by the alert action are primarily intended for business purposes, or to report errors, and not for monitoring system health. Alert destinations should be configured and chosen with this in mind. To learn more about alert destinations, see Section 6, "Alert Destinations."

If pipeline alerting is not enabled for the service or at the domain level, the configured alert action is bypassed during message processing.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an Alert action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Reporting** > **Alert**.

3. Click **Destination**. The Select Alert Destination page is displayed. Select the required alert destination from the list and click **Submit**.

   By default, the alert will always go to the Administration Console.

4. Click **Expression**. The XQuery Expression Editor page is displayed. You specify the message context to be added to the alert message through XQuery expressions on context variables. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

5. In the **alert summary** field, enter a short description of the alert. This will be the subject line in the case of an email notification, and can contain no more than 80 characters. If no description is provided, a predefined subject line that reads, "Oracle Service Bus Alert", will be used instead.

6. In the **severity level** list, select a severity level for this alert from among: **Normal**, **Warning**, **Minor**, **Major**, **Critical**, and **Fatal**.

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.26 Adding Log Actions

Use the log action to construct a message to be logged and to define a set of attributes with which the message is logged.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

> **Note:** To see log data in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:
>
> - Minimum severity to log: Info
>
> - Log file: Info
>
> - Standard out: Info
>
> For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

To add a Log action

1. Be sure Logging is enabled globally. For more information, see Section 27.2.9, "Enabling Logging Globally."

2. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

3. Click the appropriate icon, then select **Add an Action** > **Reporting** > **Log**.

4. Click **Expression**. The XQuery Expression Editor page is displayed. You specify the message context to be logged through XQuery expressions on context variables. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

5. In the **Annotation** field, enter notes for this log action. These notes are logged along with the result of the previously defined expression.

6. In the **severity level** list, select one of the options.

Table 21–7   Log Action Severity Levels

Severity Level	Typical Usage
Info	Used for reporting normal operations; a low-level informational message.
Warning	A suspicious operation or configuration has occurred but it might not affect normal operation.

Table 21–7   (Cont.) Log Action Severity Levels

Severity Level	Typical Usage
Error	A user error has occurred. The system or application can handle the error with no interruption and limited degradation of service.
Debug	While your application is under development, you might find it useful to create and use messages that provide verbose descriptions of low-level activity within the application.

Make sure the Log severity level on the proxy service's operational settings is the same as the log action severity level. For information on proxy service operational settings, see Section 26.8, "Configuring Operational Settings for Proxy Services."

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

21.27 Adding Report Actions

Use the report action to enable message reporting for a proxy service.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a report action

1. Navigate to where you want to add the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

2. Click the appropriate icon, then select **Add an Action** > **Reporting** > **Report**.

3. Click **Expression**. The XQuery Expression Editor page is displayed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions." The XQuery expression is used to create the data that will be reported to the Oracle Service Bus dashboard.

4. When you finish editing the XQuery expression, click **Add a Key**. Two fields are displayed: a **Key Name** field and a **Key Value** field, which includes an XPath link that you can click to edit an XPath expression and an in variable field in which you can enter a context variable.

   You use key value pairs to extract key identifiers from any message context variable or message payload, and ignore the rest of the message. The keys are a convenient way to identify a message. They are displayed as report indexes in the Reporting module. See Section 28.1, "Viewing Reporting Messages" and Section 28.2, "Viewing Message Details."

   a. Enter a key name in the **Key Name** field.

   b. Click **XPath**. The Edit an XPath Expression page is displayed. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

   c. Enter a context variable in the in **variable** field.

   d. To add more key values, click the **Key** icon, then select **Add a Key**. To delete a key, click the **Key** icon, then select **Delete this Key**.

For example, consider a report action configured on an error handler in a stage. The action reports the contents of the `fault` context variable in the event of an error. The report action is configured as follows:

- Key name = `errorCode`

- Key value = `./ctx:errorCode` in variable `fault`

Each time this action is executed at runtime, a message is reported through the Reporting Data Stream. The following table shows the results after the report action is executed twice.

Report Index	DB TimeStamp	Inbound Service	Error Code
errorCode=BEA-382505	04/26/07 9:45 AM	MortgageBroker/ProxySvcs/loanGateway3	BEA-382505
errorCode=BEA-382505	04/26/07 9:45 AM		BEA-382505

After you finish

When you complete the configuration of this action, continue by configuring other actions or by saving your configuration, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

22

Proxy Services: Message Flow

This chapter describes how to create and configure message flows using the Oracle Service Bus Administration Console. Sections include adding and configuring pipeline pairs, conditional branches, stages, operational branches, and route nodes. For more detailed information on message flows, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

For more detailed information on message flows and their components, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

22.1 Viewing and Editing Message Flows

This section describes how to view and edit message flows.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the proxy service whose message flow you want to view or edit. Do either of the following:

    ■   Select **Project Explorer** to display the Projects View page or the Project/Folder View page. Then navigate through projects and folders to find the proxy service.

    ■   Select **Resource Browser** > **Proxy Services** to display the Summary of Proxy Services page.

3. Click the **Edit Message Flow** icon in the row of the proxy service whose message flow you want to view or edit.

    ■   If no message flow has yet been created for the selected proxy service, the Edit Message Flow page is displayed with a single icon on the page, the Proxy Service icon. This is the starting node for the proxy service. Click this icon to begin constructing the message flow.

    ■   If the proxy service already has a message flow, the page contains a graphic representation of the flow. Click the icons to view or edit the parts of the message flow.

    See Section 22.2, "Edit Message Flow Page" for information about what you can do to construct a message flow on that page.

4. Click **Save** to commit the updates in the current session.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.2 Edit Message Flow Page

Use the Edit Message Flow page to construct a message flow for a proxy service.

The left navigation pane of the Edit Message Flow page shows a tree view of the nodes and objects in the message flow. When the details of an object are defined on a separate page, you can click the name of the object to display the associated page.

The right pane provides a field upon which to construct the message flow. When a message flow has not yet been defined, the pane includes a single Proxy Service icon that signifies the starting node for the service. Click the icon to add pipeline pair nodes, route nodes, conditional branches, operational branches, and error handling for the service.

When you add objects to the page, icons are displayed on the page to represent the objects. The relationships among the objects are shown with lines and bounding boxes. Click an icon on the **Edit Message Flow** page to display a menu of the actions you can perform on that object. The options available on the menu may differ, depending on context. See Table 22–1 for a complete list of icons and options.

Table 22–1   Edit Message Flow Page Icons and Options

Icon	Description	Menu Options
Proxy Service	The starting node for the message flow.	■ **Add Pipeline Pair** - See Section 22.4, "Adding Pipeline Pair Nodes to Message Flows." ■ **Add Route** - See Section 22.8, "Adding Route Nodes to Message Flows." ■ **Add Conditional Branch** - See Section 22.5, "Adding Conditional Branches to Message Flows." ■ **Add Operational Branch** - See Section 22.6, "Adding Operational Branches to Message Flows." ■ **Add Service Error Handler** - See Section 24.2, "Adding Proxy Service Error Handlers."
Pipeline Pair Node	A pipeline pair node consists of a request pipeline and a response pipeline.	■ **Edit Name and Annotation** ■ **Add Pipeline Pair** - See Section 22.4, "Adding Pipeline Pair Nodes to Message Flows." ■ **Add Conditional Branch** - See Section 22.5, "Adding Conditional Branches to Message Flows." ■ **Add Operational Branch** - See Section 22.6, "Adding Operational Branches to Message Flows." ■ **Paste Route** - This option is available only if you have cut or copied a route node and it is on the Clipboard.
Response Pipeline	See pipeline pair node above.	■ **Add Stage** - See Section 22.7, "Adding Stages to Pipelines." ■ **Add Pipeline Error Handler** - See Section 24.3, "Adding Pipeline Error Handlers."

Table 22–1   (Cont.) Edit Message Flow Page Icons and Options

Icon	Description	Menu Options
Request Pipeline	See pipeline pair node above.	■ **Add Stage** - See Section 22.7, "Adding Stages to Pipelines." ■ **Add Pipeline Error Handler** - See Section 24.3, "Adding Pipeline Error Handlers."
Pipeline with Error Handler	A pipeline with an error handler defined for it.	■ **Edit Pipeline Error Handler** - See Section 24.3, "Adding Pipeline Error Handlers." ■ **Delete Pipeline Error Handler**
Route Node	Route node actions define the handling of messages as they flow through the route node.	■ **Edit Route** - See Section 22.8, "Adding Route Nodes to Message Flows." ■ **Edit Name and Annotation** ■ **Add Route Error Handler** - See Section 24.5, "Adding Route Node Error Handlers."
Route Node with Error Handler	A route node with an error handler defined for it.	■ **Edit Route Error Handler** - See Section 24.5, "Adding Route Node Error Handlers." ■ **Delete Route Error Handler**
Stage Node	A stage node is a container of actions.	■ **Edit Stage** - See Section 22.7, "Adding Stages to Pipelines." ■ **Edit Name and Annotation** ■ **Add Stage** - See Section 22.7, "Adding Stages to Pipelines." ■ **Add Stage Error Handler** - See Section 24.4, "Adding Stage Error Handlers."
Stage Node with Error Handler	A stage node with an error handler defined for it.	■ **Edit Stage Error Handler** - See Section 24.4, "Adding Stage Error Handlers." ■ **Delete Stage Error Handler**
Conditional Branch Node	A conditional branch node allows processing to proceed down exactly one of several possible paths.	■ **Edit Branch** - See Section 22.5, "Adding Conditional Branches to Message Flows." ■ **Edit Name and Annotation**
Operational Branch Node	An operational branch node determines what branch to follow based on specified operations.	■ **Edit Branch** - See Section 22.6, "Adding Operational Branches to Message Flows." ■ **Edit Name and Annotation**

Table 22–1   (Cont.) Edit Message Flow Page Icons and Options

Icon	Description	Menu Options
Branch Node	A branch node is one of the alternative nodes defined by a conditional branch node or an operational branch node.	■ **Add Pipeline Pair** - See Section 22.4, "Adding Pipeline Pair Nodes to Message Flows." ■ **Add Route** - See Section 22.8, "Adding Route Nodes to Message Flows." ■ **Add Conditional Branch** - See Section 22.5, "Adding Conditional Branches to Message Flows." ■ **Add Operational Branch** - See Section 22.6, "Adding Operational Branches to Message Flows." ■ **Paste Route** - This option is available only if you have cut or copied a route node and it is on the Clipboard.
Error Handler	An error handler provides the logic for resending errors in the message flow.	■ **Add Service Error Handler** - See Section 24.2, "Adding Proxy Service Error Handlers."

22.3 Edit Stage Configuration Page

Use the Edit Stage Configuration page to add actions to pipeline stages, error handler stages, and route nodes in a message flow.

■ When nothing has yet been defined on the Edit Stage Configuration page, the only object displayed is the Add an Action icon. Click that icon to get started.

■ When a stage or a route node has already been configured, the actions and objects defined for that stage or route node appear on the page. Edit the existing actions, as appropriate, or click any of the icons representing actions to add more actions to the stage.

See Section 21.1, "Adding and Editing Actions in Message Flows" for instructions on working with all the kinds of actions you can add to a stage.

22.4 Adding Pipeline Pair Nodes to Message Flows

Message flows can include zero or more pipeline pair nodes: 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.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Pipeline Pair node to a message flow

1. Click the **Proxy Service** icon, then click **Add Pipeline Pair**.

2. To change the default name and add a description for the pipeline pair node, do the following:

    a. Click the **Pipeline Pair Node** icon, then click **Edit Name and Annotation**.

    b. Change the name and description, as desired.

    c. Click **Save**.

> **Note:** When you rename a pipeline or a route node, the number of messages displayed on the Dashboard page in the Monitoring module may not correlate with those of other components due to the pipeline counters being reset to zero. This is because Oracle Service Bus treats the rename as a delete and recreate action. The numbers should correlate again after a time period equal to the service's monitoring interval has elapsed.

3. To add stages to the pipeline, see Section 22.7, "Adding Stages to Pipelines."

4. To add actions to stages in the pipeline. See Section 21.1, "Adding and Editing Actions in Message Flows."

5. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

6. Click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.5 Adding Conditional Branches to Message Flows

A branch node allows processing to proceed along exactly one of several possible paths. Branching is driven by an XPath-based switch table. Each branch in the table specifies a condition (for example, `<500`) that is evaluated in order down the message flow against a single XPath expression (for example, `./ns:PurchaseOrder/ns:totalCost` on `$body`). Whichever condition is satisfied first determines which branch is followed. If no branch condition is satisfied, then the default branch is followed. A branch node may have several descendants in the message flow: one for each branch, including the default branch.

If the proxy service is not based on a WSDL and receives multiple document types as input, consider using a conditional branch node.

Conditional branching is driven by a lookup table with each branch tagged with a simple, but unique, string value. A variable in the message context is designated as the lookup variable for that node, and at runtime, its value is used to determine which branch to follow. If no branch matches the value of the lookup variable, the default branch is followed. You should design the proxy service in such a way that the value of the lookup variable is set before reaching the branch node.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a conditional branch to a message flow

1. Click a **Pipeline Pair Node** icon or a **Branch Node** icon, then click **Create Conditional Branch**. The conditional branch node is added, and any existing nodes after the inserted branch node are moved to the default branch of the new conditional branch node.

2. To change the default name and add a description for the branch node, do the following:

    a. Click the **Conditional Branch** icon, then click **Edit Name and Annotation**.

    b. Change the name and description, as desired.

    c. Click **Save**.

3. To add branch definitions, click the **Conditional Branch** icon, then click **Edit Branch**. The **Edit Branch Node** page is displayed.

4. Do the following:

    a. In the **Selected Path** field, click **Edit** to add an XPath expression for specifying the path. See Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

    b. In the **Variable** field, enter a context variable.

    c. From the **Operator** field, select a comparison operator.

    d. In the **Value** field, enter a value for the branch.

    e. In the **Label** field, enter a label for the branch.

5. Optionally, under Options:

- Click **Add a New Branch** to add a new branch definition to this branch node.

- Click **Delete this Branch** to delete a branch definition.

- Click **Move Branch Up** or click **Move Branch Down** to change the positions of branch definitions. This option displays only when more than one branch definition exists.

6. Click **Save** to commit the updates in the current session.

7. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.6 Adding Operational Branches to Message Flows

When message flows define Web Services Description Language (WSDL)-based proxy services, operation-specific processing is required. Instead of configuring a branching node based on operations manually, Oracle Service Bus provides a minimal configuration branching node that automatically branches based on operations. In other words, when you create an operational branch node in a message flow, you can quickly build your branching logic based on the operations defined in the WSDL because the Oracle Service Bus Administration Console presents those operations in the branch node configuration page.

A branch node allows processing to proceed along exactly one of several possible paths. Branching is driven by an XPath-based switch table. Each branch in the table specifies a condition (for example, <500) that is evaluated in order down the message flow against a single XPath expression (for example, ./ns:PurchaseOrder/ns:totalCost on $body). Whichever condition is satisfied first determines which branch is followed. If no branch condition is satisfied, then the default branch is followed. A branch node may have several descendants in the message flow: one for each branch, including the default branch.

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add an operation branch to a message flow

1. Click a **Pipeline Pair Node** icon or a **Branch Node** icon, then click **Create Operational Branch**. The operational branch node is added, and any existing nodes after the inserted branch node are moved to the default branch of the new operational branch node.

2. To change the default name and add a description for the branch node, do the following:

   a. Click the **Operational Branch** icon, then click **Edit Name and Annotation**.

   b. Change the name and description, as desired.

   c. Click **Save**.

3. To add branch definitions, click the **Operational Branch** icon, then click **Edit Branch**. The Edit Branch Node page is displayed.

4. In the Operation Branch Definitions panel, select a service operation.

5. Optionally, under Options:

   ■ Click **Add a New Branch** to add a new branch definition to this branch node.

   ■ Click **Delete this Branch** to delete a branch definition.

   ■ Click **Move Branch Up** or click **Move Branch Down** to change the positions of branch definitions. This option displays only when more than one branch definition exists.

6. Click **Save**.

7. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.7 Adding Stages to Pipelines

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a stage to a Pipeline

1. If necessary, click the plus sign to the left of the **Pipeline Pair Node** icon to expand it. A pipeline pair contains a Request Pipeline and a Response Pipeline.

2. Click the pipeline to which you want to add the stage, then click **Add Stage**.

3. To change the default name and add a description for the stage, do the following:

   a. Click the **Stage** icon, then click **Edit Name and Annotation**.

   b. Change the name and description, as desired.

   c. Click **Save**.

4. To add actions to the stage, click the **Stage** icon, then click **Edit Stage**. See Section 21.1, "Adding and Editing Actions in Message Flows."

5. To add error handling to the stage, click the **Stage** icon, then click **Add Stage Error Handler**. See Section 24.4, "Adding Stage Error Handlers." The Edit Message Flow page is displayed.

6. Continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

7. Click **Save** to commit the updates in the current session.

8. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.8 Adding Route Nodes to Message Flows

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a Route Node to a Message Flow

1. Click the **Pipeline Pair Node** icon of a pipeline pair, then click **Add Route**.

2. To change the default name and add a description for the route node, click the **Route Node** icon, then click **Edit Name and Annotation**. Change the name and description, as desired, then click **Save**.

> **Note:** When you rename a pipeline or a route node, the number of messages displayed on the Dashboard page in the Monitoring module may not correlate with those of other components due to the pipeline counters being reset to zero. This is because Oracle Service Bus treats the rename as a delete and recreate action. The numbers should correlate again after a time period equal to the service's monitoring interval has elapsed.

3. To add actions to the route node, click the **Route Node** icon, then click **Edit Route**. The Edit Message Flow page is displayed. See the following sections for information about the actions you can add to route nodes:

   - Section 21.12, "Adding If... Then... Actions"
   - Section 21.8, "Adding Dynamic Routing to Route Nodes"
   - Section 21.9, "Adding Routing Actions to Route Nodes"
   - Section 21.10, "Adding Routing Tables to Route Nodes"
   - Section 24.5, "Adding Route Node Error Handlers"

4. On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

5. Click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

22.9 Cutting, Copying, and Pasting Stages and Route Nodes

You can cut, copy, and paste stages and route nodes.

- To cut a stage or a route node, click its icon and select **Cut** or **Copy**.

- To paste a stage that you cut or copied from a different pipeline pair within the message flow of this proxy service or from the message flow of a different proxy service, do one of the following:

  - Click the **Request Pipeline or Response Pipeline** icon, then click **Paste Stage**.

  - Click the **Stage** icon in a pipeline, then click **Paste**.

- To paste a route node that you cut or copied from the message flow of another proxy service, click the **Pipeline Pair Node** icon for the pipeline pair, then click **Paste Route**.

23

Proxy Services: XQuery and XPath Editors

This chapter describes how to use XQuery and XPath expressions in message flows.

In the message flow of a proxy service, you can assign XQuery expressions to message context variables, assign If...Then... actions based on the Boolean result of an XQuery expression, insert the result of an XQuery expression at an identified place relative to an XPath expression, specify the message context that you want to log through XQuery expressions on context variables, and so on.

The XQuery Expression Editor, the XQuery Condition Editor, and the XPath Expression Editor are available in the appropriate context in message flows to construct the kind of expression called for in the context.

23.1 Creating and Editing Inline XQuery and XPath Expressions

When you add actions to stages or route nodes on the Edit Stage Configuration page, a skeleton structure is displayed on the page that prompts for configuration details. Figure 23–1 shows an example.

Figure 23–1   Example of Action Configuration Skeleton

Whenever it is appropriate for the context, the skeleton provides links for accessing the XQuery editors, where you can construct expressions that will be executed inline, as required by the context in the action.

Before you begin

These instructions assume you are creating or editing an action in the Edit Stage Configuration page of a pipeline stage, an error handler stage, or a route node. See Section 22.1, "Viewing and Editing Message Flows." and Section 21.1, "Adding and Editing Actions in Message Flows."

To create or modify an inline expression

1. On the Edit Stage Configuration page, locate the place in the action where you want to add or edit the expression.

2. Click the expression link to open the editor that is appropriate for the context. When an expression has not yet been defined, the link tells what kind of expression you can use in that position:

- Click **Expression** to create an XQuery expression.You can also import an XQuery or XSLT resource created outside Oracle Service Bus, then bind it to the inline XQuery.

- Click **Condition** to create an XQuery conditional expression for an if...then... action.

- Click **XPath** to create an XPath expression for a message context variable.

When an expression has already been defined in a position, the **Expression**, **Condition**, or **XPath** link is replaced by a link that shows expression itself, for example `true()` icon, or a fragment of the expression it is too long to fit, for example `$body/urn:POSta...` icon. Click the expression (or expression fragment) to open the expression in the appropriate editor.

3. Build the expression, as described in the following topics:

- Section 23.3, "Building Expressions in the Editor Workspace Text Fields"

- Section 23.4, "Creating Namespaces to Use in Inline Expressions"

- Section 23.5, "Creating Variable Structures in the XQuery Editors"

- Section 23.6, "Creating Custom XPath Functions in the XQuery Editors"

- Section 23.7, "Binding External XQuery Resources to Inline XQueries"

- Section 23.8, "Binding External XSLT Resources to Inline XQueries"

- Section 23.9, "Binding Dynamic XQuery Expressions to Inline XQueries"

- Section 23.10, "Entering XQuery Comparison Expressions Using the Builder Option"

- Section 23.11, "Entering Unary Expressions Using the Builder Option"

4. Optionally, do either or both of the following:

- Click **Validate** to validate the expression.

- Click **Test** to test the expression. See Section 33.3, "Performing XQuery Testing."

5. Click **Save** to close the editor and insert the expression in the action.

After you finish

Continue configuring the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

23.2 Understanding XQuery Editor Layouts and Tasks

The XQuery Expression Editor, the XQuery Condition Editor, and the XPath Expression Editor are each composed of the following components:

- Section 23.2.1, "Palettes"

- Section 23.2.2, "Workspace"

- Section 23.2.3, "Property Inspector"

23.2.1 Palettes

The left panel of the each editor contains the palettes listed below. In any of the editors, click the name of a palette to display it. Each palette contains entities that you can insert into expressions in the editors.

- The Namespace Definitions palette lists default Oracle Service Bus namespaces, variable namespaces, and user-defined namespaces. You can define new namespaces, which are then added to the list of user-defined namespaces.

  For more information, see:

  - Section 23.3, "Building Expressions in the Editor Workspace Text Fields"
  - Section 23.4, "Creating Namespaces to Use in Inline Expressions"

- The XQuery Functions palette lists a set of standard XQuery functions.When you insert a function into an expression, placeholders are used for parameter values you must supply.

  For more information, see:

  - Section 23.3, "Building Expressions in the Editor Workspace Text Fields"
  - Section 23.6, "Creating Custom XPath Functions in the XQuery Editors"
  - Section 23.7, "Binding External XQuery Resources to Inline XQueries"

- The Variable Structures palette provides a set of tools for inserting variables and paths to the variables, using XPath expressions.

  Variable structures are graphical representations of variables or variable paths that are displayed in the editor. They can help you visualize the variable structure, and you can use them to construct inline XQuery expressions that reference the content of the variable.

  > **Note:** Variable structures do not create variables. Variables are created at runtime as the target of the Assign action in the stage.

  Oracle Service Bus provides several predefined message context variables (`attachments`, `body`, `header`, `outbound`, and `inbound`), whose contents you can display as variable structures. You can also define your own variable structures.

  For more information, see

  - Section 23.3, "Building Expressions in the Editor Workspace Text Fields"
  - Section 23.5, "Creating Variable Structures in the XQuery Editors"

23.2.2 Workspace

The right side of the page provides a workspace for constructing the XQuery expression, XQuery condition, or XPath. The workspace is different in the three editors.

23.2.3 Property Inspector

In all three editors, the Property Inspector is displayed on the bottom right of the page. When you select an item from one of the palettes to add to the expression, that item appears in the Property Inspector. You can then paste the item into the workspace. See Section 23.3, "Building Expressions in the Editor Workspace Text Fields."

23.3 Building Expressions in the Editor Workspace Text Fields

The XQuery Expression Editor, the XQuery Condition Editor, and the XPath Expression Editor all provide text fields in which you can build expressions by typing directly or by pasting items from the palettes.

Before you begin

These instructions assume you are creating or editing an expression in the XQuery Expression Editor, XQuery Condition Editor, or XPath Expression Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To build an expression in a text field:

1. Display the panel containing the text field. Depending on the editor, do one of the following:

    - In the XQuery Expression Editor, select **XQuery Text** (located under the workspace button bar), if it is not already selected.

    - In the XQuery Condition Editor, select **Text** (located under the workspace button bar), if it is not already selected.

    - In the XPath Expression Editor, you do not have to select anything, because there are no options for selecting other tools.

    > **Note:** Selecting any of the above links displays a text field where you can create a complete expression appropriate for the context. However, the tools and techniques described in this topic can be used wherever text fields are provided in the editors, for example when binding variables from imported resources to the inline expression, as described in Section 23.7, "Binding External XQuery Resources to Inline XQueries" and Section 23.8, "Binding External XSLT Resources to Inline XQueries."

2. If desired, type or paste an expression or expression fragment into the field. If you create the complete expression this way, skip to step 7, below. Otherwise, proceed to the next step.

3. Select the palette containing the item(s) you want to add to the expression, and locate the item you want to add, as described in Table 23–1, below.

Table 23–1 Palettes

Palette	Description and Use
Namespace Definitions	Lists default Oracle Service Bus namespaces, variable namespaces, and user-defined namespaces. Namespace abbreviations are listed when defined.
	Scroll through the lists to find the desired namespace.
	You can also define a namespace. See Section 23.4, "Creating Namespaces to Use in Inline Expressions."

Table 23–1   (Cont.) Palettes

Palette	Description and Use
XQuery Functions	Contains a set of standard XQuery functions, organized alphabetically and by type, including any custom functions you have developed. Expand or collapse nodes in the tree. To do this, click the plus sign (+) or minus sign (-).
	See also:
	■ Section 11, "XQuery Transformations"
	■ "XQuery Implementation" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
Variable Structures	Displays variables and their contents as trees, which can help you to visualize.
	To display a variable structure and its contents
	■ Select the name of the structure from the list at the top of the palette. The list displays Built-in message context variables (`attachments`, `body`, `header`, `outbound`, and `inbound`), as well as any user-defined structures, organized by type (XML Type, Service Interface, and Simple Type).
	■ Expand or collapse nodes in the tree. To do this, click the plus sign (+) or minus sign (-).
	You can also define your own variable structures. See Section 23.5, "Creating Variable Structures in the XQuery Editors."
	Variable structures do not create variables. Variables are created at runtime as the target of the Assign action in the stage.
	When you insert an item from the a variable structure tree into the text field, it is inserted as an XPath expression that describes the path.
	See also Section 11, "XQuery Transformations."

4. Paste the desired item into the text field using any of the methods shown below in Table 23–2:

Table 23–2   Ways to Paste Items Into the Editor Text Fields

From this palette...	Do this...
Namespace Definitions palette	Use standard mouse or keyboard select, copy, and paste a namespace, for example:
	1. Select the entire namespace string (or its abbreviation, if one exists) by dragging the mouse pointer over the string.
	2. Press **Ctrl-C** to copy the string.
	3. Click the location in the text field where you want to insert the namespace.
	4. Press **Ctrl-V** to paste the string.
XQuery Functions palette or Variable Structures palette	Drag an item from the palette to the text field.
	Note: Dragging from the palette to the workspace is supported only in Internet Explorer.

Table 23–2   (Cont.) Ways to Paste Items Into the Editor Text Fields

From this palette...	Do this...
XQuery Functions palette or Variable Structures palette	1. Click an item in the palette. The item is displayed in the Property Inspector pane: Functions are displayed with placeholders for any values you have to supply. Variables and their attributes are displayed as XPath expressions. 2. Click in the text field where you want to insert the item 3. Click **Copy Property** to paste the item into the location selected in the text field.
XQuery Functions palette or Variable Structures palette	1. Click an item in the palette. The item is displayed in the Property Inspector pane. 2. Select and copy the item in the Property Inspector, using standard keyboard or mouse actions. 3. Select a location in the text field, and paste the item into the text field, using standard keyboard or mouse actions.

5. Continue to drag and drop functions to build the desired expression.

6. Edit the expression in the text field, as needed.

7. Optionally, do either or both of the following:

   - Click **Validate**. A message is displayed if the expression is validated successfully.

   - Click **Test** to test the expression. See Section 33.3, "Performing XQuery Testing."

8. Click **Save** to close the editor and insert the expression in the action.

After you finish

Continue configuring the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

23.4 Creating Namespaces to Use in Inline Expressions

The Namespace Definitions palette includes a list of default namespaces, but you can also define new ones.

Before you begin

These instructions assume you are creating or editing an expression in the XQuery Expression Editor, XQuery Condition Editor, or XPath Expression Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To create and use a namespace in an inline expression

1. Select **Namespace Definitions**. The Namespace Definitions palette includes a list of default namespaces, plus lists of variable namespaces and user defined namespaces, if any exist.

2. To define and add a user namespace,

   a. Click **Add Namespace**.

   b. In the **Prefix** field, enter a unique identifier for the namespace. You cannot use the same prefix more than once.

    c. In the **URI** field, enter a URL for this namespace in the format `http://url/.../` or enter a URN in the format `uddi:server:`.

    d. Click **Add** to add the namespace to the User Defined Namespaces list.

    e. Copy and paste the user-defined namespace into the XQuery expression, XQuery condition, or XPath, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

After you finish

Continue as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

23.5 Creating Variable Structures in the XQuery Editors

The Variable Structures palette in the XQuery and XPath editors displays graphical representations of the contents of variables. It includes by default the built-in message context variables `attachments`, `body`, `header`, `outbound`, and `inbound`.

Each variable structure mapping entry has a label and maps a variable or variable path to one or more structures. The scope of these mappings is a stage or a route node.

You can also declare your own variable structures, based on:

- XML types, including:
  - Schema elements
  - WSDL elements
  - Schema types
  - WSDL types
  - MFLs
- Service interfaces
- Simple types (string or any XML)

You can use this feature directly for all user-defined variables, as well as `$inbound`, `$outbound`, and `$fault`. However, you cannot use it directly to access XML attachments in `$attachments`, headers in `$header`, or documents and RPC parameters in `$body`, with one exception— you can use it directly to access documents and parameters in `$body` for request messages received by a WSDL proxy service.

Before you begin

These instructions assume you are creating or editing an expression in the XQuery Expression Editor, XQuery Condition Editor, or XPath Expression Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To create a variable structure:

1. Select **Variable Structures**.

2. In the Variable Structures palette, click **Add New Structure**.

3. Continue with any of the tasks listed in Table 23–3.

Table 23–3   Create a New Variable Structure

To...	Complete these steps...
Create a variable structure that maps a variable to an XML Schema type	1. Select **XML Type** at the top of the Variable Structures palette, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. The path must begin with $. 4. Under the **Type** field, select the appropriate option, then select **Schema Type**. 5. Click **Browse**. The XML Schema Browser is displayed. Select an XML Schema from the list, select an XML Schema type from the Definitions pane, then click **Submit**. 6. Click **Add** to create the variable structure.
Create a variable structure that maps a variable to a WSDL type	1. Select **XML Type** at the top of the Variable Structures palette, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. 4. Under the **Type** field, select the appropriate option, then select **WSDL Type**. 5. Click **Browse**. The WSDL Browser is displayed. Select a WSDL from the list of WSDLs, select a WSDL type from the Definitions pane, then click **Submit**. 6. Click **Add** to create the variable structure.
Create a variable structure that maps a variable to an XML Schema element	1. At the top of the Variable Structures palette, select **XML Type**, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. 4. Under the **Type** field, select the appropriate option, then select **Schema Element**. 5. Click **Browse**. The XML Schema Browser is displayed. Select an XML Schema from the list, select an XML Schema type from the Definitions pane, then click **Submit**. 6. Click **Add** to create the variable structure.

Table 23–3   (Cont.) Create a New Variable Structure

To...	Complete these steps...
Create a variable structure that maps a variable to a WSDL element	1. At the top of the Variable Structures palette, select **XML Type**, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. 4. Under the **Type** field, select the appropriate option, then select **WSDL Element**. 5. Click **Browse**. The WSDL Browser is displayed. Select a WSDL from the list of WSDLs, select a WSDL element from the Definitions pane, then click **Submit**. 6. Click **Add** to create the variable structure.
Create a variable structure that maps a variable to a child element	1. At the top of the Variable Structures palette, select **XML Type**, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. 4. Under the **Type** field, select the type of variable you want to create: To create an XML Schema Element or WSDL Element variable, select the option associated with this option, then select **Schema Element** or **WSDL Element**. To create an MFL variable, select the option associated with this option, then select **MFL**. 5. For the XML Schema, WSDL, or MFL, click **Browse** to select an object from the list that the browser displays, then click **Submit**. For example, select an MFL from a list of MFLs, then click **Submit**. 6. Select the **Set as child** check box to set the element as a child of the structure being created. 7. Click **Add** to create the variable structure.
Create a variable structure that uses an MFL resource	1. At the top of the Variable Structures palette, select **XML Type**, if it is not already selected. 2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime. 3. In the **Structure Path** field, enter the path of the variable structure at runtime. 4. Under the **Type** field, select the appropriate option, then click **Browse**. The MFL Browser is displayed. 5. Select an MFL from the list of MFLs, then click **Submit**. 6. Click **Add** to create the variable structure.

Table 23–3   (Cont.) Create a New Variable Structure

To...	Complete these steps...
Create a Service Interface variable structure	1. At the top of the Variable Structures palette, select **Service Interface**.
	2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime.
	3. In the **Structure Path** field, the default is already set as **$body**. You cannot change this field.
	4. In the **WSDL Based Service** field, select the **Service Browser** icon, select a service from the list of services the Service Browser displays, then click **Submit**.
	The service you selected is displayed in the WSDL Based Service field.
	1. In the **Operation** field, select an operation or select **None** to not include an operation.
	2. Click **Add** to create the variable.
Create a Simple variable structure	1. At the top of the Variable Structures palette, select **Simple Type**.
	2. In the **Structure Label** field, enter a display name for the variable you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime.
	3. In the **Structure Name** field, enter a name for the variable structure you want to create.
	4. Under the **Type** field, select **String** or **Any XML**.
	5. Click **Add** to create the variable.

After you finish

Continue as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

23.6 Creating Custom XPath Functions in the XQuery Editors

Creating custom XQuery functions is a development process that requires writing Java code. For more information, see "Creating and Using Custom XPath Functions" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

23.7 Binding External XQuery Resources to Inline XQueries

You can bind XQuery resources to inline XQuery expressions, so they will be executed inline as part of an action.

Before you begin

These instructions assume you are creating or editing an expression in the XQuery Expression Editor, XQuery Condition Editor, or XPath Expression Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To bind an XQuery Resource to an inline expression:

1. Click **Variable Structures**.

2. In the workspace (under the button bar), select **XQuery Resources**.

3. In the **1. Select an XQuery resource to execute** box, click **Browse**.

4. In the XQuery Browser, select the option associated with the XQuery you want to use, then click **Submit**.

5. In the **2. Bind Variables** box, define the input parameters for the transformation. For each variable listed under **Variable Name** enter an XQuery expression to be mapped to it. You must define a mapping for each parameter. For example, if an XQuery transformation has two input parameters named **one** and **two**, the **Variable Name** field has two labels—**one** and **two**. A text box, into which the XQuery expression is entered, is associated with each label.

   The following XQuery expressions are examples of valid input to this field:

   ```
   $body/*[1]
   $body/po:PurchaseOrder
   ```

   > **Note:** The following variable name is not a valid entry for this field and results in an exception: `body`.

6. After you finish, continue with any of the following tasks.

   - Click **Validate**. A message is displayed if the expression is validated successfully.

   - Click **Test**. See Section 33.3, "Performing XQuery Testing."

   - Save or discard your changes.

23.8 Binding External XSLT Resources to Inline XQueries

The XQuery Expression Editor page lets you select an XSLT resource for execution. To learn more about this editor, see Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To Select an XSLT Resource for Execution

1. Select the **XSLT Resources** option.

2. Under the **Select the XSLT resource to execute** field, select the **XSLT Browser** icon.

3. In the XSLT Browser, select the option associated with the XSLT you want to execute, then click **Submit**.

4. Under the **Bind Variables** field, a label and a corresponding text box is displayed for each input parameter of the transformation. Each label corresponds to the name of a parameter, and each text box is for defining an XQuery expression to be mapped to the parameter. You must define a mapping for each parameter. For example, if an XSL transformation has two input parameters named **one** and **two**, the **Variable Mapping** field has two labels—**one** and **two**—with a text box associated with each into which the XQuery expression is entered. In addition to the mapping for any input variables, you must also specify an XQuery expression for the Input Document to the transformation. The mapping is specified in the text box with the label Input Document.

   The following XQuery expressions are examples of valid input to this field:

   ```
   $body/*[1]
   $body/po:PurchaseOrder
   ```

> **Note:** The following variable name is not a valid entry for this field and results in an exception: `body`.

5. Continue with any of the following tasks:

   - Section 23.4, "Creating Namespaces to Use in Inline Expressions"

   - Section 23.5, "Creating Variable Structures in the XQuery Editors"

   - Section 23.3, "Building Expressions in the Editor Workspace Text Fields"

   - Section 23.7, "Binding External XQuery Resources to Inline XQueries"

   - Click **Validate**. A message is displayed if the expression is validated successfully.

   - Click **Test**. See Section 33.3, "Performing XQuery Testing."

23.9 Binding Dynamic XQuery Expressions to Inline XQueries

The XQuery Expression Editor page lets you specify a dynamic XQuery expression that evaluates at runtime to the name of a pre-registered XQuery resource. To learn more about this editor, see Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To define a dynamic XQuery expression

1. Select the **Dynamic XQuery** option.

2. In the Enter Expression for XQuery Resource area, enter the XQuery expression that will evaluate at runtime to the name of a pre-registered XQuery resource.

   The following shows the syntax for the XQuery resource (representing the full name of the resource):

   Project/folder1/folder2/XQueryResourceName

3. In the Bind Variables using XQuery Template of Custom Variables area, define the input parameters for the transformation.

   - Click Browse in the **Select XQuery Template** field to select an existing registered resource to serve as a template for the shape of the query (the number and names of the variables). After selecting a template, the variables appear in the Bind Variables area. Note that the template is not persisted with the configuration. Instead, the template serves as a quick start to help you specify the variables for the query.

   - Type a variable name in the **Add Custom Variable** field, and click Add. For each variable listed under Variable Name, enter an XQuery expression to be mapped to it. You must define a mapping for each parameter. For example, if an XQuery transformation has two input parameters named **one** and **two**, the **Variable Name** field has two labels—**one** and **two**. A text box, into which the XQuery expression is entered, is associated with each label.

   The following XQuery expressions are examples of valid input to the variable fields:

   ```
   $body/*[1]
   $body/po:PurchaseOrder
   ```

4. After you finish, continue with any of the following tasks.

- Click **Validate**. A message is displayed if the expression is validated successfully.

- Click **Test**. See Section 33.3, "Performing XQuery Testing."

- Save or discard your changes.

23.10 Entering XQuery Comparison Expressions Using the Builder Option

Before you begin

These instructions assume you are creating or editing an XQuery conditional expression in the XQuery Condition Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To enter an XQuery comparison expression:

1. In the XQuery Condition Editor, select **Builder** (located under the workspace button bar), if it is not already selected. option.

2. In the Expression Builder box, select **Comparison Expression** if it is not already selected.

3. In the **Operand** field, enter a context variable, namespace definition, or XQuery function.

   To build the operand, you can paste XQuery functions from the XQuery Functions palette and namespaces from the Namespace Definitions palette. See Section 23.3, "Building Expressions in the Editor Workspace Text Fields."

4. From the **Operator** list, select a comparison operator.

5. In the **Value** field, enter text or enter a context variable.

   You must enter text in quotations—for example, `"true"` is valid; `true` is not.

6. Click **Add**. The text you entered is displayed in the Expressions pane.

7. Repeat steps 3-6 to build additional conditions. Each condition is added to the end of the list of conditions.

   Consider the following when using multiple conditions:

   - When you build additional expressions, make sure to select the **And** or the **Or** options in the **Conjunction** field.

   - You can select a condition and click the **Up** arrow to move it up in the list of conditions or click the **Down** arrow to move it down the list of conditions. You can also click the **Edit** icon to update a condition, or click the **Delete** icon to delete it.

   - Unary expressions may be intermixed with Comparison expressions in the overall definition of a condition.

8. Optionally, do either or both of the following:

   - Click **Validate**. A message is displayed if the expression is validated successfully.

   - Click **Test** to test the expression. See Section 33.3, "Performing XQuery Testing."

9. Click **Save** to close the editor and insert the expression in the action.

After you finish

Continue configuring the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

23.11 Entering Unary Expressions Using the Builder Option

Before you begin

These instructions assume you are creating or editing an XQuery conditional expression in the XQuery Condition Editor, as described in Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

To enter an XQuery comparison expression:

1. In the XQuery Condition Editor, select **Builder** (located under the workspace button bar), if it is not already selected. option.

2. In the Expression Builder box, select **Unary Expression** if it is not already selected.

3. Select the **Not** check box to make this a negative expression, or leave it blank.

4. Enter a context variable, namespace definition or XQuery function in the **Expression** field.

   To build the expression, you can paste XQuery functions from the XQuery Functions palette and namespaces from the Namespace Definitions palette. See Section 23.3, "Building Expressions in the Editor Workspace Text Fields."

5. Click **Add** to add the text to the Expressions pane.

6. Repeat steps 3-5 to build additional conditions. Each condition is added to the end of the list of conditions.

   Consider the following when building unary expressions.

   - When you build additional expressions, make sure to select the **And** or the **Or** options in the **Conjunction** field.

   - You can select a condition and click the **Up** arrow to move it up in the list of conditions, click the **Down** arrow to move it down the list of conditions, click the **Edit** icon to update it, or click the **Delete** icon to delete it.

   - Unary expressions may be intermixed with Comparison expressions in the overall definition of a condition.

7. Optionally, do either or both of the following:

   - Click **Validate**. A message is displayed if the expression is validated successfully.

   - Click **Test** to test the expression. See Section 33.3, "Performing XQuery Testing."

8. Click **Save** to close the editor and insert the expression in the action.

After you finish

Continue configuring the action, as described in Section 21.1, "Adding and Editing Actions in Message Flows."

24

Proxy Services: Error Handlers

This chapter describes how to implement error handling in message flows.

You can configure error handling at the message flow, pipeline, route node, and stage level.

24.1 Edit Error Handler Page

You can configure error handling at the message flow, pipeline, route node, and stage level.

Configure error handlers on the Edit Error Handler page. You must always add at least one stage to the page to specify how the error handler will work. See the following topics:

- Section 24.2, "Adding Proxy Service Error Handlers"
- Section 24.3, "Adding Pipeline Error Handlers"
- Section 24.4, "Adding Stage Error Handlers"
- Section 24.5, "Adding Route Node Error Handlers"
- Section 24.6, "Editing Error Handlers"
- "Handling Errors in Message Flows" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

24.2 Adding Proxy Service Error Handlers

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To add a proxy service Error Handler

1. Click the **Proxy Service** icon, then click **Add Service Error Handler**. The Edit Error Handler page is displayed.

2. Click the **Error Handler** icon, then click **Add Stage**.

3. Click the **Stage** icon, then click **Edit Stage**. The Edit Stage Configuration page is displayed.

4. Click **Add an Action**, then select the action you want to add.

   An error handler is a pipeline and is therefore configured like any other pipeline. For example, you can use the Publish action to send error notifications to other

services, use the Assign action to modify the context variables, and so on. See Section 21.1, "Adding and Editing Actions in Message Flows." There is no restriction on what actions may be chained together.

Three commonly-used error actions are Raise Error, Reply, and Resume.

5. Add other actions and make other edits on the Edit Stage Configuration page, as desired.

6. On the Edit Stage Configuration page, click **Save** to commit the updates in the current session.

7. On the Edit Error Handler page, click **Save** to commit the updates in the current session.

After you finish

On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

24.3 Adding Pipeline Error Handlers

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

The instructions also assume you have created a pipeline pair node, as explained in Section 22.4, "Adding Pipeline Pair Nodes to Message Flows."

To add a Pipeline Error Handler

1. Navigate to the pipeline pair node containing the pipeline to which you want to add an error handler. If the pipeline pair is not already expanded, click the plus sign next to the icon to display the pipelines.

2. Click the **Request Pipeline** icon or the **Response Pipeline** icon, then click **Add Pipeline Error Handler**. The Edit Error Handler page is displayed.

3. Click the **Error Handler** icon, then click **Add Stage**.

4. Click the **Stage** icon, click **Edit Stage**. The Edit Stage Configuration page is displayed.

5. Click **Add an Action**, then select the action you want to add.

   An error handler is a pipeline and is therefore configured like any other pipeline. For example, you can use the Publish action to send error notifications to other services, use the Assign action to modify the context variables, and so on. To learn more about the type of action you want to add, see the appropriate procedure in Section 21.1, "Adding and Editing Actions in Message Flows." There is no restriction on what actions may be chained together.

   Three commonly-used error actions are Raise Error, Reply, and Resume.

6. Add other actions and make other edits on the Edit Stage Configuration page, as desired.

7. On the Edit Stage Configuration page, click **Save** to commit the updates in the current session.

8. On the Edit Error Handler page, click **Save** to commit the updates in the current session.

After you finish

On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

24.4 Adding Stage Error Handlers

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

The instructions also assume you have added a stage to a pipeline, as explained in Section 22.7, "Adding Stages to Pipelines."

To add a Stage Error Handler

1. Navigate to the stage to which you want to add error handling.

2. Click the **Stage** icon, then click **Add Stage Error Handler**. The Edit Error Handler page is displayed.

3. Click the **Error Handler** icon, then click **Add Stage**.

4. Click the **Stage** icon, then click **Edit Stage**. The Edit Stage Configuration page is displayed.

5. Click **Add an Action**, then select the action you want to add.

    An error handler is a pipeline and is therefore configured like any other pipeline. For example, you can use the Publish action to send error notifications to other services, use the Assign action to modify the context variables, and so on. To learn more about the type of action you want to add, see the appropriate procedure in Section 21.1, "Adding and Editing Actions in Message Flows." There is no restriction on what actions may be chained together.

    Three commonly-used error actions are Raise Error, Reply, and Resume.

6. Add other actions and make other edits on the Edit Stage Configuration page, as desired.

7. Click **Save** to commit the updates in the current session.

After you finish

On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

24.5 Adding Route Node Error Handlers

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

The instructions also assume you have created a route node, as explained in Section 22.8, "Adding Route Nodes to Message Flows."

To add a Route Node Handler

1. Click the **Route Node** icon, then click **Add Error Handler**. The Edit Error Handler page is displayed.

2. Click the **Error Handler** icon, then click **Add Stage**.

3. Click the **Stage** icon, then click **Edit Stage**. The Edit Stage Configuration page is displayed.

4. Click **Add an Action**, then select the action you want to add.

   Since an error handler is another pipeline, it is configured like any other pipeline. For example, the Publish action may be used to send error notifications to other services, the Assign action may be used to modify the context variables, and so on. To learn more about the type of action you want to add, see the appropriate procedure in Section 21.1, "Adding and Editing Actions in Message Flows." There is no restriction on what actions may be chained together.

   Three commonly-used error actions are Raise Error, Reply, and Resume.

5. Add other actions and make other edits on the Edit Stage Configuration page, as desired.

6. On the Edit Stage Configuration page, click **Save** to commit the updates in the current session.

7. On the Edit Error Handler page, click **Save** to commit the updates in the current session.

After you finish

On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

24.6 Editing Error Handlers

Before you begin

These instructions assume you are already editing a message flow in the Edit Message Flow page, as explained in Section 22.1, "Viewing and Editing Message Flows."

To view and change an Error Handler

Do one of the following:

Table 24–1   Viewing and Changing the Error Handler

To...	Complete This Step...
View and change the proxy service message flow error handler	Click the **Proxy Service** icon, then click **Edit Service Error Handler**. The Edit Error Handler page is displayed. See Section 24.2, "Adding Proxy Service Error Handlers."
View and change the pipeline error handler	Click the appropriate **Request Pipeline** icon or the **Response Pipeline** icon, then click **Edit Pipeline Error Handler**. The Edit Error Handler page is displayed. See Section 24.3, "Adding Pipeline Error Handlers."
View and change the route node error handler	Click the appropriate **Route Node** icon, then click **Edit Route Error Handler**. The Edit Error Handler page is displayed. See Section 24.5, "Adding Route Node Error Handlers."
View and change the stage error handler	Click the appropriate **Stage** icon, then click **Edit Stage Error Handler**. The Edit Error Handler page is displayed. See Section 24.4, "Adding Stage Error Handlers."

After you finish

On the Edit Message Flow page, continue to construct the message flow, as described in Section 22.1, "Viewing and Editing Message Flows."

25

Security Configuration

This chapter describes how to create users, groups, and roles for use in Oracle Service Bus inbound security and administrative security.

Inbound transport-level security and message-level security use the user, group, and role data to authenticate inbound client requests. It applies access control policies to determine which authenticated users are authorized to use proxy services and business services.

Administrative security uses the user, group, and role data to determine which authenticated users are authorized to create or modify Oracle Service Bus configuration data or to monitor Oracle Service Bus performance. For more information, see Section 25.1, "Understanding Users, Groups, Security Roles and Policies.". See also "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

> **Note:** You cannot export users, groups, or roles when you export a configuration because these objects are located in security provider stores. You must create these objects again when you import the exported configuration or use WebLogic Server tools (if available) to export and import them.

25.1 Understanding Users, Groups, Security Roles and Policies

This section includes the following topics:

- Section 25.1.1, "Users"
- Section 25.1.2, "Groups"
- Section 25.1.3, "Roles"
- Section 25.1.4, "Access Control Policies"
- Section 25.1.5, "Security Configuration Data and Sessions"

25.1.1 Users

Users are entities that can be authenticated. A user can be a person or a software entity, such as a Web Services client. You must give each user a unique identity (name) within a security realm.

Typically, the users that you create fall into two categories:

- Client users who can access your proxy services or business services.

If you create a large number of client users, consider organizing them into security groups.

■ Administrative users who can use the Oracle Service Bus Administration Console to create or modify proxy services, business services, and other Oracle Service Bus resources.

Oracle Service Bus uses role-based security for its administrative functions. Instead of giving access privileges directly to users, Oracle Service Bus gives administrative privileges only to security roles. To give administrative privileges to a user, you place the user in one of the default security groups, which is in one of the pre-defined security roles.

25.1.2 Groups

To facilitate administering a large number of users, you can organize users into named groups. Then, instead of giving access privileges or role identities to individual users, you give privileges or identities to groups.

25.1.2.1 Administrative Security Groups

Oracle Service Bus provides default security groups to facilitate giving users access to administrative functions such as creating proxy services. Each group is in one of the pre-defined Oracle Service Bus security roles that have been granted administrative privileges.

For more information, see "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

25.1.3 Roles

A security role is an identity that can be granted to a user or group based on conditions in the runtime environment. When you create access control policies, you can grant access to a role, group, or user.

For example, you can create two of your groups, MyCustomersEast and MyCustomersWest. You create a security role named PrivilegedCustomer and create conditions so that the MyCustomersWest group is in the role from 8am to 8pm EST, while the MyCustomersEast group is in the role from 8pm to 8am EST. Then you create an access control policy for a proxy service that gives the PrivilegedCustomer role access to the service. Different users will have access at different times depending on whether they are in the MyCustomersEast and MyCustomersWest group.

25.1.3.1 Administrative Security Roles

Oracle Service Bus provides four, pre-defined security roles (plus four pre-defined roles from WebLogic Server) that give administrative privileges. You cannot change the access privileges for the Oracle Service Bus administrative security roles, but you can change the conditions under which a user or group is in one of the roles.

For more information about these roles and the privileges available for each role, see "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

25.1.4 Access Control Policies

An access control policy specifies conditions under which users, groups, or roles can access a proxy service. For example, you can create a policy that always allows users in

the GoldCustomer role to access a proxy service and that allows users in the SilverCustomer role to access the proxy service only after 12pm on weeknights.

For all proxy services, you can create a transport-level policy, which applies a security check when a client attempts to establish a connection with the proxy service. Only requests from users who are listed in the transport-level policy are allowed to proceed.

A message-level access control policy applies a security check when a client attempts to invoke a proxy service with message-level security. You can create a message-level access control policy in the following cases:

- For proxy services that are active Web Service security intermediaries
- For proxy services that have message level custom authentication

Only users who are listed in the message-level policy are allowed to invoke the operation.

25.1.5 Security Configuration Data and Sessions

Users, groups, and roles are persisted in security providers, which are not governed by Oracle Service Bus sessions. Therefore, you can create or modify this data when you are in or out of a session. Any additions or modifications to this data take effect immediately and are available to all sessions. If you discard a session in which you added or modified the data, the security data is not discarded.

Access control policies are persisted in authorization providers. And there is now a reference to them in the Oracle Service Bus repository.

Access control policies are managed within an Oracle Service Bus design session and not outside the session. Because the changes are made within a session, you can commit or discard the changes as with other resources.

Although ACLs can be managed from the Oracle Service Bus Administration Console, you can change policies outside Oracle Service Bus. However, changing policies outside of Oracle Service Bus can make the reference in Oracle Service Bus out-of-date and invalid.

Therefore, for consistent management, either completely manage ACLs outside of Oracle Service Bus sessions (using the authorization provider MBeans or third-party authorization provider tools) or completely manage them from within Oracle Service Bus sessions. Any combination of the two approaches can result in an inconsistent view of policies.

25.2 Locating Users

To locate users:

1. Select **Security Configuration** > **Users**. The Summary of Users page displays the information shown in Table 25–1. For a more detailed description of the properties, see Section 25.4, "Editing Users."

2. To restrict the number of items in the list or locate a specific user, you can filter by user name. Enter part or all of the user name in the **Name** field and click **Search**.

   You can use the asterisk (*) wildcard character. (Other wildcard characters are not supported.)

   Click **View All** to remove the search filters and display all users.

Table 25–1   Users Information

Property	Description
User Name	The name assigned to the user. The name is a link to the View User Details page. See Section 25.4, "Editing Users."
Group Membership	The name of the group to which this user belongs. The name is a link to the View Group Details page. See Section 25.8, "Editing Groups."
Authentication Provider	The authentication provider for this user.
Options	Click the **Delete** icon to delete a specific user. See Section 25.5, "Deleting Users."

25.3 Adding Users

To add users:

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** to display the Summary of Users page.

3. Click **Add New** to display the Create a New User - General Configuration page.

    You can add a user from inside or outside a session.

4. In the **User Name** field, enter a unique name. This is a required field.

5. In the **Password** field, enter a password. This is a required field.

    > **Note:**   Authentication providers can impose a minimum password length. For a user defined in the WebLogic Authentication provider, the default minimum password length is 8 characters. You can customize this setting using the WebLogic Server Administration Console. (The WebLogic Authentication provider is configured in the default security realm with the name DefaultAuthenticator.)

6. In the **Confirm Password** field, enter the same password you entered for the **Password** field. This is a required field.

7. In the **Authentication Provider** field, select the authentication provider for this user.

    If multiple authentication providers are configured in the security realm, they will appear in the list. Select the authentication provider database that should store information for the new user.

8. In the **Group Membership** field, select a group for this user.

    a. Select a group from the **Available Groups** field.

    b. Click the arrow to move the group into the **Current Groups** field.

9. Click **Save** to create the user.

    Oracle Service Bus Administration Console saves the user and the user becomes available immediately to all sessions. If you are in a session when you add the user and then you discard the session, Oracle Service Bus Administration Console does not delete the new user.

25.4 Editing Users

Use the View User Details page to view and change details of a specific user.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Locate the user. See Section 25.2, "Locating Users."

3. Click the user name. The View User Details page displays the information shown in Table 25–2.

Table 25–2   User Details

Property	Description
User Name	The name of this user
Authentication Provider	The authentication provider that contains this user definition.
Group Membership	The name of the group to which this user belongs.

4. To edit the user details, click **Reconfigure** to display the Edit User Details page.

   You can edit user details from inside or outside a session.

5. Make the appropriate changes to the **New Password**, **Confirm Password**, and **Group Membership** fields. See Section 25.3, "Adding Users" for descriptions of the fields.

   You cannot change the **User Name** field.

6. Click **Save Changes** to update the user.

   Oracle Service Bus Administration Console updates the user details and the update becomes available immediately to all sessions. If you are in a session when you update the user and then you discard the session, Oracle Service Bus Administration Console does not delete the updates.

25.5 Deleting Users

Use the Summary of Users page to delete a selected user or multiple users.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** to display the Summary of Users page.

3. Select the user you want to delete. You can select multiple users if necessary.

   You can delete a user from inside or outside a session.

4. Click **Delete**. A message prompting you to confirm that you want to delete the user is displayed.

5. To delete the user, click **OK**.

   Oracle Service Bus Administration Console deletes the user. If you are in a session when you delete the user and then you discard the session, Oracle Service Bus Administration Console does not delete the updates.

6. Alternatively, you can click the **Delete** icon in the Options column of the user you want to delete.

25.6 Locating Groups

To locate groups:

1. Select **Security Configuration** > **Groups**. The Summary of Groups page displays the information shown in Table 25–3. For a more detailed description of the properties, see Section 25.8, "Editing Groups."

2. To restrict the number of items in the list or locate a specific group, you can filter by group name. Enter part or all of the group name in the **Name** field and click **Search**.

You can use the asterisk (*) wildcard character. (Other wildcard characters are not supported.)

Click **View All** to remove the search filters and display all groups.

Table 25–3 Group Information

Property	Description
Group Name	The name of the group. The name is a link to the View Group Details page. See Section 25.8, "Editing Groups."
Group Membership	The group to which this group belongs. The name is a link to the View Group Details page. See Section 25.8, "Editing Groups."
Authentication Provider	The authentication provider that contains this group definition.
Delete	Click the **Delete** icon to delete a specific group. See Section 25.9, "Deleting Groups."

25.7 Adding Groups

To add groups:

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** > **Groups** to display the Summary of Groups page.

3. Click **Add New**.

You can add a group from inside or outside a session.

4. In the **Group Name** field, enter a unique name.

Do not enter spaces or special characters. This is a required field.

5. In the **Authentication Provider** field, select the authentication provider.

6. In the **Group Membership** field, select a group to which this group belongs.

a. Select a group from the **Available Groups** field.

b. Click the arrow to move the group into the **Current Groups** field.

7. Click **Save** to create the group.

Oracle Service Bus Administration Console saves the group and the group becomes available immediately to all sessions. If you are in a session when you

add the group and then you discard the session, Oracle Service Bus
Administration Console does not delete the new group.

25.8 Editing Groups

Use the View Group Details page to view and change details of a specific group.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic
   Server Admin privileges. Only users in the Admin role can modify security
   configuration data. See "Configuring Administrative Security" in the *Oracle Fusion
   Middleware Developer's Guide for Oracle Service Bus*.

2. Locate the group. See Section 25.6, "Locating Groups."

3. Click the group name. The View Group Details page displays the information
   shown in Table 25–4.

Table 25–4   Group Details

Property	Description
Group Name	The name of this group.
Authentication Provider	The authentication provider that contains this group definition.
Groups	The group to which this group belongs.

4. To edit the group details, click **Reconfigure** to display the Edit Group Details
   page.

   You can edit group details from inside or outside a session.

5. Make the appropriate changes to the **Group Membership** field. See Section 25.7,
   "Adding Groups" for descriptions of the field.

   You cannot change the **Group Name** field.

6. Click **Save Changes** to update the group.

   Oracle Service Bus Administration Console updates the group details and the
   update becomes available immediately to all sessions. If you are in a session when
   you update the group and then you discard the session, Oracle Service Bus
   Administration Console does not delete the updates.

25.9 Deleting Groups

Use the Summary of Groups page to delete a selected group or multiple groups.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic
   Server Admin privileges. Only users in the Admin role can modify security
   configuration data. See "Configuring Administrative Security" in the *Oracle Fusion
   Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** to display the Summary of Groups page.

3. Select the group you want to delete. You can select multiple groups if necessary.

   You can delete groups from inside or outside a session.

4. Click **Delete**. A message prompting you to confirm that you want to delete the
   group is displayed.

5. To delete the group, click **OK**.

Oracle Service Bus Administration Console deletes the group. If you are in a session when you delete the group and then you discard the session, Oracle Service Bus Administration Console does not un-delete the group.

6. Alternatively, you can click the **Delete** icon in the Options column of the group you want to delete.

25.10 Locating Roles

To locate roles:

1. Select **Security Configuration** > **Roles**. The Global Roles page displays the information shown in Table 25–5. For a more detailed description of the properties, see Section 25.12, "Editing Roles."

2. To restrict the number of items in the list or locate a specific role, scroll through the pages. Use the page controls above or below the table.

Table 25–5   Role Information

Property	Description
Role Name	The name of the role. The name is a link to the View Role Details page. See Section 25.12, "Editing Roles."
Provider Name	The authentication provider for this group.

25.11 Adding Roles

To add roles:

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** > **Roles** to display the Global Roles page.

3. Click **New**.

   You can add a role from inside or outside a session.

4. In the **Role Name** field, enter a unique name. This is a required field.

   Be sure that there are no spaces or < > characters in the security role name. Security role names are case sensitive. The Oracle convention is that all security role names are singular.

5. To create the role, click **OK**.

   Oracle Service Bus Administration Console saves the role and the role becomes available immediately to all sessions. If you are in a session when you add the role and then you discard the session, Oracle Service Bus Administration Console does not delete the new role.

   When you click **OK** to create the role, the next step is to define the conditions under which the role applies.

6. Continue in Section 25.11.1, "Defining Role Conditions."

25.11.1 Defining Role Conditions

To define role conditions:

1. On the Global Roles page, click the name of the new global role to display the Global Role Conditions page.

2. Under **Role Conditions**, click **Add Condition**.

    The following prompt is displayed:

    Choose the predicate you wish to use as your new condition

3. Choose a predicate from the list box. Typically, you choose **Group**. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

4. Click **Next**. Depending on what you chose for your condition predicate, do one of the following steps, described in Table 25–6.

Table 25–6   Condition Predicate Options

Condition Predicate...	Complete These Steps...
If you selected **Group**, enter one or more arguments that define the group or groups that should hold this role	1. In the **Group Argument Name** field, enter an argument that defines the group. 2. Click **Add**. 3. If necessary, repeat steps 1 and 2 until you have finished adding arguments. You can click **Remove** to remove the arguments from the list. 4. Click **Finish**.
If you selected **User**, enter one or more arguments that define the user or users that should hold this role	1. In the **User Argument Name** field, enter an argument that defines the user. 2. Click **Add**. 3. If necessary, repeat steps 1 and 2 until you have finished adding arguments. You can click **Remove** to remove the arguments from the list. 4. Click **Finish**.
If you selected **Server is in development mode**, **Allow access to everyone** or **Deny access to everyone**	Click **Finish**.
If you selected a time-constrained predicate such as **Access occurs between specified hours**, select start and end times and a GMT offset	1. In the **Starting Time** field, enter the earliest permissible time in the format hh:mm:ss AM|PM. For example, enter 12:45:00 AM. 2. In the **Ending Time** field, enter the latest permissible time in the format hh:mm:ss AM|PM. For example, enter 12:45:00 AM. 3. In the **GMT offset** field, enter the time ahead of GMT in the format GMT+hh:mm, or behind GMT in the format GMT-hh:mm. For example, Eastern Standard Time in the USA is GMT-5:00. 4. Click **Finish**.
If you selected **Context element defined**, enter a context element name	1. In the **Context element name** field, enter the name of the context element. 2. Click **Finish**.

Table 25–6   (Cont.) Condition Predicate Options

Condition Predicate...	Complete These Steps...
If you selected **Context element's value equals a numeric constant**, **Context element's value is greater than a numeric constant**, or **Context element's value is less than a numeric constant**, enter a context element name and a numeric value to compare it against	1. In the **Context element name** field, enter the name of the context element the value of which is to be evaluated. 2. In the **Numeric Value** field, enter a numeric value. 3. Click **Finish**.
If you selected **Context element's value equals a string value**, enter a context element name and a string value to compare it against	1. In the **Context element name** field, enter the name of the context element the value of which is to be evaluated. 2. In the **String Value** field, enter the string value that you want to compare. 3. Click **Finish**.
If you selected a time-constrained predicate such as **Access occurs before** or **Access occurs after**	1. In the **Date** field, enter a date in the format `mm/dd/yy`. For example, enter 1/1/04. You can add an optional time in the format `hh:mm:ss AM|PM`. For example, you can enter 1/1/04 12:45:00 AM. 2. Click **Finish**.
If you selected the time-constrained predicate **Access occurs on specified days of the week**, select the day of the week and a GMT offset	1. In the **Day of week** field, enter the day of the week. 2. In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00. 3. Click **Finish**.
If you selected a time-constrained predicate such as **Access occurs on a specified day of the month**, **Access occurs before a specified day of the month**, or **Access occurs after a specified day of the month**	1. In the Day of the Month field, enter the ordinal number of the day within the current month with values in the range from -31 to 31. Negative values count back from the end of the month, so the last day of the month is specified as -1. 0 indicates the day before the first day of the month. 2. In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00. 3. Click **Finish**.

5. If necessary, repeat the steps to add expressions based on different role conditions. In the Role Conditions section, you can do the following steps, described in Table 25–7, to modify the expressions.

Table 25–7   Role Conditions Options

To...	Complete These Steps...
Change the ordering of the selected expression.	Click **Move Up** and **Move Down**.
Merge or unmerge role conditions and switch the highlighted `and` or statements between expressions.	Click **Combine** and **Uncombine**.

Table 25–7   (Cont.) Role Conditions Options

To...	Complete These Steps...
Make a condition negative; for example, NOT Group Operators excludes the Operators group from the role.	Click **Negate**.
Delete a selected expression.	Click **Remove**.

6. When all the expressions in the Role Conditions section are correct, click **Save**.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

25.12 Editing Roles

Use the View Role Details page to view and change details of a specific role.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Locate the role. See Section 25.10, "Locating Roles."

3. Click the role name. The View Role Details page displays the information shown in Table 25–8.

Table 25–8   Role Details

Property	Description
Name	The name of the role.
Role Conditions	The conditions which determine membership in this role.

You can change role details from inside or outside a session.

4. Perform one of the following steps, described in Table 25–9.

Table 25–9   Role Options

To...	Complete This Step...
Change the ordering of the selected expression.	Click **Move Up** and **Move Down**.
Merge or unmerge role conditions and switch the highlighted and or statements between expressions.	Click **Combine** and **Uncombine**.
Make a condition negative; for example, NOT Group Operators excludes the Operators group from the role.	Click **Negate**.
Delete a selected expression.	Click **Remove**.

5. Click **Save**.

   Oracle Service Bus Administration Console updates the role and the update becomes available immediately to all sessions. If you are in a session when you update the role and then you discard the session, Oracle Service Bus Administration Console does not delete the updates.

25.13 Deleting Roles

To delete roles:

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Select **Security Configuration** > **Roles** to display the Global Roles page.

3. Select the role you want to delete. You can select multiple roles if necessary.

   You can delete roles from inside or outside a session.

4. Click **Delete**. A message prompting you to confirm that you want to delete the role is displayed.

5. To delete the role, click **OK**.

   Oracle Service Bus Administration Console deletes the role. If you are in a session when you delete the role and then you discard the session, Oracle Service Bus Administration Console does not un-delete the role.

25.14 Locating Access Control Policies

The Security page provides a link to the access control policies for a proxy service in the current Oracle Service Bus domain.

This page lists does not list proxy services that you have created in session but have not yet activated. To edit access control policies for a new proxy service, first activate the session in which you created the proxy service.

1. To locate the access control policies for a new proxy service, activate the session in which you created the proxy service.

2. In the Access Control column select the name of the proxy service from **Transport Access Control**, or the name of the proxy service or a particular operation from **Message Access Control**.

25.15 Editing Transport-Level Access Policies

Use the View Policy Details page to edit the transport-level access control policy of a proxy service. The page displays the information shown in Table 25–10.

Table 25–10   Policy Details

Property	Description
Proxy Service Name	Displays the name of the proxy service name for which you selected **Transport Access Control** on the Security page.
Providers	Displays the authorization providers that are configured for the security realm.

Table 25–10   (Cont.) Policy Details

Property	Description
Policy Conditions	Displays the conditions that determine for which users the proxy service will process requests.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Locate the access policy. See Section 25.14, "Locating Access Control Policies."

3. On the Security page, in the Transport Access Control column, click the name of the proxy service. The policy editor is displayed.

    You can edit an access control policy from inside a session.

4. In the **Authorization Providers** field, select an authorization provider. Oracle recommends that you select the XACMLAuthorizer.

    > **Note:**   Oracle Service Bus has deprecated support for the WebLogic Default Authorization provider. Instead, Oracle recommends that you use the WebLogic XACML Authorization provider.

5. Add policy conditions. See Section 25.17, "Adding Policy Conditions."

6. When you have finished entering conditions in the Policy Conditions section, click **Save**.

25.16 Editing Message-Level Access Policies

Use the View Policy Details page to edit the message-level access control policy of a proxy service that is a Web Service and is configured to require message-level security. The page displays the information shown in Table 25–11.

Table 25–11   Policy Details

Property	Description
Proxy Service Name	Displays the name of the proxy service name for which you selected **View Policies** on the Access Control for Proxy Services page.
Providers	Displays the authorization providers that are configured for the realm.
Operation	Lists the operation in the proxy service that can be secured.
Policy Conditions	Displays the conditions that determine which users can invoke the operations that are selected under Service Operations.

1. Log in to the Oracle Service Bus Administration Console as a user with WebLogic Server Admin privileges. Only users in the Admin role can modify security configuration data. See "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

2. Locate the access policy. See Section 25.14, "Locating Access Control Policies."

3. On the Security page, in the Message Access Control column, click the name of the proxy service or the particular operation in that proxy service. The policy editor is displayed.

   You can edit an access control policy from inside a session.

4. In the **Authorization Providers** field, select an authorization provider. Oracle recommends that you select the XACMLAuthorizer.

   > **Note:**   Oracle Service Bus has deprecated support for the WebLogic Default Authorization provider. Instead, Oracle recommends that you use the WebLogic XACML Authorization provider.

5. Add policy conditions. See Section 25.17, "Adding Policy Conditions."

6. When you have finished entering conditions in the Policy Conditions section, click **Save**.

25.17 Adding Policy Conditions

To add policy conditions:

1. Access the policy editor for an access control policy. See Section 25.15, "Editing Transport-Level Access Policies" or Section 25.16, "Editing Message-Level Access Policies."

2. In the policy editor, under Policy Conditions, click **Add Condition**.

   The following prompt is displayed:

   Choose the predicate you wish to use as your new condition

3. Select a predicate from the list.

4. Click **Next**. Depending on what you chose for your condition predicate, do one of the following steps, shown in Table 25–12.

Table 25–12   Condition Predicate Options

If You Selected...	Complete These Steps...

Role
 	
(For transport-level security, this condition applies only if the proxy service uses a protocol that enables a client to supply credentials.)
	1. In the **Role Argument Name** field, enter the role to which you want to grant access.
	
	   If you have not already created the role that you entered in this field, you can do so after you finish creating access control policies. See Section 25.11, "Adding Roles." If you do not create this role, then no one will be granted access.
	
	2. Click **Add**.
	
	3. If necessary, repeat steps 1 and 2 until you have finished adding arguments. You can click **Remove** to remove the arguments from the list.
	
	4. Do one of the following:
	
	   To save the arguments and return to the predicate list, click **Finish**.
	
	   To discard the changes and return to the predicate list, click **Back**.
	
	   To discard the changes and return to the View Policy Details page, click **Cancel**.

Table 25–12   (Cont.) Condition Predicate Options

If You Selected...	Complete These Steps...
Group (For transport-level security, this condition applies only if the proxy service uses a protocol that enables a client to supply credentials.)	**1.** In the **Group Argument Name** field, enter the group to which you want to grant access. If you have not already created the group that you entered in this field, you can do so after you finish creating access control policies. See Section 25.7, "Adding Groups." If you do not create this group, then no one will be granted access. **2.** Click **Add**. **3.** If necessary, repeat steps 1 and 2 until you have finished adding arguments. You can click **Remove** to remove the arguments from the list. **4.** Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.
User (For transport-level security, this condition applies only if the proxy service uses a protocol that enables a client to supply credentials.)	**1.** In the **User Argument Name** field, enter the user to which you want to grant access. If you have not already created the user that you entered in this field, you can do so after you finish creating access control policies. See Section 25.3, "Adding Users." If you do not create this user, then no one will be granted access. **2.** Click **Add**. **3.** If necessary, repeat steps 1 and 2 until you have finished adding arguments. You can click **Remove** to remove the arguments from the list. **4.** Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.
Access occurs on specified days of the week	**1.** In the **Day of week** field, enter the day of the week. **2.** In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00. **3.** Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.

Table 25–12   (Cont.) Condition Predicate Options

If You Selected...	Complete These Steps...
Access occurs between specified hours	1. In the **Starting Time** field, enter the earliest permissible time in the format `hh:mm:ss AM|PM`. For example, enter 12:45:00 AM.
	2. In the **Ending Time** field, enter the latest permissible time in the format `hh:mm:ss AM|PM`. For example, enter 12:45:00 AM.
	3. In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00.
	4. Do one of the following:
	To save the arguments and return to the predicate list, click **Finish**.
	To discard the changes and return to the predicate list, click **Back**.
	To discard the changes and return to the View Policy Details page, click **Cancel**.
Access occurs before or **Access occurs after**	1. In the **Date** field, enter a date in the format `mm/dd/yy`. For example, enter 1/1/04. You can add an optional time in the format `hh:mm:ss AM|PM`. For example, you can enter 1/1/04 12:45:00 AM.
	2. In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00.
	3. Do one of the following:
	To save the arguments and return to the predicate list, click **Finish**.
	To discard the changes and return to the predicate list, click **Back**.
	To discard the changes and return to the View Policy Details page, click **Cancel**.
Access occurs on a specified day of the month, **Access occurs before a specified day of the month**, or **Access occurs after a specified day of the month**	1. In the **The day of the month** field, enter the ordinal number of the day within the current month with values in the range from -31 to 31. Negative values count back from the end of the month, so the last day of the month is specified as -1. 0 indicates the day before the first day of the month.
	2. In the **GMT offset** field, enter the time ahead of GMT in the format `GMT+hh:mm`, or behind GMT in the format `GMT-hh:mm`. For example, Eastern Standard Time in the USA is GMT-5:00.
	3. Do one of the following:
	To save the arguments and return to the predicate list, click **Finish**.
	To discard the changes and return to the predicate list, click **Back**.
	To discard the changes and return to the View Policy Details page, click **Cancel**.

Table 25–12 (Cont.) Condition Predicate Options

If You Selected...	Complete These Steps...
Context element's value equals a string constant (Applies only to transport-level security. A context element is a parameter/value pair that a container such as a Web container can optionally provide to a security provider. Context elements are not available for message-level access control policies.)	1. In the **Context element name** field, enter the name of the context element for which to evaluate the value. For possible values, see "Context Properties Are Passed to Security Providers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. 2. In the **String Value** field, enter the string value that you want to compare. 3. Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.
Context element's value is greater than a numeric constant, **Context element's value equals a numeric constant**, or **Context element's value is less than a numeric constant** (Applies only to transport-level security. A context element is a parameter/value pair that a container such as a Web container can optionally provide to a security provider. Context elements are not available for message-level access control policies.)	1. In the **Context element name** field, enter the name of the context element for which to evaluate the value. For possible values, see "Context Properties Are Passed to Security Providers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. 2. In the **Numeric Value** field, enter a numeric value. 3. Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.
Context element defined (Applies only to transport-level security. A context element is a parameter/value pair that a container such as a Web container can optionally provide to a security provider. Context elements are not available for message-level access control policies.)	1. In the **Context element name** field, enter the name of the context element. For possible values, see "Context Properties Are Passed to Security Providers" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. 2. Do one of the following: To save the arguments and return to the predicate list, click **Finish**. To discard the changes and return to the predicate list, click **Back**. To discard the changes and return to the View Policy Details page, click **Cancel**.
Deny access to everyone, **Allow access to everyone** or **Server is in development mode**	Click **Finish**. Alternatively, you can click **Cancel** to discard the changes and return to the View Policy Details page.

5. If necessary, repeat steps 3-5 to add expressions based on different policy conditions. In the Policy Conditions section, you can do the following steps, shown in Table 25–13, to modify the expressions.

Table 25–13   Policy Conditions Options

To...	Complete These Steps...
Change the ordering of the selected expression.	Select the check box associated with the condition, then click **Move Up** and **Move Down**.
Merge or unmerge policy conditions and switch the highlighted `and or` statements between expressions.	Select the check box associated with the appropriate conditions, then click **Combine** and **Uncombine**.
Make a condition negative; for example, NOT Group Operators excludes the Operators group from the policy.	Select the check box associated with the condition, then click **Negate**.
Delete a selected expression.	Select the check box associated with the condition, then click **Remove**.

Related Topics

See "Security Policy Conditions" in *Oracle Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server*.

Monitoring

This chapter describes how to monitor your services in Oracle Service Bus (OSB) using the OSB Administration Console. Topics include working with SLA and pipeline alerts, statistics, operational settings, and metrics.

When you create a business or proxy service, monitoring is disabled by default for that service. To learn how to enable monitoring for proxy services, see Section 26.8, "Configuring Operational Settings for Proxy Services"; for business services, see Section 26.9, "Configuring Operational Settings for Business Services."

To enable or disable monitoring of all services that have individually been enabled or disabled for monitoring, use the **Enable Monitoring** option on the **Operations** > **Global Settings** page. See Section 27.2, "Enabling Global Settings."

To learn more about monitoring at runtime, see "Monitoring Oracle Service Bus at Runtime" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

26.1 Viewing SLA Alerts

Use the SLA Alerts page to view a summary of SLA alerts and access detailed alert history information, shown in Table 26–1.

The SLA Alerts page includes the following:

- Tabs for displaying Pipeline Alerts, Service Health, and Server Health information.

  On the SLA Alerts tab header, a numeric indicator shows the number of new alerts that have occurred since you last viewed them or during the last Dashboard refresh period, whichever is more recent. By default, the Dashboard refresh rate is **No Refresh**. To change the Dashboard refresh rate interval, see Section 27.3, "Setting User Preferences."

  Caution: By selecting a dashboard refresh rate interval other than the default (**No Refresh**), the browser refreshes the Dashboard at regular intervals. This prevents your session from timing out, even without you interacting with the Administration Console.

  Oracle recommends that you use the Dashboard refresh rate feature with caution. You should never leave an Administration Console session unattended. While your Administration Console session is active others can gain access to it from your computer, without re-authentication.

- A pie chart of the services with SLA alerts over the past 30 minutes, which is the default alert history duration interval. To change the time interval for displaying

historical alert data, see Section 27.3, "Setting User Preferences."

The number of SLA alerts on the page match the pie chart which displays the breakdown of alerts by severity level, over the same time interval. You can click the area in the chart corresponding to the alert severity to display more details.

- The top 10 services with SLA alerts are listed in descending order of the number of alerts issued by those services. Each service name is a link to the service monitoring details. See Section 26.11, "Viewing Service Metrics."

- Fields that display alert information depending on your customized settings for the alert table. For example, timestamp, alert name, alert severity, action, and the names of the service and service type associated with the alert. To learn more about these fields, see Table 26–1.

An Annotation icon next to an alert name indicates annotations. Positioning the mouse pointer over the icon previews the annotation text. Click the icon to view the entire annotation text.

- A Table Customizer icon. See Section 2.2, "Customizing Table Views."

- A link to the Extended Alert History page. See Section 26.3, "Locating Alerts."

Table 26–1   SLA Alert History Information

Property	Description
Timestamp	The date and time that the alert rule last evaluated to true. By default, all the alerts are sorted according to the timestamp.
Alert Name	The name assigned to the alert. The name is a link to the Alert Details page. See Section 26.5, "Viewing Alert Details." An Annotation icon next to an alert name indicates annotations. Positioning the mouse pointer over the icon previews the annotation text.
Alert Severity	The user-defined severity of the alert. • Fatal • Critical • Major • Minor • Warning • Normal
Service	The name of the service associated with the alert. The name is a link to the Service Monitoring Details page (only when the service is being actively monitored). See Section 26.11, "Viewing Service Metrics."
Service Type	The parent service type: proxy service or business service.
Action	The View Alert Rule Details icon is a link to the Alert Rule General Configuration page. See Section 26.24, "Viewing Alert Rule Configurations."

26.2 Viewing Pipeline Alerts

Use the Pipeline Alerts page to view a summary of pipeline alerts and access detailed alert history information, shown in Table 26–2.

The Pipeline Alerts page includes the following:

- Tabs for displaying SLA Alerts, Service Health, and Server Health information.

  On the Pipeline Alerts tab header, a numeric indicator shows the number of new alerts that have occurred since you last viewed them or during the last Dashboard refresh period, whichever is more recent. By default, the Dashboard refresh rate is **No Refresh**. To change the Dashboard refresh rate interval, see Section 27.3, "Setting User Preferences."

  > **Caution:** By selecting a dashboard refresh rate interval other than the default (**No Refresh**), the browser refreshes the Dashboard at regular intervals. This prevents your session from timing out, even without you interacting with the Administration Console.
  >
  > Oracle recommends that you use the Dashboard refresh rate feature with caution. You should never leave an Administration Console session unattended. While your Administration Console session is active others can gain access to it from your computer, without re-authentication.

- A pie chart of the services with pipeline alerts over the past 30 minutes, which is the default alert history duration interval. To change the time interval for displaying historical alert data, see Section 27.3, "Setting User Preferences."

  The number of pipeline alerts on the page match the pie chart which displays the breakdown of alerts by severity level, over the same time interval. You can click the area in the chart corresponding to the alert severity to display more details.

- The top 10 services with pipeline alerts are listed in descending order of the number of alerts issued by those services. Each service name is a link to the service monitoring details. See Section 26.11, "Viewing Service Metrics."

- Fields that display alert information depending on your customized settings for the alert table. For example, timestamp, alert summary, alert severity, action, and the names of the service and service type associated with the alert. To learn more about these fields, see Table 26–2.

  An Annotation icon next to an alert summary indicates annotations. Positioning the mouse pointer over the icon previews the annotation text. Click the icon to view the entire annotation text.

- A Table Customizer icon. See Section 2.2, "Customizing Table Views."

- A link to the Extended Alert History page.

Table 26–2   Pipeline Alert History Information

Property	Description
Timestamp	The date and time that the alert occurred.
	By default, all the alerts are sorted according to the timestamp.
Alert Summary	A short description of the pipeline alert action or Oracle Service Bus Alert (the default value provided to pipeline alert actions that were designed without alert summary text). The alert summary is a link to the Alert Details page. See Section 26.5, "Viewing Alert Details."
	An Annotation icon next to an alert summary indicates annotations. Positioning the mouse pointer over the icon previews the annotation text.

Table 26–2 (Cont.) Pipeline Alert History Information

Property	Description
Alert Severity	The user-defined severity of the alert. • Fatal • Critical • Major • Minor • Warning • Normal
Service	The name of the service associated with the alert. The name is a link to the Service Monitoring Details page (only when the service is being actively monitored). See Section 26.11, "Viewing Service Metrics."
Service Type	Only proxy services can define pipeline alerts, therefore, for pipeline alerts, all parent services are proxy services.
Action	The Edit Message Flow icon is a link to the Edit Message Flow page of that proxy service.

26.3 Locating Alerts

To locate alerts:

1. Select **Operations** > **Dashboard**.

2. Select the **SLA Alerts** or **Pipeline Alerts** tab to display SLA or pipeline alerts.

3. Click the **Extended Alert History** link. The Extended SLA Alert History and Extended Pipeline Alert History pages display detailed alert history information.

4. To restrict the number of items in the list or locate specific alerts:

   ■ Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   ■ Filter by date range. Do one of the following:

     – Select **All** to ignore the date range and time interval filter criteria and display all alerts.

     – In the **From** and **To** fields, enter a date and time in the format `MM/DD/YY HH:MM AM|PM`, then click **Search**. The alerts that occurred over the specified date and time range are displayed.

     – In the **For the Last** field, enter the number of days, then specify hours and minutes, then click **Search**. The alerts that occurred over the specified time range are displayed.

   ■ Filter by alert severity. From the **Alert Severity** list, select to restrict alerts to the specified severity level, then click **Search**.

     Select the **or above** check box to restrict your search to the specified severity level or above (listed from the most inclusive to the most restrictive level): Normal, Warning, Minor, Major, Critical, and Fatal.

   ■ Filter by parent **Service**. Click **Browse** to display the Select Service page. Select the parent service from the list and click **Submit**. Click **Clear** to remove the parent service filter criteria, or click **Search**.

- For SLA alerts only, filter by **Alert Name (Any String)**. Enter any string, including wildcard characters, then click **Search**. For more details, see Section 26.3.1, "WebLogic Diagnostics Framework Query Language."

- For pipeline alerts only, filter by **Alert Summary (Any String)**. Enter any string, including wildcard characters, then click **Search**. For more details, see Section 26.3.1, "WebLogic Diagnostics Framework Query Language." To find pipeline alert actions that were designed without alert summary text, enter Oracle Service Bus Alert.

- Click **View All** to remove the search filters and display all alerts.

Use the Extended SLA Alert History and Extended Pipeline Alert History pages also to do the following:

- To display the alert data in a pie chart, select **Pie Chart View**.

- To display the same data in a bar chart, select **Bar Chart View**.

- To display the data in tabular format, select **Table View**.

- To display an updated version of the data, click **Refresh**.

- To delete individual alerts, select the corresponding check boxes and click **Delete** at the bottom of the page.

- To delete all alerts or those that were issued within a specified date and time range, click the **Purge Alert History** link at the top of the page. See Section 26.4, "Purging Alerts."

26.3.1 WebLogic Diagnostics Framework Query Language

Alerts are stored using the WebLogic Diagnostics Framework, which provides its own query language, including wildcards. For filtering alerts in extended alert history, use the syntax described in "WLDF Query Language" in *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

26.4 Purging Alerts

Use this page to permanently delete the specified SLA or pipeline alerts.

> **Note:** This action cannot be undone.

1. Select **Operations** > **Dashboard**.

2. Select the **SLA Alerts** or **Pipeline Alerts** tab, then click the **Extended Alert History** link.

3. On the upper right-hand side of the Extended Alert History page, click **Purge SLA Alert History** or **Purge Pipeline Alert History**.

4. Do one of the following steps, shown in Table 26–3.

Table 26–3   Alert Purging Options

To...	Complete These Steps...
Purge all SLA or pipeline alerts	1. Select **Purge All Alerts**. 2. Click **Purge**.

Table 26–3   (Cont.) Alert Purging Options

To...	Complete These Steps...
Purge SLA or pipeline alerts within a specified time frame	1. Select **Purge Alerts From**... **To**... 2. In the **Purge From** field, enter a month, day, year, and time. 3. In the **Purge To** field, enter a month, day, year, and time. 4. Click **Purge**.

> **Note:**   If the server from which the alerts are being purged is unavailable, the logging framework persists a record of the purge and polls the server every 30 minutes to check its availability. When the logging framework finds the server available, it purges the alert(s).

26.5 Viewing Alert Details

This page displays details of specific alerts including the information shown in Table 26–4.

Table 26–4   Alert Details

Property	Description
Alert Name	For SLA alerts only: The name assigned to the alert. The name is a link to the Alert Rule Configuration page. See Section 26.24, "Viewing Alert Rule Configurations."
Alert Summary	For pipeline alerts only: A short description of the pipeline alert action or Oracle Service Bus Alert (the default value provided to pipeline alert actions that were designed without alert summary text).
Description	A description for the alert rule.
Timestamp	■ For pipeline alerts: The date and time that the alert occurred. ■ For SLA alerts: The date and time that the alert rule last evaluated to true.
Severity	The user-defined severity of the alert. ■ Fatal ■ Critical ■ Major ■ Minor ■ Warning ■ Normal
Service	The name and path of the service associated with this alert. The field is a link to the Service Monitoring Details page (only when the service is being actively monitored). See Section 26.11, "Viewing Service Metrics."
Service Type	The parent service type: proxy service or business service. Only proxy services can define pipeline alerts, therefore, for pipeline alerts, all parent services are proxy services.
Server	The name of the server in which this alert was generated. For SLA alerts only: N/A is displayed.

Table 26–4 (Cont.) Alert Details

Property	Description
Annotation	A text box in which you can enter notes for this alert. An Annotation icon is displayed in the Alert History for alerts with annotations.

26.6 Viewing Service Monitoring Information

The Service Health page displays metrics for services that have monitoring enabled. To learn how to enable monitoring for services, see Section 26.8, "Configuring Operational Settings for Proxy Services" and Section 26.9, "Configuring Operational Settings for Business Services."

By default, the Dashboard refresh rate is **No Refresh**. To change the Dashboard refresh rate interval, see Section 27.3, "Setting User Preferences."

Caution: By selecting a dashboard refresh rate interval other than the default (**No Refresh**), the browser refreshes the Dashboard at regular intervals. This prevents your session from timing out, even without you interacting with the Administration Console.

Oracle recommends that you use the Dashboard refresh rate feature with caution. You should never leave an Administration Console session unattended. While your Administration Console session is active others can gain access to it from your computer, without re-authentication.

On the Service Health tab header, these icons may be present:

- A Warning icon and numeric indicator.

  – Click the icon to filter the services displayed using **Has Errors** criteria.

  – The number indicates the number of services with errors since the page was last refreshed.

- An Endpoints Offline icon and numeric indicator.

  – Click the icon to filter the services displayed using **Has Offline Endpoint URIs** criteria.

  – The number indicates the number of services with endpoints that have become offline since the page was last refreshed.

The Service Health page displays service metrics and monitoring information, shown in Table 26–5.

1. In the **Display Statistics** field, do one of the following:

   - To display monitoring statistics for each service for the period of the current aggregation interval, select **Current Aggregation Interval**. The Current Aggregation Interval view displays a moving statistic view of the service metrics.

   - To display monitoring statistics for the period since you last reset statistics for a service, select **Since Last Reset**. The Since Last Reset view displays a running count of the metrics.

2. Select a **Server** from the list to display metrics for that server.

If a cluster exists, cluster-wide metrics are displayed by default. Select an individual Managed Server to display metrics for that server.

3. To restrict the number of items in the list or locate specific services:

- Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

- Filter by service name. In the **Name** field, enter the name of the search target or enter wildcard characters (use * and ? as wildcard characters to perform a more general search), then click **Search**.

  This search method is preferable if the name of the service is unique across all projects and paths.

- Filter by service path. In the **Path** field, enter the path of the search target, which is the project name and the name of the folder in which services reside. You can use * and ? as wildcard characters to perform a more general search. All the services that reside in that path are displayed.

  The format for the **Path** field is as follows:

  ```
  project-name/root-folder/ . . ./parent-folder
  ```

  If a service is directly under the project, the format is as follows: `project-name`

- Filter by service name and path. This search method is preferable if there is more than one service with the same name that reside in different paths.

- Filter by parent service. From the **Service** list, select the parent service type: Proxy Services or Business Services, then click **Search**. Select All Services to ignore the parent service filter criteria.

- Filter by services with messages. Select the **Has Messages** option, then click **Search**.

- Filter by services with alerts. Select the **Has Alerts** option, then click **Search**.

- Filter by services with errors. Select the **Has Errors** option, then click **Search**.

- Filter by services with offline URIs. Select the **Has Offline Endpoint URIs** option, then click **Search**.

- Filter by services invoked by a particular proxy service. Enter the name and path of a proxy service in the field provided or click **Browse** to display the Select Proxy Service page. Select the proxy service from the list and click **Submit**. Click **Search**. The business services invoked by the proxy service you specified are displayed.

  Wildcard characters are not supported in the **Invoked by Proxy** field.

- Click **View All** to remove the search filters and display all services.

Table 26–5   Service Monitoring Information

Property	Description
Name	The name assigned to the service. The name is a link to the Service Monitoring Details page. See Section 26.11, "Viewing Service Metrics."

Table 26–5   (Cont.) Service Monitoring Information

Property	Description
Path	The project associated with the service. If the service resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Service Type	The parent service type: proxy service or business service.
Aggregation Interval	This field is displayed only when you have selected **Current Aggregation Interval** in the **Display Statistics** field. The current aggregation interval set for monitoring this service, in terms of hours and minutes. You set this interval on the Operational Settings page. See Section 26.10, "Setting the Aggregation Interval for a Service."
Avg. Response Time	The average response time (in msec) that this service has taken to execute messages within the period of the current aggregation interval or for the period since the last reset.
Messages	The number of messages associated with this service for the period of the current aggregation interval or for the period since the last reset.
Errors	The number of error messages associated with this service for the period of the current aggregation interval or for the period since the last reset.
Alerts	This field is hidden by default. The number of alerts (SLA and pipeline) associated with this service for the period of the current aggregation interval or for the period since the last reset. For an individual Managed Server in a cluster, for proxy services, it shows the number of pipeline alerts *only* associated with this service for the period of the current aggregation interval or for the period since the last reset. For business services, for individual Managed Servers, displays N/A.
SLA Alerts	The number of SLA alerts associated with this service for the period of the current aggregation interval or for the period since the last reset. For an individual Managed Server in a cluster, SLA Alerts will be N/A (only cluster-wide SLA alerts are displayed).
Pipeline Alerts	For proxy services only: The number of pipeline alerts associated with this service for the period of the current aggregation interval or for the period since the last reset.

Table 26–5   (Cont.) Service Monitoring Information

Property	Description
Endpoint URI Status	For business services only, the status of endpoint URIs. The status is a link to the Endpoint URIs page. See Section 26.15, "Viewing Business Services Endpoint URIs Metrics."
	For a single node domain (individual server instances):
	■ `Online`—All the endpoint URIs for the business service are online.
	■ `Offline`—All the endpoint URIs for the business service are offline.
	■ `Partial`—At least one of the endpoint URIs for the business service is offline.
	For a clustered domain:
	■ When the **Server** field is set to Cluster or to one of the Managed Servers, `Online` status denotes that all of the endpoint URIs are online across the cluster or on the selected Managed Server, respectively.
	■ When the **Server** field is set to Cluster or to one of the Managed Servers, `Offline` status denotes that all of the endpoint URIs are offline across the cluster or on the selected Managed Server, respectively.
	■ When the **Server** field is set to Cluster, `Partial` status denotes that at least one of the endpoint URIs for the business service is offline on at least one of the servers, or that one of the endpoint URIs is offline on all the servers, but the other endpoint URIs for the same business service are still available on one or all the servers.
	■ When the **Server** field is set to one of the Managed Servers, `Partial` status denotes that at least one of the endpoint URIs for the business service is offline on the selected Managed Server.
Action	This field is displayed only when you have selected **Since Last Reset** in the **Display Statistics** field.
	In this column, you can click the **Reset** icon for a specific service to reset the statistics for that service. When you confirm you want to do this, the system deletes all monitoring statistics that were collected for the service since you last reset statistics. However, the system does not delete the statistics being collected during the Current Aggregation Interval for the service. Additionally, after you click the **Reset** icon, the system immediately starts collecting monitoring statistics for the service again.

26.7 Resetting Statistics for Services

You can use the Service Health page to reset monitoring statistics for all services in your configuration. You can reset statistics whether you are in a session.

1. Select **Operations** > **Dashboard** > **Service Health**.

2. In the **Display Statistics** field, select **Since Last Reset**. The Since Last Reset view displays a running count view of the metrics.

3. To reset the monitoring statistics for a specific service, click the **Reset** icon in the Action column for that service.

    When you confirm that you want to do this, the system deletes all monitoring statistics that were collected for the service since you last reset statistics. However, the system does not delete the statistics being collected during the **Current**

Aggregation Interval for the service. Additionally, after you click the **Reset** icon, the system immediately starts collecting monitoring statistics for the service again.

4. To reset monitoring statistics for all services, click the **Reset All Statistics** link at the top of the page.

> **Note:** Clicking **Reset All Statistics** resets the statistics for all monitored services regardless of whether they are displayed on the page or not.

When you confirm that you want to reset statistics, the system deletes all monitoring statistics that were collected for all services in your configuration since you last reset statistics. The system does not delete the statistics being collected during the **Current Aggregation Interval** for any of the services. Additionally, after you click **Reset All Statistics**, the system immediately starts collecting monitoring statistics for the services again.

26.8 Configuring Operational Settings for Proxy Services

Use the Operational Settings page to enable and disable a proxy service and monitoring for specific proxy services. Similarly, you can specify these same service-specific operational settings on the **Operations** > **Smart Search** page, with the following exceptions: you cannot set an aggregation interval or specify an alerting or logging severity level. For more information, see Section 27.1, "Finding and Updating Operational Settings."

The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and service-level settings for a service to be completely enabled at runtime. Additionally, the **Service State** must also be enabled. See Section 27.2, "Enabling Global Settings."

You can change and save monitoring configuration settings even if the service will be not be enabled at runtime. For example, you can change and save the Aggregation Interval even if **Service Monitoring** is disabled. In this manner, you can edit settings and later enable them.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Proxy Services**.

3. Click the proxy service name.

   The View a Proxy Service page displays the information shown in Table 26–6.

Table 26–6   Proxy Service Information

Property	Description
Last Modified By	The user who created or edited this service, or imported it into the configuration.
Last Modified On	The date and time that the user created or edited this service, or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this service references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."

Table 26–6   (Cont.) Proxy Service Information

Property	Description
Referenced By	The number of objects that reference this service. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of the service, if one exists.

4. Select the **Operational Settings** tab.

You must be in an active session to configure proxy service operational settings shown in Table 26–7.

Table 26–7   Configuring Proxy Service Operational Settings

To...	Do This...
Disable **State**	Clear the **Enabled** check box. When you activate the session, the service stops processing messages. Disabling a service no longer causes the system to delete all monitoring statistics previously collected for the service.
Enable **State**	Select the **Enabled** check box. When you activate the session, the service resumes processing messages.
Disable **Monitoring**	Clear the **Enabled** check box. When you activate the session, the system stops collecting monitoring statistics for the service, and deletes any statistics collected previously.
Enable **Monitoring**	Select the **Enabled** check box next to Monitoring. From the **Enable Pipeline Monitoring at** list, select to restrict monitoring to the specified level or above (listed from the most inclusive to the most restrictive level): Action, Pipeline, and Service. When you activate the session, the system starts collecting monitoring statistics for the service.
Set an **Aggregation Interval** for a service	Select the interval in terms of hours or minutes, then click **Update**. If your selection for hours exceeds 1, then the default selection for minutes is always zero. However, if your selection for hours is 0 or 1, then you can configure intervals in terms of minutes. See Section 26.10, "Setting the Aggregation Interval for a Service." You can change and save the Aggregation Interval setting whether or not you have enabled monitoring. To enable monitoring at runtime, make sure the **Enable Monitoring** check box is selected.
Disable **SLA Alerts**	Clear the **Enabled** check box. The system stops evaluating any alert rules configured for the service; therefore, you no longer receive SLA alerts associated with the service.
Enable **SLA Alerts**	Select the **Enabled** check box. From the **Enable Alerting at** list, select to restrict SLA alerts to the specified level or above (listed from the most inclusive to the most restrictive level): Normal (default), Warning, Minor, Major, Critical, and Fatal. Service SLA alerting depends on both the global and service-level monitoring states—both must be enabled for SLA alerting to be enabled at runtime. See Section 27.2.3, "Enabling SLA Alerts Globally."
Disable **Pipeline Alerts**	Clear the **Enabled** check box. When you activate the session, the system stops executing pipeline alert actions; therefore, you no longer receive pipeline alerts associated with the service.

Table 26–7   (Cont.) Configuring Proxy Service Operational Settings

To...	Do This...
Enable **Pipeline Alerts**	Select the **Enabled** check box. From the **Enable Alerting at** list, select to restrict pipeline alerts to the specified severity level or above (listed from the most inclusive to the most restrictive level): Normal (default), Warning, Minor, Major, Critical, and Fatal.
Disable the report action of a proxy service	Clear the **Enabled** check box next to **Message Reporting**. When you activate the session, the system stops executing report actions; therefore, you no longer receive message reporting associated with the service.
Enable the report action of a proxy service	Select the **Enabled** check box next to **Message Reporting**.
Disable logging output for a pipeline log action	Clear the **Enabled** check box next to **Logging**. When you activate the session, the system stops executing log actions; therefore, you no longer receive logging output associated with the service.
Enable logging output for a pipeline log action	Select the **Enabled** check box next to **Logging**. From the **Enable Logging at** list, choose to restrict logging output to the specified level or above (listed from the most inclusive to the most restrictive level): Debug (default), Info, Warning, and Error.
Enable runtime execution tracing for a proxy service	Select the **Enabled** check box next to **Execution Tracing**. After you enable execution tracing, the system logs various details culled from the Message Flow Context and the Message Context. These details include: stage name; pipeline or route node name; and the current Message Context. **Note**: To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels: ■ Minimum severity to log: Info ■ Log file: Info ■ Standard out: Info For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.
Disable runtime tracing for a proxy service	Clear the **Enabled** check box next to **Execution Tracing**.

Table 26–7   (Cont.) Configuring Proxy Service Operational Settings

To...	Do This...
Enable message tracing for a proxy service	Select the **Enabled** check box next to **Message Tracing**. After you enable message tracing, the system logs messages exchanged between the Oracle Service Bus pipeline and the proxy service (inbound request and response, as well as outbound request and response messages).
	Note: To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:
	■ Minimum severity to log: Info
	■ Log file: Info
	■ Standard out: Info
	For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.
	When applicable, logged outbound messages can also include the retry number, error code, and error message.
	From the **Detail Level** list, choose to specify the level of detail from among the following:
	■ Terse—Display the date, time, service type, service name, and URI
	■ Headers—Display terse information along with the XML representation of the message metadata
	■ Full—Display the headers information along with the raw payload, including attachments if any
	If you choose Full from the **Detail Level** list, specify the maximum size (in kilobytes) for the message payload using the **Payload Tracing Limit** field. Also, specify the default encoding for logging the payload using the **Default Encoding** field. This can be useful when logging binary payloads or SOAP messages with binary attachments.
	The default encoding value can be Base64 or any Java-supported encoding as defined in: http://docs.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html.
	Leaving the **Default Encoding** field empty causes Oracle Service Bus to use the host's default encoding for the payload. The default encoding depends on a combination of the JVM, the underlying operating system (OS), and OS-level locale settings.
	If the setting specified in the **Default Encoding** field cannot be used (for example, it is not a valid option for the configuration), then Oracle Service Bus uses Base64 encoding for the payload.
Disable message tracing for a proxy service	Clear the **Enabled** check box next to **Message Tracing**.

26.9 Configuring Operational Settings for Business Services

Use the Operational Settings page to enable and disable a business service and monitoring for specific business services. Similarly, you can specify these same service-specific operational settings on the **Operations** > **Smart Search** page, with the following exceptions: you cannot set an aggregation interval or specify an alerting severity level. For more information, see Section 27.1, "Finding and Updating Operational Settings."

The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and service-level settings for a service to be completely enabled at runtime. Additionally, the **Service State** must also be enabled. See Section 27.2, "Enabling Global Settings."

You can change and save monitoring configuration settings even if the service will be not be enabled at runtime. For example, you can change and save the Aggregation Interval even if **Service Monitoring** is disabled. In this manner, you can edit settings and later enable them.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Business Services**.

3. Click the business service name.

   The View a Business Service page displays the information shown in Table 26–8.

Table 26–8   Business Service Information

Property	Description
Last Modified By	The user who created or edited this service, or imported it into the configuration.
Last Modified On	The date and time that the user created or edited this service, or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."Section 4.23, "View Change History Page."
References	The number of objects that this service references. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Referenced By	The number of objects that reference this service. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of the service, if one exists.

4. Select the **Operational Settings** tab.

   You must be in an active session to configure business service operational settings shown in Table 26–9.

Table 26–9   Configuring Business Service Operational Settings

To...	Do This...
Disable **State**	Clear the **Enabled** check box. When you activate the session, the service stops processing messages. Disabling a service no longer causes the system to delete all monitoring statistics previously collected for the service.
Enable **State**	Select the **Enabled** check box. When you activate the session, the service resumes processing messages.

Table 26–9   (Cont.) Configuring Business Service Operational Settings

To...	Do This...
Enable business service endpoint URIs to be taken offline Set a retry interval for offline URIs	Select the **Enable with Retry Interval** check box. When you select this option, the business service removes non-responsive endpoint URIs (takes them offline), at runtime, so that only the responsive URIs are used for retry attempts and for processing subsequent requests. To keep non-responsive URIs offline until you take corrective action and then re-enable the URIs, do not provide a retry interval. For example, a zero retry interval indicates that the endpoint remains offline indefinitely. To bring it back online, use the **Mark Endpoint Online** action on the Endpoint URIs page. See Section 26.15, "Viewing Business Services Endpoint URIs Metrics." To configure the business service to take non-responsive URIs offline and then re-attempt accessing the same URI endpoint for subsequent message processing after a specified time interval, you do so by specifying a retry interval value. Use the **hours**, **minutes**, and **seconds** fields to specify how long to keep the non-responsive URIs out of the active URIs processing loop. The URI is kept offline for the specified time interval and then retried. If the endpoint responds, the URI becomes online again, or else it remains offline and the process repeats itself. You can specify a retry interval only if you have selected the **Enable with Retry Interval** option.
Disable **Monitoring**	Clear the **Enabled** check box. When you activate the session, the system stops collecting monitoring statistics for the service, and deletes any statistics collected previously.
Enable **Monitoring**	Select the **Enabled** check box. When you activate the session, the system starts collecting monitoring statistics for the service.
Set an **Aggregation Interval** for a service	Select the interval in terms of hours or minutes, then click **Update**. If your selection for hours exceeds 1, then the default selection for minutes is always zero. However, if your selection for hours is 0 or 1, then you can configure intervals in terms of minutes. See Section 26.10, "Setting the Aggregation Interval for a Service." You can change and save the Aggregation Interval setting whether or not you have enabled monitoring. To enable monitoring at runtime, make sure the **Enable Monitoring** check box is selected.
Disable **SLA Alerts**	Clear the **Enabled** check box. The system stops evaluating any alert rules configured for the service; therefore, you no longer receive SLA alerts associated with the service.
Enable **SLA Alerts**	Select the **Enabled** check box. From the **Enable Alerting at** list, select to restrict SLA alerts to the specified level or above (listed from the most inclusive to the most restrictive level): Normal (default), Warning, Minor, Major, Critical, and Fatal. Service SLA alerting depends on both the global and service-level monitoring states—both must be enabled for SLA alerting to be enabled at runtime. See Section 27.2.3, "Enabling SLA Alerts Globally."

Table 26–9   (Cont.) Configuring Business Service Operational Settings

To...	Do This...
Restrict the flow of messages sent to the set of endpoints configured in the business service	Select the **Enabled** check box next to **Throttling State**. When enabled, messages are processed by priority. You can optionally assign messages a priority using routing options; otherwise, messages are de-queued on a first-in, first-out basis. For more information, see "Throttling" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
Restrict the number of messages being dispatched to a set of endpoints configured in the business service	If **Throttling State** is enabled, this field is required. Specify a **Maximum Concurrency** value; it must be a positive integer value, it cannot be zero. If exceeded, incoming messages are put into a wait (throttling) queue, if one exists. If not, the messages are discarded. If the queue is full, messages in the queue with a lower priority are removed and a runtime error is raised. If an error handler is configured, it will be triggered. The new incoming messages will be enqueued.
Specify whether messages sent to business services that have exceeded their message concurrency limit should be held in a wait queue Set the maximum number of messages that are held in the wait queue	If **Throttling State** is enabled, this field is required. Specify a **Throttling Queue** value; it must be a positive integer value or keep the default value, zero, meaning there is no throttling queue. When the throttling queue value is zero (no throttling queue) and the number of messages exceed the **Maximum Concurrency** value, the messages are discarded and an error is raised. If an error handler is configured, it will be triggered. If you disable throttling at runtime, all the messages in the throttling queue will be processed. When you delete or rename a business service, all the messages in the throttling queue are discarded.
Specify the maximum length of time that messages will be held in a wait queue	If **Throttling State** is enabled, this field is optional. Specify a **Message Expiration** value in milliseconds; it must be a positive integer value. A value of zero means the message will never be expired. This value is set at the queue level; all messages put in the queue have the same **Message Expiration** value. When exceeded, messages are removed from the queue without being processed.

Table 26–9   (Cont.) Configuring Business Service Operational Settings

To...	Do This...
Enable message tracing for a business service	Select the **Enabled** check box next to **Message Tracing**. After you enable message tracing, the system logs messages exchanged between the Oracle Service Bus pipeline and the business service (outbound request and response messages). **Note**: To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels: ■ Minimum severity to log: Info ■ Log file: Info ■ Standard out: Info For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*. When applicable, logged outbound messages can also include the retry number, error code, and error message. From the **Detail Level** list, choose to specify the level of detail from among the following: ■ Terse—Display the date, time, service type, service name, and URI ■ Headers—Display terse information along with the XML representation of the message metadata ■ Full—Display the headers information along with the raw payload, including attachments if any If you choose Full from the **Detail Level** list, specify the maximum size (in kilobytes) for the message payload using the **Payload Tracing Limit** field. Also, specify the default encoding for logging the payload using the **Default Encoding** field. This can be useful when logging binary payloads or SOAP messages with binary attachments. The default encoding value can be Base64 or any Java-supported encoding as defined in: http://docs.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html. Leaving the **Default Encoding** field empty causes Oracle Service Bus to use the host's default encoding for the payload. The default encoding depends on a combination of the JVM, the underlying operating system (OS), and OS-level locale settings. If the setting specified in the **Default Encoding** field cannot be used (for example, it is not a valid option for the configuration), then Oracle Service Bus uses Base64 encoding for the payload.
Disable message tracing for a business service	Clear the **Enabled** check box next to **Message Tracing**.
Enabling and Disabling Result Caching	Selecting **Enabled** activates result caching for the business service. You must also ensure that result caching is enabled globally. See Section 27.2.11, "Enabling Result Caching Globally." When you deselect the **Enabled** option and click **Update** to disable result caching for a business service, all results cached for that business service are flushed.

26.10 Setting the Aggregation Interval for a Service

Use the Operational Settings page to set the aggregation interval for a specific service. The aggregation interval is the period over which aggregated statistics are computed for display in the Administration Console. The default aggregation interval setting is 10 minutes.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Business Services** or **Proxy Services**.

3. Click the service name.

4. Select the **Operational Settings** tab.

5. In the **Aggregation Interval** field, select the length of the aggregation interval in hours and minutes, then click **Update**.

   If your selection for hours exceeds 1, then the default selection for minutes is always zero. However, if your selection for hours is 0 or 1, then you can configure intervals in terms of minutes.

6. Click **Update** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

26.11 Viewing Service Metrics

Use this page to view the monitoring details of specific services, shown in Table 26–9, and the Service Metrics shown in Table 26–11.

Table 26–10   Service Metric Information

Property	Description
Service Name	The name and path of the service.
Service Type	The parent service type: proxy service or business service.
Display Statistics	To display monitoring statistics for the service for the period of the current aggregation interval, select **Current Aggregation Interval**. The Current Aggregation Interval view displays a moving statistic view of the service metrics.
	To display monitoring statistics for this service for the period since you last reset statistics, select **Since Last Reset**. The Since Last Reset view displays a running count view of the metrics.
Server	Select a server from the list to display metrics for that server.
	If a cluster exists, cluster-wide metrics are displayed by default. Select an individual Managed Server to display metrics for that server.
	If you display metrics for an individual Managed Server in a cluster, the **SLA Alert Count** will be N/A (only cluster-wide SLA Alert Counts are displayed).
Aggregation Interval	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the current aggregation interval set for monitoring this service, in terms of hours and minutes. You set this interval on the Operational Settings page. See Section 26.10, "Setting the Aggregation Interval for a Service."

Table 26–10   (Cont.) Service Metric Information

Property	Description
Reset Statistics	If you have selected **Since Last Reset** in the **Display Statistics** field, you can reset the monitoring statistics for this service. Click **Reset Statistics** at the bottom of the page.

Table 26–11   Service Metrics

Property	Description
SLA Alert Count	■ The alert count is based on the Display Statistics view, either for the **Current Aggregation Interval** or for the period **Since Last Reset**. ■ The alert severity and timestamp are displayed only in the Current Aggregation Interval view. For the Since Last Reset view, only the count value is displayed. ■ The alert severity for the most severe alert within the current aggregation interval. If there are multiple alerts, the highest alert severity in the current aggregation interval is shown. If there are no alerts in the current aggregation interval, NORMAL is displayed. ■ The timestamp displays the time the server was last polled for alert information and displays the alert status at the time the server was last polled. If there were no alerts when the server was last polled, NORMAL is displayed and the polling time is shown.
Pipeline Alert Count	For proxy services only: ■ The pipeline alert count is based on the Display Statistics view, either for the **Current Aggregation Interval** or for the period **Since Last Reset**. ■ The alert severity and timestamp are displayed only in the Current Aggregation Interval view. For the Since Last Reset view, only the count value is displayed. ■ The alert severity for the most severe alert within the current aggregation interval. If there are multiple alerts, the highest alert severity in the current aggregation interval is shown. If there are no pipeline alerts in the current aggregation interval, NORMAL is displayed. ■ The timestamp displays the time the server was last polled for pipeline alert information and displays the alert status at the time the server was last polled. If there were no alerts when the server was last polled, NORMAL is displayed and the polling time is shown.
Min Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the minimum response time this service has taken to execute messages within the period of the current aggregation interval. If you have selected **Since Last Reset** in the **Display Statistics** field, the minimum response time this service has taken to execute messages within the period since you last reset statistics.

Table 26–11   (Cont.) Service Metrics

Property	Description
Max Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the maximum response time this service has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the maximum response time this service has taken to execute messages within the period since you last reset statistics.
Overall Avg. Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the overall average time (in milliseconds) this service has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the overall average time this service has taken to execute messages within the period since you last reset statistics.
Message Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the total number of messages that this service has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the total number of messages that this service has executed within the period since you last reset statistics.
Error Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of error messages that this service has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with errors that this service has executed within the period since you last reset statistics.
Failover Count	For business services only:
	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of failovers that this service has attempted within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of failovers that this service has attempted within the period since you last reset statistics.
Success Ratio	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the success ratio of this service within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the success ratio of this service within the period since you last reset statistics.
	For example, if the service has executed 9 out of 10 messages successfully, then the success ratio is 90%.
Failure Ratio	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the failure ratio of this service within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the failure ratio of this service within the period since you last reset statistics.
	For example, if the service has failed to execute 1 out of 10 messages, then the failure ratio is 10%.

Table 26–11 (Cont.) Service Metrics

Property	Description
WS Security Errors	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of messages with WS security errors that this service has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with WS security errors that this service has executed within the period since you last reset statistics.
Validation Errors	For business services: N/A is displayed.
	For proxy services:
	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of messages with validation errors that this service has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with validation errors that this service has executed within the period since you last reset statistics.
Hit Count	For business services only:
	Hit Count displays the number of times a business service result cache has been used.
Hit Ratio	The sum of Hit Count and Message Count.
Min Throttling Time	For business services only:
	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the least amount of time that messages have spent in a throttling queue within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the least amount of time that messages have spent in a throttling queue within the period since you last reset statistics.
Max Throttling Time	For business services only:
	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the maximum amount of time that messages have spent in a throttling queue within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the maximum amount of time that messages have spent in a throttling queue within the period since you last reset statistics.
Average Throttling Time	For business services only:
	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the average amount of time that messages have spent in a throttling queue within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the average amount of time that messages have spent in a throttling queue within the period since you last reset statistics.

26.12 Viewing Operations Metrics for WSDL-Based Services

Use this page to view the service monitoring information, shown in Table 26–10, and the Operations metrics for WSDL-based services, shown in Table 26–12.

Table 26–12   Operations Metrics

Property	Description
Operation Name	The name of the operation associated with this service.
Message Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the total number of messages that this operation has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the total number of messages that this operation has executed within the period since you last reset statistics.
Error Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of messages with errors that this operation has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with errors that this operation has executed within the period since you last reset statistics.
Min Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the minimum response time this operation has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the minimum response time this operation has taken to execute messages within the period since you last reset statistics.
Max Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the maximum response time this operation has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the maximum response time this operation has taken to execute messages within the period since you last reset statistics.
Avg. Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the overall average time this operation has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the overall average time this operation has taken to execute messages within the period since you last reset statistics.

26.13 Viewing Pipeline Metrics

Use this page to view the service monitoring information, shown in Table 26–10, and the Pipeline metrics for proxy services that have pipelines, shown in Table 26–13.

Table 26–13   Pipeline Metrics

Property	Description
Component Name	The name of the component.

Table 26–13   (Cont.) Pipeline Metrics

Property	Description
Message Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the total number of messages that this component has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the total number of messages that this component has executed within the period since you last reset statistics.
Error Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of messages with errors that this component has executed within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with errors that this component has executed within the period since you last reset statistics.
Min Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the minimum response time this component has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the minimum response time this component has taken to execute messages within the period since you last reset statistics.
Max Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the maximum response time this component has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the maximum response time this component has taken to execute messages within the period since you last reset statistics.
Avg. Response Time (msecs)	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the overall average time this component has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the overall average time this component has taken to execute messages within the period since you last reset statistics.

26.14 Viewing Action Metrics

Use this page to view the service monitoring information, shown in Table 26–10, and the Action metrics for proxy services that have pipelines, shown in Table 26–14.

Table 26–14   Action Metrics

Property	Description
Node	A hierarchical display of the nested actions.

Table 26–14   (Cont.) Action Metrics

Property	Description
Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of invocations within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of invocations within the period since you last reset statistics.
	This is different than the number of messages since an action can be invoked many times per message.
Avg. Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the average execution time per invocation within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the average execution time per invocation within the period since you last reset statistics.
Total Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the total execution time within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the total execution time within the period since you last reset statistics.

26.15 Viewing Business Services Endpoint URIs Metrics

Use this page to view the service monitoring information, shown in Table 26–10, and the Endpoint URIs metrics for business services, shown in Table 26–15. For more information, see "Managing Endpoint URIs for Business Services" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Table 26–15   Endpoint URIs Metrics

Property	Description
Endpoint URI	The endpoint URL associated with this business service.
Message Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the total number of messages executed by this endpoint URI within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the total number of messages executed by this endpoint URI within the period since you last reset statistics.
Error Count	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the number of messages with errors executed by this endpoint URI within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the number of messages with errors executed by this endpoint URI within the period since you last reset statistics.

Table 26–15   (Cont.) Endpoint URIs Metrics

Property	Description
Min Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the minimum response time this endpoint URI has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the minimum response time this endpoint URI has taken to execute messages within the period since you last reset statistics.
Max Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the maximum response time this endpoint URI has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the maximum response time this endpoint URI has taken to execute messages within the period since you last reset statistics.
Avg. Response Time	If you have selected **Current Aggregation Interval** in the **Display Statistics** field, the overall average time this endpoint URI has taken to execute messages within the period of the current aggregation interval.
	If you have selected **Since Last Reset** in the **Display Statistics** field, the overall average time this endpoint URI has taken to execute messages within the period since you last reset statistics.
Status	The status of the business service endpoint URI.
	For single server domains or for an individual Managed Server in a cluster: Online or Offline.
	For cluster-level statistics: Online, Offline, or Partial, when that particular URI is offline on at least one of the Managed Servers in the cluster.
Action	Click the **Mark Endpoint Online** icon to mark an offline URI, Online. This action is enabled only if the URI has an Offline or Partial status.
	To enable this action, a business service must be enabled for taking URIs offline and its service and global-level monitoring operational settings must be enabled. See Section 26.9, "Configuring Operational Settings for Business Services."
	Note: If successful, the action is immediate. However, reflecting that its endpoint URI status is now online takes one monitoring aggregation cycle, which might take as long as 1 minute and a half.

26.16 Viewing Server Information

The Server Health page includes the following:

- On the Server Health tab header, a Shutdown icon and numeric indicator which shows the number of servers that are shutdown, may be present:
  - Click the icon to filter the servers displayed using the `Shutdown` **Health** criteria and `SHUTDOWN` **State** criteria.
  - The number indicates the number of shutdown servers.

- A pie chart that depicts the health of the servers in the domain. You can click the area in the chart that corresponds to the server to display more details.

- The Log Summary. To display the log, click the numeric link that displays the number of messages for a given severity. For example, Alert:1, click the number 1. See Section 26.20, "Viewing Details of Domain Log Files."

- Fields that display server information, depending on your customized settings for the Server Summary table. For example, server name, health, cluster name, computer name, the state of the server, and the length of time the server has been running. To learn more about these fields, see Table 26–16.

- By default, the Dashboard refresh rate is **No Refresh**. To change the Dashboard refresh rate interval, see Section 27.3, "Setting User Preferences."

> **Caution:** By selecting a dashboard refresh rate interval other than the default (**No Refresh**), the browser refreshes the Dashboard at regular intervals. This prevents your session from timing out, even without you interacting with the Administration Console.
>
> Oracle recommends that you use the Dashboard refresh rate feature with caution. You should never leave an Administration Console session unattended. While your Administration Console session is active others can gain access to it from your computer, without re-authentication.

- A Table Customizer icon. See Section 2.2, "Customizing Table Views."

- To restrict the number of items in the list or locate specific servers:

    - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match *all* the criteria will be returned.

    - Filter by health status. From the **Health** list, select to restrict servers to the specified health status level, then click **Search**.

      Select the **or above** check box to restrict your search to the specified server health status level or above (listed from the most inclusive to the most restrictive level): OK, Warning, Overloaded, Critical, and Fatal.

    - Filter by server. From the **Server** list, select All or the name of the server, then click **Search**.

    - Filter by cluster name. From the **Cluster Name** list, select All or the name of the cluster associated with the server, then click **Search**.

    - Filter by computer name. From the **Machine Name** list, select All or the name of the computer associated with the server, then click **Search**.

    - Filter by state. From the **State** list, select All or the state of the server: RUNNING, FAILED, or SHUTDOWN, then click **Search**.

    - Click **View All** to remove the search filters and display all servers.

The Server Health page displays the following information for each server, shown in Table 26–16, depending on the filter settings you have specified.

Table 26–16   Server Summary Details

Property	Description
Server	The name of the server. The name is a link to the View Server Details page. See Section 26.17, "Viewing Server Details."
Health	The health status of the server: ■ OK—the server is functioning without any problems. ■ OK/Not Processing—the server is in the ADMIN state, available only for administration operations. All runtime Oracle Service Bus messages will fail. ■ Warning—the server might have problems in the future. Check the server logs and the corresponding runtime MBean for more details. ■ Overloaded—the server has more work assigned to it than the configured threshold; it might refuse more work. ■ Critical—server failure pending; something must be done immediately to prevent failure. Check the server logs and the corresponding runtime MBean for more details. ■ Shutdown—the server has failed and must be restarted. ■ Critical—server failure pending; something must be done immediately to prevent failure. Check the server logs and the corresponding runtime MBean for more details.
Cluster Name	The name of the cluster associated with this server.
Machine Name	The name of the computer associated with this server.
State	The state of the server: ■ RUNNING ■ ADMIN ■ FAILED ■ SHUTDOWN
Uptime	The length of time this server has been running.

26.17 Viewing Server Details

The objects displayed on this page are WebLogic Server entities. To learn more about these entities, see the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

26.18 Viewing Domain Log Files

This page displays a summary of domain log file entries, including the information shown in Table 26–17.

■ To customize the display of domain log file information, see Section 26.19, "Customizing Your View of Domain Log File Entries."

■ To display details of a specific log file, select the option for the appropriate entry, then click **View**. See Section 26.20, "Viewing Details of Domain Log Files."

Table 26–17   Domain Log Summary Information

Property	Description
Date	The date the entry was logged.

Table 26–17   (Cont.) Domain Log Summary Information

Property	Description
Subsystem	The subsystem associated with the entry.
Severity	The severity of the message.
Message ID	The unique identification for the message.
Message	The message description.

26.19 Customizing Your View of Domain Log File Entries

To customize the view of domain log file entries:

1. Click **Customize this table**. Additional fields are displayed.

    You can click the **Customize this table** link at any time to close this table and retain the original settings.

2. In the **Filter** fields, do the following:

    a. In the **Time** field, select the time interval for which you want to view log entries.

    b. In the **Start Time** field, enter a start time in the format `MM/DD/YYYY HH:MM:SS`. For example, enter 10/25/06 08:39:48. You use the **Start Time** and **End Time** fields to specify a window of time for which you want to view log entries.

    c. In the **End Time** field, enter a end time in the format `MM/DD/YYYY HH:MM:SS`. For example, enter 10/25/06 13:20:51. You use the **Start Time** and **End Time** fields to specify a window of time for which you want to view log entries.

    d. In the **Criteria** text box, specify a filtering criteria (a text string). The filtering criteria is specified as a string in the WLDF Query Language. The query language supports Boolean operators: AND, OR, and NOT, and relational operators. For more information on query syntax, see "WLDF Query Language" in *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

3. In the **Columns** field, select the columns you want to display:

    a. Select a column name from the **Available** field.

    b. Click the arrow to move this column name to the **Chosen** field.

    c. Repeat until you have listed all the column names you want to display in the **Chosen** field.

        In the **Chosen** field, you can use the Up and Down arrows to reorder the column names as required.

4. In the **Number of rows displayed per page** field, select the number of log entries you want to display on a single page.

5. In the **Maximum Results Returned** field, select the maximum number of log entries you want to display in total or select **Show All** to display all of them.

6. Do one of the following:

    ▪ To save the new settings, click **Apply**.

    ▪ To discard your changes and retain the original settings, click **Reset**.

26.20 Viewing Details of Domain Log Files

Use this page to view details of domain log file entries, shown in Table 26–18.

Table 26–18   Domain Log Details

Property	Description
Message	A description of the event or condition.
Date	Displays the time and date when the message originated, in a format that is specific to the locale. The Java Virtual Machine (JVM) that runs each WebLogic Server instance refers to the host computer operating system for information about the local time zone and format.
Subsystem	Indicates the subsystem of WebLogic Server that was the source of the message; for example, Enterprise Java Bean (EJB) container or Java Messaging Service (JMS).
Message ID	A unique six-digit identifier.
	All message IDs that WebLogic Server system messages generate start with BEA- and fall within a numerical range of 0-499999.
Severity	Indicates the degree of impact or seriousness of the event reported by the message:
	■ Alert—A particular service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; the immediate attention of the administrator is needed to resolve the problem.
	■ Critical—A system or service error has occurred. The system can recover but there might be a momentary loss or permanent degradation of service.
	■ Emergency—The server is in an unusable state. This severity indicates a severe system failure or panic.
	■ Error—A user error has occurred. The system or application can handle the error with no interruption and limited degradation of service.
	■ Info—Used for reporting normal operations; a low-level informational message.
	■ Notice—An informational message with a higher level of importance.
	■ Warning—A suspicious operation or configuration has occurred but it might not affect normal operation.
Machine	The DNS name of the computer that hosts the server instance.
Server	The name of the WebLogic Server instance on which the message was generated.
Thread	The ID that the JVM assigns to the thread in which the message originated.
User ID	The user ID under which the associated event was executed.
Transaction ID	Present only for messages logged within the context of a transaction.
Context ID	Context information to correlate messages coming from a specific request or application.
Detail	A description of the event or condition.
Cause	The cause of the message.
Action	The action that should be taken.

26.21 Viewing Alert Rules

Use this page to view the alert rules information shown in Table 26–19.

Table 26–19   Alert Rules Information

Property	Description
Name	The name assigned to this alert rule. The name is a link to the Alert Rule General Configuration page. See Section 26.24, "Viewing Alert Rule Configurations."
SLA State	The status of the alert rule: Enabled or Disabled.
Description	This field is hidden by default. A description of the alert rule.
Severity	The severity of the alert that will be triggered by this rule: Normal, Warning, Minor, Major, Critical, or Fatal.
Aggr. Interval	The length of the aggregation interval in terms of hours and minutes.
Expiration Date	The date when this alert rule is no longer in effect.
Stop Processing	Displays Yes or No.
Frequency	The frequency of this alert: ■ Every Time ■ Notify Once
Options	This column includes the following functionality: ■ Click the arrows to reorder the alert rules. You can move individual alert rules up or down the list. You can do this only when this more than one alert rule configured for this service. ■ Click the **Rename** icon to rename an alert rule. ■ Click the **Delete** icon to delete a specific alert rule. See Section 26.26, "Deleting Alert Rules."

26.22 Locating Alert Rules

To locate alert rules:

1. Select **Resource Browser** > **Business Services** or **Proxy Services**.

2. Click the service name.

3. Select the **SLA Alert Rules** tab.

4. To locate a specific alert rule for this service, scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

You can use this page also to do the following:

■ To configure monitoring for this service, select the **Operational Settings** tab. See Section 26.8, "Configuring Operational Settings for Proxy Services" and Section 26.9, "Configuring Operational Settings for Business Services."

■ To add a new alert rule, click **Add New**. See Section 26.23, "Creating and Editing Alert Rules."

26.23 Creating and Editing Alert Rules

The tasks in creating and editing alert rules include:

- Section 26.23.1, "Configuring General Information for Alert Rules"
- Section 26.23.2, "Defining Alert Rule Conditions"
- Section 26.23.3, "Reviewing the Alert Rule Configuration"

Note: When a service is created from another service, alert rules are maintained in the following way:

- When a proxy service is created from a business service or a business service is created from a proxy service, the alert rules, if any, are removed.

- When a proxy service is created from another proxy service or a business service is created from another business service, the alert rules, if any, are retained.

26.23.1 Configuring General Information for Alert Rules

To configure general information for alert rules:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Resource Browser** > **Proxy Services** or **Business Services**.

3. Click the service name.

4. Select the **SLA Alert Rules** tab and click **Add New**.

5. In the **Rule Name** field, enter a name for the alert rule. This is a required field.

6. In the **Alert Summary** field, enter a short description of the alert rule. This text is also used as the subject line for an email message if this alert rule is configured with an email alert notification, and can contain no more than 80 characters. If you do not provide an alert summary, the default text, Oracle Service Bus Alert, will be used instead.

7. In the **Rule Description** field, enter a description for the alert rule.

8. In the **Alert Destination** field, enter the name of the alert destination resource or click **Browse** to display the Select Alert Destination page. Select an alert destination from the list and click **Submit**.

   By default, the alert will always go to the Administration Console. To configure and select other alert destination resources, see Section 6.2, "Adding Alert Destinations" and Section 6.1, "Locating Alert Destinations."

9. In the **Start Time** field, enter a start time in the format `HH:MM`. For example, enter 09:00 AM. You use the **Start Time** and **End Time** fields to specify a window of time during which the rule is active on each day prior to the expiration date. For example, between 9am and 5pm every day.

10. In the **End Time** field, enter an end time in the format `HH:MM`. For example, enter 05:00 PM. You use the **Start Time** and **End Time** fields to specify a window of time during which the rule is active on each day prior to the expiration date. For example, between 9am and 5pm every day.

    The alert rule is active daily during the start time you specified until the end time you specified, until the rule expires.

11. In the **Rule Expiration Date** field, enter an expiration date in the format `MM/DD/YYYY`. For example, enter 12/31/2010. The rule expires at 11:59 pm on the specified date. If you do not specify a date, the rule never expires.

12. In the **Rule Enabled** field, keep **Yes** as the default to ensure that this rule is enabled, or select **No** to disable this rule.

13. In the **Alert Severity** field, select one of the following:

    - Normal

    - Warning

    - Minor

    - Major

    - Critical

    - Fatal

14. In the **Alert Frequency** field, select one of the following settings, shown in Table 26–20.

Table 26–20   Alert Frequency Setting

Property	Description
Every Time	If you select this option, the actions included in the alert rule are executed every time the alert rule evaluates to `true`. For example, if you set the condition that the average response time is greater than 300 milliseconds, you receive an alert every time this condition evaluates to `true`.
	The number of times an alert rule is evaluated depends on the aggregation interval and the sample interval associated with that rule. If the aggregation interval is set to 5 minutes, the sample interval is 1 minute. Rules are evaluated each time 5 samples of data are available. Therefore, the rule is evaluated for the first time approximately 5 minutes after it is created and every minute thereafter.
Notify Once	If you select this option, the actions included in the rule are executed the first time the rule evaluates to `true`, and no more alerts are generated until the condition resets itself and evaluates to `true` again. For example, if you set the condition that the average response time is less than 300 milliseconds, you receive an alert the first time this condition evaluates to `true`, but you do not receive any more alerts until the condition evaluates to `false` and then to `true` again. The alert timestamp gets updated and is displayed on the Dashboard.

15. Select **Yes** for the **Stop Processing More Rules** option to abort executing further rules after one of the rules associated with a service evaluates to `true`. Use this option to stop evaluating subsequent rules when there are multiple rules associated with a particular service. Keep **No** as the default to continue processing rules.

    This option behaves like the **Stop Processing More Rules** option in the Rules Wizard in Microsoft Outlook.

After you finish

Click **Next** to display the New Alert Rule - Conditions Configuration page. Continue in Section 26.23.2, "Defining Alert Rule Conditions."

26.23.2 Defining Alert Rule Conditions

Use this page to define conditions for the alert rule. You must specify at least one condition. If you specify multiple conditions, you must use the **And/Or** operators to combine them.

1. In the **Select Aggregation Interval for the Condition** field, select the number of hours and minutes to set the aggregation interval for this condition. The aggregation interval determines the frequency at which the condition is tested. The condition is tested each time the monitoring subsystem aggregates enough samples of data to constitute 1 aggregation interval.

   For example, if you select an aggregation interval of 1 hour, the condition is tested each time an hour's worth of data is available. The first time the condition is tested is at the end of the first hour. After that, the condition is tested every 10 minutes because the sampling interval for an aggregation interval of 1 hour is set to 10 minutes.

2. You start by defining a simple expression. Two or more simple expressions can be combined to form a complex expression. To define a complex simple expression, do the following:

   a. In the first field, select **Count**, **Minimum**, **Maximum**, **Average**, or **Status**.

   b. In the next field, select an operand.

      Depending on whether you select **Count**, **Minimum**, **Maximum**, **Average**, or **Status** in the first field, the list of the operands varies. For example, if you select Minimum, Maximum, or Average, the Response Time operand is available. Use this operand to set the response time in milliseconds (msec). The operands available also depends on the configuration of the service itself. The number of operands varies according to whether a service has pipelines, route nodes, operations, and so on.

      For information on how to generate alerts based on endpoint URI status, see "Managing Endpoint URIs for Business Services" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

      When you select **Count**, this field displays the operands shown in Table 26–21.

Table 26–21   Count Operand Details

Property	Description
Cache Hit Count	For business services that use result caching, Cache Hit Count increments each time the cache is used to return a response to a client.
Error Count	The error count gets incremented when message processing returns a failure.
Failure Ratio (%)	The ratio of errors encountered to the total number of messages successfully processed over the specified aggregation interval.
Message Count	The total number of messages processed.
Success Ratio (%)	The ratio of messages successfully processed to the total number of messages encountered over the specified aggregation interval.
Failover count	For business services only: The failover count.
<Request Pipeline>.Error Count	For proxy services only: The number of erroneous messages processed by the request pipeline.
<Request Pipeline>.Message Count	For proxy services only: The number of messages processed by the request pipeline.

Table 26–21 (Cont.) Count Operand Details

Property	Description
Validation Error Count	For proxy services only that have a validate action in the pipeline: The number of validation errors.
WSS Error Count	Available depending on the transport for the service (such as with HTTP): The number of Web Service Security (WSS) erroneous messages processed. This counter is only available for WSDL-based services and is updated when a WSS error is encountered.

3. In the next field, select the desired comparison operator: **=**, **!=**, **>** or **<**.

4. In the next field, enter the value to compare against.

5. Click **Add**.

6. To define a complex expression:

    a. Repeat steps a - e in step 2 to define a simple expression.

    b. Repeat steps a - e again to define another simple expression.

    c. Select the check box for each of the two expressions.

    d. Click **And** or **Or** to combine the expressions into a complex expression.

    The **Split** option is available after you combine multiple expressions. This option is used to split complex expressions back into separate simple expressions.

After you finish

Click **Prev** to return to the General Configuration page. Or, click **Last** to display the New Alert Rule - [service name] page. Continue in Section 26.23.3, "Reviewing the Alert Rule Configuration."

26.23.3 Reviewing the Alert Rule Configuration

Use this page to review the configuration data that you have entered for this alert rule. If necessary, you can click **Edit** to make changes to the configuration before you save the alert rule.

Do any of the following:

- To make a change to one of the configuration pages, click the **Edit** icon for the appropriate page.

- To return to the previous page, click **Prev**.

- To create the alert, click **Save**. The alert rule is created and saved in the current session.

- To disregard changes, click **Cancel**.

- To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

26.24 Viewing Alert Rule Configurations

Use this page to view alert rule configuration details, shown in Table 26–22.

Table 26–22   Alert Rule - General Configuration

Property	Description
Rule Name	The name of the alert rule.
Rule Description	A description of the alert rule.
Alert Summary	A short description of the alert rule. This text is also used as the subject line for an email message if this alert rule is configured with an email alert notification.
Alert Destination	The path and name of the alert destination resource that contains the destination address for alert notifications. This field is a link to the View Alert Destination - General Configuration page. See Section 6.4, "Editing Alert Destinations."
Start Time (HH:MM)	The start time for this alert rule.
End Time (HH:MM)	The end time for this alert rule.
Rule Expiration Date (MM/DD/YYYY)	The date when this alert rule no longer applies.
Rule Enabled	An indicator of whether this alert rule is enabled.
Alert Severity	The severity of the alert raised as a result of this alert rule: ■ Normal ■ Warning ■ Minor ■ Major ■ Critical ■ Fatal
Alert Frequency	The frequency of the alert: ■ Every Time ■ Notify Once
Stop Processing More Rules	An indicator of whether the alert stops processing more rules.

The page displays the Conditions information shown in Table 26–23.

Table 26–23   Alert Rule Conditions

Property	Description
Condition Expression	The aggregation interval for this condition, in terms of hours and minutes. A list of one or more conditions that govern this alert rule.

26.25 Editing Alert Rules

To edit alert rules:

1.   If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2.   Locate the alert rule. See Section 26.22, "Locating Alert Rules."

3.   Click the alert rule name.

4.   To make a change to the fields on the configuration pages, click the **Edit** icon for the appropriate page or the **Edit** button at the bottom of the page. See

Section 26.23, "Creating and Editing Alert Rules" for descriptions of the pages and fields.

5. Do any of the following:

    ■   To go to the next page, click **Next**.

    ■   To return to the previous page, click **Prev**.

    ■   To go to the summary page, click **Last**.

    ■   To commit the updates in the current session, click **Save**.

    ■   To disregard changes and return to the SLA Alert Rules page, click **Cancel**.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

26.26 Deleting Alert Rules

To delete alert rules:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the alert rule. See Section 26.22, "Locating Alert Rules."

3. Click the **Delete** icon in the **Options** field of the alert rule you want to delete. The alert rule is deleted in the current session.

4. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27

Configuration

This chapter describes how to find and update operational settings in the Oracle Service Bus (OSB) Administration Console, as well as configure global setting such as monitoring, alerts, reporting, logging, and result cache. This chapter also describes how to set display options in the OSB Administration Console.

You use the Configuration module to locate and manage proxy services, business services, alert destinations, SLA alert rules, and the runtime tracing for proxy services.

You can specify operational settings for all services, at the service and global level, and use the global settings to turn on and off monitoring, SLA alerts, and, for proxy services only, pipeline message reporting and pipeline message logging.

27.1 Finding and Updating Operational Settings

Use the Smart Search page to easily locate proxy services, business services, alert destinations, and SLA alert rules. Additionally, you can use the Smart Search page to specify service-specific operational settings, with the following exceptions: you cannot set an aggregation interval or specify an alerting or logging severity level. For more information, see Section 26.8, "Configuring Operational Settings for Proxy Services" and Section 26.9, "Configuring Operational Settings for Business Services."

The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and the service-level settings for a service to be completely enabled at runtime. Additionally, the service **State** must also be enabled. See Section 27.2, "Enabling Global Settings."

You can enable or disable operational settings only from within a session.

27.1.1 Finding All Services (Proxy and Business Services)

To find all proxy and business services:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **All Services**, then click **Search**. The Summary of All Services page displays the information shown in Table 27–1.

3. To restrict the number of items in the list or locate specific services:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - Filter by service name. In the **Name** field, enter the name of the search target or enter wildcard characters (use * and ? as wildcard characters to perform a more general search), then click **Search**.

This search method is preferable if the name of the service is unique across all projects and paths.

- Filter by service path. In the **Path** field, enter the path of the search target, which is the project name and the name of the folder in which services reside. You can use * and ? as wildcard characters to perform a more general search. All the services that reside in that path are displayed.

   The format for the **Path** field is as follows:

   ```
   project-name/root-folder/ . . ./parent-folder
   ```

   If a service is directly under the project, the format is as follows:

   ```
   project-name
   ```

- Filter by service name and path. This search method is preferable if there is more than one service with the same name that reside in different paths.

- Filter by service state. From the **State** list, select the state of the service: Enabled or Disabled, then click **Search**. Select All to ignore the service state filter criteria.

- Filter by service monitoring. From the **Monitoring** list, select the monitoring status of the service: Enabled or Disabled, then click **Search**. Select All to ignore the service monitoring filter criteria.

- Filter by service's SLA Alerts setting. From the **SLA Alerts** list, select the SLA alerts setting of the service: Enabled or Disabled, then click **Search**. Select All to ignore the service SLA alerts filter criteria.

Table 27–1   Services Summary Information

Property	Description
Name	The name assigned to the service. The name is a link to the Operational Settings page. See Section 26.8, "Configuring Operational Settings for Proxy Services" or Section 26.9, "Configuring Operational Settings for Business Services."
Path	The project associated with the service. If the service resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Type	The parent service type: proxy service or business service.
State	The state of the service: Enabled or Disabled.
Monitoring	The monitoring status of the service: Enabled or Disabled.
SLA Alerts	The SLA alerts status: Enabled or Disabled, and the level enabled at and above: Normal (N), Warning (W), Minor (Mn), Major (Mj), Critical (C), or Fatal (F).
Pipeline Alerts	For proxy services only: The pipeline alerts status: Enabled or Disabled, and the level enabled at and above: Normal (N), Warning (W), Minor (Mn), Major (Mj), Critical (C), or Fatal (F).
Reports	For proxy services only: The message reporting status of the service: Enabled or Disabled.

Table 27–1   (Cont.) Services Summary Information

Property	Description
Logs	For proxy services and Split-Joins only:
	The logging status: Enabled or Disabled, and the level enabled at and above: Debug (D), Info (I), Warning (W), or Error (E).
Tracing	For proxy services and Split-Joins only:
	The runtime tracing status of the proxy service: Enabled or Disabled.
Caching	For business services only. Whether result caching is Enabled or Disabled.
Actions	For proxy services: The Edit Message Flow icon is a link to the pipeline for that proxy service.

You can use this page also to do the following:

- To remove the search filter and display all services, click **View All**.

- To discard your changes and refresh the page with the currently stored settings using the same search criteria, click **Reset**.

- From within a session, to enable or disable any service-level settings, select or clear the check box next to it and click **Update**.

> **Note:**   The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and the service-level settings for a service to be completely enabled at runtime. Additionally, SLA Alerts depends on both the global and service-level Monitoring states—both must be enabled for SLA alerts to be enabled at runtime. See Section 27.2, "Enabling Global Settings."
>
> In addition, the service **State** must also be enabled.

- To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.1.2 Finding Proxy Services

To find proxy services:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **Proxy Services**, then click **Search**. The Summary of Proxy Services page displays the information shown in Table 27–2.

3. To restrict the number of items in the list or locate specific proxy services:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - Filter by proxy service name and path. In the **Name** and **Path** fields, enter the name and path of the search target, then click **Search**. The path is the project name and the name of the folder in which the proxy service resides. You can use * and ? as wildcard characters to perform a more general search.

   - Filter by service state. From the **State** list, select the proxy service state: Enabled or Disabled, then click **Search**. Select All to ignore the service state filter criteria.

- Filter by service monitoring. From the **Monitoring** list, select the monitoring status of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service monitoring filter criteria.

- Filter by service's SLA Alerts setting. From the **SLA Alerts** list, select the SLA alerts setting of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service SLA alerts filter criteria.

- Filter by service's pipeline alerts setting. From the **Pipeline Alerts** list, select the pipeline alerts setting of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service pipeline alerts filter criteria.

- Filter by services with message reports. From the **Message Reports** list, select the message reports status of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service message reports filter criteria.

- Filter by services with logging. From the **Logs** list, select the logging status of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service logging filter criteria.

- Filter by services with runtime tracing. From the **Tracing** list, select the runtime tracing status of the proxy service: Enabled or Disabled, then click **Search**. Select All to ignore the service tracing filter criteria.

Table 27–2   Proxy Services Summary Information

Property	Description
Name	The name assigned to the proxy service. The name is a link to the Operational Settings page. See Section 26.8, "Configuring Operational Settings for Proxy Services."
Path	The project associated with the proxy service. If the service resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Type	The parent service type: proxy service.
State	The state of the proxy service: Enabled or Disabled.
Monitoring	The monitoring status of the proxy service: Enabled or Disabled.
SLA Alerts	The SLA alerts status: Enabled or Disabled, and the level enabled at and above: Normal (N), Warning (W), Minor (Mn), Major (Mj), Critical (C), or Fatal (F).
Pipeline Alerts	The pipeline alerts status: Enabled or Disabled, and the level enabled at and above: Normal (N), Warning (W), Minor (Mn), Major (Mj), Critical (C), or Fatal (F).
Reports	The message reports status of the service: Enabled or Disabled.
Logs	The logging status: Enabled or Disabled, and the level enabled at and above: Debug (D), Info (I), Warning (W), or Error (E).
Tracing	The runtime tracing status of the proxy service: Enabled or Disabled.
Actions	The Edit Message Flow icon is a link to the pipeline for that proxy service.

You can use this page also to do the following:

- To remove the search filters and display all proxy services, click **View All**.

- To discard your changes and refresh the page with the currently stored settings using the same search criteria, click **Reset**.

- From within a session, to enable or disable any service-level settings, select or clear the check box next to it and click **Update**.

> **Note:** The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and the service-level settings for a service to be completely enabled at runtime. Additionally, SLA Alerts depends on both the global and service-level Monitoring states—both must be enabled for SLA alerts to be enabled at runtime. See Section 27.2, "Enabling Global Settings."
>
> In addition, the service **State** must also be enabled.

- To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.1.3 Finding Business Services

To find business services:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **Business Services**, then click **Search**. The Summary of Business Services page displays the information shown in Table 27–3.

3. To restrict the number of items in the list or locate specific business services:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - Filter by business service name and path. In the **Name** and **Path** fields, enter the name and path of the search target, then click **Search**. The path is the project name and the name of the folder in which the business service resides. You can use * and ? as wildcard characters to perform a more general search.

   - Filter by service state. From the **State** list, select the business service state: Enabled or Disabled, then click **Search**. Select All to ignore the service state filter criteria.

   - Filter by service monitoring. From the **Monitoring** list, select the monitoring status of the business service: Enabled or Disabled, then click **Search**. Select All to ignore the service monitoring filter criteria.

   - Filter by business service's SLA alerts setting. From the **SLA Alerts** list, select the SLA alerts setting of the business service: Enabled or Disabled, then click **Search**. Select All to ignore the service SLA alerts filter criteria.

   - Filter by Message Tracing. Select whether you want to filter on business services that have message tracing Enabled or Disabled, then click **Search**. Select All to ignore the message tracing filter criteria.

   - Filter by Caching. Select whether you want to filter on business services that have result caching Enabled or Disabled, then click **Search**. Select All to ignore the result caching filter criteria.

Table 27–3   Business Services Summary Information

Property	Description
Name	The name assigned to the business service. The name is a link to the Operational Settings page. See Section 26.9, "Configuring Operational Settings for Business Services."
Path	The project associated with the business service. If the service resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Type	The parent service type: business service.
State	The state of the business service: Enabled or Disabled.
Monitoring	The monitoring status of the business service: Enabled or Disabled.
SLA Alerts	The SLA alerts status: Enabled or Disabled, and the level enabled at and above: Normal (N), Warning (W), Minor (Mn), Major (Mj), Critical (C), or Fatal (F).
Msg. Tracing	Whether message tracing is enabled or disabled for business services.
Caching	Whether result caching is enabled or disabled for business services.

You can use this page also to do the following:

- To remove the search filter and display all business services, click **View All**.

- To discard your changes and refresh the page with the currently stored settings using the same search criteria, click **Reset**.

- From within a session, to enable or disable any service-level settings, select or clear the check box next to it and click **Update**.

> **Note:** The runtime effects of the service-level settings depend on their corresponding global settings. You must enable both the global and the service-level settings for a service to be completely enabled at runtime. Additionally, SLA Alerts depends on both the global and service-level Monitoring states—both must be enabled for SLA alerts to be enabled at runtime. See Section 27.2, "Enabling Global Settings."
>
> In addition, the service **State** must also be enabled.

- To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.1.4 Finding Split-Joins

To find Split-Joins:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **Split-Joins**, then click **Search**. The Summary of Split-Joins page displays the information shown in Table 27–4.

3. To restrict the number of items in the list or locate specific business services:

- Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

- Filter by Split-Join name and path. In the **Name** and **Path** fields, enter the name and path of the search target, then click **Search**. The path is the project name and the name of the folder in which the Split-Join resides. You can use * and ? as wildcard characters to perform a more general search.

- Filter by Split-Joins with logging. From the **Logs** list, select the logging status of the Split-Join: Enabled or Disabled, then click **Search**. Select All to ignore the logging filter criteria.

- Filter by execution tracing. From the **Execution Tracing** list, select the execution tracing status of the Split-Join: Enabled or Disabled, then click **Search**. Select All to ignore the service logging filter criteria.

Table 27–4   Split-Join Summary Information

Property	Description
Name	The name assigned to the Split-Join. The name is a link to the Operational Settings page. See Section 10.3, "Editing Split-Joins."
Path	The project associated with the Split-Join. If the Split-Join resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Type	The parent service type: Split-Join.
Logs	The logging status: Enabled or Disabled, and the level enabled at and above: Debug (D), Info (I), Warning (W), or Error (E).
Execution Tracing	The runtime tracing status of the Split-Join: Enabled or Disabled.

You can use this page also to do the following:

- To remove the search filter and display all Split-Joins, click **View All**.

- To discard your changes and refresh the page with the currently stored settings using the same search criteria, click **Reset**.

- From within a session, to enable or disable any service-level settings, select or clear the check box next to it and click **Update**.

> **Note:**   The runtime effects of the Split-Join-level settings depend on their corresponding global settings. You must enable both the global and the service-level settings for a service to be completely enabled at runtime.

- To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.1.5 Finding Alert Destinations

To find alert destinations:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **Alert Destinations**, then click **Search**. The Summary of Alert Destinations page displays the information shown in Table 27–5

3. To restrict the number of items in the list or locate specific alert destinations:

- Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

- Filter by alert destination name and path. In the **Name** and **Path** fields, enter the name and path of the search target, then click **Search**. The path is the project name and the name of the folder in which the alert destination resides. You can use * and ? as wildcard characters to perform a more general search.

- Filter by alert destination target. In the **Target** multi-select box, select one or more alert destination targets: SNMP Trap, Reporting, Alert Logging, email, or JMS, then click **Search**. Only alert destinations with at least one of the selected targets will be displayed. Select no alert destination targets to ignore the alert destination filter criteria.

- Filter by search pattern. In the **Search Pattern (Any String)** text box, specify any string, then click **Search**. The system uses the string to search all the **Description** fields of the alert destinations, as well as the specific detailed fields of the email and JMS destinations. If the string appears in any of the alert destination fields, the alert destinations matching the search criteria are displayed.

Table 27–5   Alert Destinations Summary Information

Property	Description
Name	The name of the alert destination resource. This field is a link to the View Alert Destination - Configuration page. See Section 6.1, "Locating Alert Destinations."
Path	The project associated with the alert destination. If the alert destination resides in a project folder, this folder is also listed. The path is displayed in the format: `project-name/root-folder/ . . ./parent-folder` The path is a link to the corresponding path in the Project Explorer.
Options	Click the **Delete** icon to delete a specific alert destination. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource. See Section 6.5, "Deleting Alert Destinations."

You can use this page also to do the following:

- To remove the search filter and display all alert destinations, click **View All**.

- To delete a specific alert destination if not referenced by other resources, select the check box next to it and click **Delete**.

- To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.1.6 Finding SLA Alert Rules

To find SLA alert rules:

1. Select **Operations** > **Smart Search**.

2. From the **Type** list, select **SLA Alerts**, then click **Search**. The Summary of SLA Alert Rules page displays the information shown in Table 27–6.

3. To restrict the number of items in the list or locate specific SLA alert rules:

- Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

- Filter by alert rule name. In the **Name** field, enter the name of the search target or enter wildcard characters (use * and ? as wildcard characters to perform a more general search), then click **Search**.

  This search method is preferable if the name of the alert rule is unique across all projects and paths.

- Filter by parent service path. In the **Path** field, enter the path of the parent service, which is the project name and the name of the folder in which parent service resides. You can use * and ? as wildcard characters to perform a more general search. All alert rules whose parent services reside in that path are displayed.

  The format for the **Path** field is as follows:

  ```
  project-name/root-folder/ . . ./parent-folder
  ```

  If a service is directly under the project, the format is as follows:

  ```
  project-name
  ```

- Filter by parent service. In the **Parent Service** field, click **Browse** to display the Select Service page. Select the parent service from the list and click **Submit**. Click **Clear** to remove the parent service filter criteria, or click **Search**.

  Note: Your selection in the **Parent Service** field restricts your choices in the **Service Type** field.

- Filter by parent service type. From the **Service Type** list, select Proxy Services or Business Services, then click **Search**. Select All Services to ignore filtering by parent service type criteria.

  Note: Your selection in the **Service Type** field restricts your choices in the **Parent Service** field.

- Filter by rule state. From the **Rule State** list, select the state of the alert rule: Enabled or Disabled, then click **Search**. Select All to ignore the rule state filter criteria.

- Filter by severity. From the **Severity** list, select to restrict SLA alerts rules to the specified level, then click **Search**.

  Select the **or above** check box to restrict your search to the specified severity level or above (listed from the most inclusive to the most restrictive level): Normal, Warning, Minor, Critical, and Fatal.

Table 27–6   SLA Alert Rules Summary Information

Property	Description
Name	The name assigned to this alert rule. The name is a link to the Alert Rule Configuration page. See Section 26.24, "Viewing Alert Rule Configurations."
SLA State	The status of the alert rule: Enabled or Disabled.
Description	**Note**: This field is hidden by default.
	A description of the alert rule.
Service Name	The name of the parent service. The name is a link to the SLA Alert Rules page. See Section 26.21, "Viewing Alert Rules."
Path	The project associated with the parent service of the alert rule. If the parent service of the alert rule resides in a project folder, this folder is also listed. The path is displayed in the format:
	`project-name/root-folder/ . . ./parent-folder`
	The path is a link to the corresponding path in the Project Explorer.
Severity	The severity of the alert that is triggered by this rule: Normal, Warning, Minor, Major, Critical, or Fatal.
Aggr. Interval	The length of the aggregation interval in terms of hours and minutes.
Expiration Date	The date when this alert rule is no longer in effect.
Stop Processing	**Note**: This field is hidden by default.
	Displays Yes or No.
Frequency	The frequency of this alert:
	■ Every Time
	■ Notify Once

You can use this page also to do the following:

■ To remove the search filter and display all SLA Alert Rules, click **View All**.

■ To discard your changes and refresh the page with the currently stored settings using the same search criteria, click **Reset**.

■ From within a session, to enable or disable an SLA Alert Rule, select or clear the check box next to it and click **Update**.

■ To customize your view of the operational settings information, click the **Table Customizer** icon. See Section 2.2, "Customizing Table Views."

27.2 Enabling Global Settings

You can enable or disable all global settings only from within a session.

This section includes the following topics:

■ Section 27.2.1, "Enabling Global Monitoring of Services"

■ Section 27.2.2, "Disabling Global Monitoring of Services"

■ Section 27.2.3, "Enabling SLA Alerts Globally"

■ Section 27.2.4, "Disabling SLA Alerts Globally"

- Section 27.2.5, "Enabling Pipeline Alerts Globally"

- Section 27.2.6, "Disabling Pipeline Alerts Globally"

- Section 27.2.7, "Enabling Message Reporting Globally"

- Section 27.2.8, "Disabling Message Reporting Globally"

- Section 27.2.9, "Enabling Logging Globally"

- Section 27.2.10, "Disabling Logging Globally"

- Section 27.2.11, "Enabling Result Caching Globally"

- Section 27.2.12, "Disabling Result Caching Globally"

27.2.1 Enabling Global Monitoring of Services

To turn monitoring for all services on at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable Monitoring** check box for the system to start collecting monitoring statistics for all services whose monitoring is enabled at the service level.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.2 Disabling Global Monitoring of Services

To turn monitoring for all services off at the domain level:

> **Note:** If you disable monitoring for all services, all statistics collected so far for those services are deleted as well. These statistics cannot be restored using the session **Undo** function. You can use **Undo** to enable monitoring again for the services, but the deletion of the statistics is irreversible.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable Monitoring** check box for the system to stop collecting monitoring statistics for all services in your configuration.

> **Note:** This option overrides the **Enable Service Monitoring** option that you can select for specific business and proxy services. See Section 26.9, "Configuring Operational Settings for Business Services" and Section 26.8, "Configuring Operational Settings for Proxy Services."
>
> This option also overrides the **Enable SLA Alerts** option. If you disable monitoring at the global level, SLA alerts will also be disabled at runtime as well, even though the corresponding **Enable SLA Alerts** check box may be selected. See Section 27.2.3, "Enabling SLA Alerts Globally."

4. Click **Update**. The monitoring configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.3 Enabling SLA Alerts Globally

Although you can configure SLA Alerts independently from Monitoring, there is an interaction between the two settings at runtime. If global monitoring is enabled, SLA alerts can be enabled or disabled. However, if global monitoring is disabled then SLA alerts will be effectively disabled because SLA alert rule conditions depend on monitoring statistics being evaluated.

To turn SLA alerts for all services on at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable SLA Alerts** check box for the system to start evaluating alert rules for all services in your configuration.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.4 Disabling SLA Alerts Globally

To turn SLA alerts for all services off at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable SLA Alerts** check box for the system to stop evaluating alert rules for all services in your configuration.

> **Note:** This option overrides any SLA Alerts settings that you can select for specific business and proxy services. See Section 26.9, "Configuring Operational Settings for Business Services" and Section 26.8, "Configuring Operational Settings for Proxy Services."

4. Click **Update**. The SLA alerts configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.5 Enabling Pipeline Alerts Globally

For proxy services, to turn pipeline alerts for all proxy services on at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable Pipeline Alerts** check box for the system to start executing any pipeline alert actions for proxy services.

   Pipeline Alerts has no dependency on global **Monitoring**.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.6 Disabling Pipeline Alerts Globally

For proxy services, to turn pipeline alerts for all proxy services off at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable Pipeline Alerts** check box for the system to stop executing any pipeline alert actions for proxy services.

> **Note:** This option overrides any Pipeline Alerts settings that you can select for specific proxy services. See Section 26.8, "Configuring Operational Settings for Proxy Services."

4. Click **Update**. The pipeline alerts configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.7 Enabling Message Reporting Globally

This option controls proxy service pipeline Report actions on the message context only. It does not effect SLA alerts or pipeline alerts targeted to the reporting framework.

To turn message reporting for all proxy services on at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable Reporting** check box to start any pipeline Report actions for all proxy services.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.8 Disabling Message Reporting Globally

For proxy services, to turn message reporting for all proxy services off at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable Reporting** check box to stop any pipeline Report actions for all proxy services.

> **Note:** This option overrides any Message Reporting settings that you can select for specific proxy services. See Section 26.8, "Configuring Operational Settings for Proxy Services."

4. Click **Update**. The message reporting configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.9 Enabling Logging Globally

If you select **Enable Logging**, pipeline Log action messages are sent to the WebLogic Server logging service. To be able to view them, you must configure WebLogic Server to forward these messages to the domain log.

To turn logging output for all proxy services on at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable Logging** check box to start any pipeline Log actions for all proxy services.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.10 Disabling Logging Globally

For proxy services, to turn logging output for all proxy services off at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable Logging** check box to stop any pipeline Log actions for all proxy services.

> **Note:** This option overrides any Logging settings that you can select for specific proxy services. See Section 26.8, "Configuring Operational Settings for Proxy Services."

4. Click **Update**. The logging configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.2.11 Enabling Result Caching Globally

If you invoke business services whose results seldom change, result caching improves business service performance by returning cached results to the client instead of waiting for the results of a service invocation.

Before configuring result caching for individual business services, you must first enable results caching globally using the following steps. The default setting for result caching is enabled.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Select the **Enable Result Caching** check box.

4. Click **Update** to save all of the settings on this page.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

For information on configuring result caching for business services, see Section 19.2.23, "Message Handling Configuration Page."

27.2.12 Disabling Result Caching Globally

When you disable result caching globally, Oracle Service Bus flushes the entire cache, removing cached results for all business services with result caching enabled. The default setting for result caching is enabled.

To disable result caching at the domain level:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **Operations** > **Global Settings**.

3. Clear the **Enable Result Caching** check box.

> **Note:** This setting overrides any result caching configured on individual business services.

4. Click **Update**. The configuration is saved in the current session.

5. Alternatively, click **Reset** at any time during the session to discard your changes and refresh the page with the currently stored settings.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

27.3 Setting User Preferences

Use this page to customize and persist your Administration Console display options and settings. You can update (change) your User Preferences without activating a change session.

Table 27–7 describes the options and controls on the page.

Table 27–7   User Preferences - Options and Controls

To...	Do This...
Change your Home Page, the page that opens when you launch the Administration Console	From the first list, select any main menu choice: **Operations**, **Resource Browser**, **Project Explorer**, **Security Configuration**, or **System Administration**.
	From the second list, select any one of the sub-menu choices. (The contents of the sub-menus dynamically change based on your first menu selection.)
Display Search Filters	Select **Yes** or **No**.
	By default, search filters are not displayed (closed). By selecting **Yes**, search filters are always displayed (open).
Display Stage Annotations	Select **Yes** or **No**.
	By default, pipeline stage annotations are not displayed (closed). By selecting **Yes**, stage annotations are always displayed (open).
Display Resource Metadata	Select **Yes** or **No**.
	By default, resource metatdata sections are displayed (open). By selecting **No**, resource metatdata sections are not displayed (closed).

Table 27–7   (Cont.) User Preferences - Options and Controls

To...	Do This...
Change how often the Administration Console updates the display of data on the Dashboard	From the **Dashboard Refresh Rate** list, select **No Refresh**, **1**, **2**, **3**, **5**, **10**, **20**, **30** or **60 minutes** as the refresh rate for the Dashboard.
	The default rate is **No Refresh**, but you can select another predefined time. For example, if you select **5 minutes**, the Dashboard is updated with data every 5 minutes.
	Caution: By selecting a dashboard refresh rate interval other than the default (**No Refresh**), the browser refreshes the Dashboard at regular intervals. This prevents your session from timing out, even without you interacting with the Administration Console. Oracle recommends that you use the Dashboard refresh rate feature with caution. You should never leave an Administration Console session unattended. While your Administration Console session is active others can gain access to it from your computer, without re-authentication.
Change the time interval for displaying historical data for alerts on the Dashboard	From the **Alert History Duration** list, select **30 minutes**, **1**, **2**, **3**, or **6 hours** as the time interval for displaying historical alert data. The default is 30 minutes (displays alerts received within the last 30 minutes) but you can select another predefined duration. For example, if you select **1 hour**, the Dashboard displays the alerts received within the last hour.

28

Reporting

This chapter describes how to view and purge messages generated by the reporting feature in the Oracle Service Bus Administration Console.

Before you can use the Reporting module, you must first create a proxy service and then add a report action to the message flow of that proxy service. See Section 21.1, "Adding and Editing Actions in Message Flows" and Section 21.27, "Adding Report Actions." See also "Reporting" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

28.1 Viewing Reporting Messages

Use the Message Report Summary page to view reporting messages information shown in Table 28–2. The page displays up to 100 messages.

1. To restrict the number of items in the list or locate specific messages:

    ■ Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

    ■ To set the filter parameters, do one or more of the following steps shown in Table 28–1.

Table 28–1   Message Filter Parameters

To...	Complete These Steps...
Filter by time parameters	Set a time period: 1. Select the **Messages From**... **To**... option. 2. In the **Messages From** field, enter a date and time in the format `MM/DD/YY HH:MM:SS AM|PM`. For example, you can enter 10/10/07 12:45:00 AM. 3. In the **Messages To** field, enter a date and time in the format `MM/DD/YY HH:MM:SS AM|PM`. Alternatively, you can select parameters for the most recent messages: 1. Select the **For the Last** option. 2. In the **For the Last** field, enter the number of days, then specify hours and minutes. **Note**: By default, message reports for the last 30 minutes are displayed.
Filter by service name	In the **Inbound Service Name** field, enter the service name.
Filter by error code	In the **Error Code** field, enter an error code.

Table 28–1   (Cont.) Message Filter Parameters

To...	Complete These Steps...
Filter by report index	In the **Report Index** field, enter the report indexes. These are the key value pairs you defined when you configured the report action for this proxy service. This is a string search.

- You can use wildcard characters in service name, error code, and report index field searches.

  - Entering only asterisk (*) returns all messages. For example, if you enter * in the error code field, messages with and without error codes are displayed.

  - In the Error Code field only, strict search limitations apply. For example, to find messages with the error code BEA12345, you can enter *, BE*, or BEA12345.

2. Click **Search**. The page displays the messages that fall within the filter parameters you set. The maximum number of messages that can be displayed when filtering messages is 1000.

3. Click **View All** to remove the search filters and display all messages.

4. To locate a message, you can also do the following:

   - Resort the list of messages. Click an underlined column name. Ascending and descending arrows indicate the sort order. Click the column name to change the sort order.

   - Scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

Table 28–2   Message (Reporting) Summary Details

Property	Description
Report Index	A link to the View Message Details page. See Section 28.2, "Viewing Message Details."
	Report indexes are key value pairs that you use to extract key identifiers from a message context variable. Keys are a convenient way to filter messages. You use the report action to add key value pairs. See Section 21.1, "Adding and Editing Actions in Message Flows."
DB TimeStamp	The date and time that the message was added in the database.
Inbound Service	The inbound service associated with the message. The service is a link to the View a Proxy Service - Configuration Details page. See Section 20.5, "Editing Proxy Service Configurations."
Error Code	The error code associated with this message, if there is one.
	You specify error codes when you configure the raise error action. See "Handling Errors in Message Flows" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus* and Section 24.2, "Adding Proxy Service Error Handlers."

Related Topics

See also "How to Enable Message Reporting" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

28.2 Viewing Message Details

Use this page to view details of a specific message.

1. Locate the message. See Section 28.1, "Viewing Reporting Messages."

2. Click the report index. The View Message Details page displays the General Configuration information shown in Table 28–3.

Table 28–3   Message Configuration Parameters

Property	Description
Message ID	The unique identification for this message.
Database Timestamp	The date and time that the message was registered in the database.
Time at point of Logging	The date and time that the message was logged.
Server name	The name of the server in which this message was generated.
State	State of the pipeline in which this message was generated: REQUEST—indicates that the reporting action is executed in a request pipeline. RESPONSE—indicates that the reporting action is executed in a response pipeline. ERROR—the action is running in the service-level error handler.
Node Name	The node name of the pipeline in which this message was generated.
Pipeline Name	The name of the pipeline in which this message was generated.
Stage Name	The name of the stage in which this message was generated.

The page displays the Inbound Service information shown in Table 28–4.

Table 28–4   Inbound Service Details

Property	Description
Name	The inbound proxy service associated with this message. This is a link to the View a Proxy Service - Configuration Details page. See Section 20.5, "Editing Proxy Service Configurations."
URI	The URI associated with the proxy service.
Operation	The inbound operation associated with this message.

The page displays the Outbound Service information shown in Table 28–5.

Table 28–5   Outbound Service Details

Property	Description
Name	The outbound business service associated with this message. This is a link to the View a Business Service - Configuration Details page. See Section 19.5, "Editing Business Service Configurations."
URI	The URI to the outbound business service end point.
Operation	Name of the operation invoked on the outbound service.

The page displays the Report Index information shown in Table 28–6.

Table 28–6   Report Index Details

Property	Description
Report Index Text	Report indexes are key value pairs that you use to extract key identifiers from a message context variable. Keys are a convenient way to filter messages. You use the report action to add key value pairs. See Section 21.1, "Adding and Editing Actions in Message Flows."

The page displays the Fault information shown in Table 28–7.

Table 28–7   Fault information Details

Property	Description
Error Code	The error code associated with the message, if there is one.
	You specify error codes when you configure the raise error action. See "Handling Errors in Message Flows" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus* and Section 24.2, "Adding Proxy Service Error Handlers."
Reason	The reason for the error code.
Detail	The fault details associated with the error code. These details, if present, are typically a stack trace of where a particular fault occurred, and may be truncated due to a size limitation in the database. The limit is 2048 characters.

The page displays the Report Body information shown in Table 28–8.

Table 28–8   Report Body Details

Property	Description
Report Body Text	This link opens another browser that displays the details of the report body. You use an expression in the report action to capture the report body text. See the procedure for the report action in Section 21.1, "Adding and Editing Actions in Message Flows."

3. Click **OK** when you have finished viewing the details of a specific message.

28.3 Purging Messages

Use this page to purge selected messages from the reporting datastore.

Message purging occurs in the background so the Message Report Summary page is not locked up while the purge occurs. Therefore, you can do a purge that takes a while to finish; the length of time depends on how many messages need to be purged. However, if you filter reporting messages, the purging process is slowed down; the Message Report Summary page may display incorrect data as some of the data hasn't been purged yet.

The Purge Messages feature may be useful during design and test phases of your project. However, in a production environment, Oracle recommends that the management of the data in your database (including purging) is handled by the Database Administrator using database management tools.

1. Select **Operations** > **Purge Messages**.

2. Do one of the following steps, shown in Table 28–9.

Table 28–9   Message Purging Options

To...	Complete These Steps...
Purge all messages	1. Select the **Purge All Messages** option. 2. Click **Purge**.
Purge messages within a specified time frame	1. Select the **Purge Messages From**... **To**... option. 2. In the **Purge Messages From** field, enter a month, day, year, and time in the format `MM/DD/YY HH:MM:SS`. For example, you can enter 10/10/07 12:45:00 AM. 3. In the **Purge Messages To** field, enter a month, day, year, and time in the format `MM/DD/YY HH:MM:SS`. 4. Click **Purge**.

Related Topics

To learn more about purging messages, see "Reporting" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

29

Import/Export

This chapter describes how to import and export Oracle Service Bus (OSB) projects and resources using the OSB Administration Console.

This chapter includes the following sections:

Section 29.1, "Importing Resources"

Section 29.2, "Exporting Resources"

29.1 Importing Resources

Use the Import Resources page to import a full project JAR file or a resource JAR file that has been previously exported from another Oracle Service Bus domain. You can choose to import only a subset of the exported data. If the resource already exists in the importing system, it will be updated. Otherwise, it will be created. Resources are only scheduled for deletion when the JAR being imported is a full project JAR and there are resources located in the same project in the importing system that are not present in the imported JAR file. It will not delete resources which are located in other projects.

You can import resources at the resource level even though they were exported at the project level. For example, even if system resources were exported in a full project JAR file, you can deselect them when importing.

You cannot export users, groups, roles, or certificates when you export a configuration. Therefore, you must create these objects again when you import an exported configuration.

> **Note:** For information on importing resources that are not part of an Oracle Service Bus configuration, see Section 4.15, "Loading Resources from a Zip File" and Section 4.17, "Loading Resources from a URL."

---

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Import Resources**.

3. Select an exported project or resource JAR file.

   a. In the **File Name** field, click **Browse** to locate the directory where the JAR file is stored.

   b. Select the JAR file, then click **Open**.

4. Click **Next**. Continue in Section 29.1.1, "Selecting Projects and Resources to Import."

29.1.1 Selecting Projects and Resources to Import

To select projects and resources to import:

1. If you are importing a resource JAR file and want to ensure that all its associated resources are imported, select **Include Dependencies**.

   > **Note:** **Include Dependencies** cannot be selected if you are importing a full project JAR file or a pre-version 2.6 configuration file.

2. Click the **Open** icon to display Advanced Settings for preserving values in the importing system.

3. When you are re-importing a resource but desire to protect any customization made to the existing resource, select **Preserve Environment Variables**.

   > **Note:** Customizing any references within an existing resource cannot be preserved. See Section 32.2, "Creating Customization Files."

4. When you are importing the Global Operational Settings resource or other resources like business or proxy services that contain associated service-level operational settings, and desire to preserve the operational values in the importing system, select **Preserve Operational Values**.

5. To prevent overwriting of existing security-related configuration in resources which are being updated, select among the following options:

   a. **Preserve Security and Policy Configuration**—preserves the security configuration (excluding access control policies) and the references to the WS-Policies bound directly to the service (instead of bound to the WSDL).

      For more information, see "Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

   b. **Preserve Credentials**—preserves PKI credentials in service key providers, user name and passwords in service accounts, and user name and password credentials in SMTP Servers, Proxy Servers, JNDI Providers, and UDDI Registries.

   c. **Preserve Access Control Policies**—preserves access control policies.

   The resource summary information is shown in Table 29–1.

Table 29–1   Resource Information

Property	Description
Name	The name of the project.
	Global resources, such as UDDI Registry information, SMTP Servers, Proxy Servers, JNDI Providers, and Global Operational Settings appear under the System project in their respective folders.
Type	The resource type, which is Project.
	Expand the Project folder to display and deselect resources.

Table 29–1   (Cont.) Resource Information

Property	Description
Operations	Displays --- or !Deletions!
	Expand the Project folder to display the operations (create, update, delete) that will be performed on the resources.
	To prevent resources from being deleted, deselect them. Resources are only scheduled for deletion when the JAR being imported is a full project JAR and there are project resources in the importing system that are not present in the imported JAR file.
References	Expand the Project folder to display the number of objects that each resource references.

6. Select the objects on which you want to operate (create, update, or delete).

To display each object contained in the JAR file, expand the folder for the file. The check box associated with each object is selected.

The following resource file details, displayed for each object, are shown in Table 29–2.

Table 29–2   Resource File Details

Property	Description
Name	The name and path of the resource. The path is the project name and the name of the folder in which the resource resides.
Type	The Oracle Service Bus resource types.
	Note: If a resource has sensitive security data, an Encryption icon is displayed.
	Caution: If you select a service account or service key provider, you might overwrite existing security data *unless* you specify to preserve security settings during import. See Section 29.1.3, "Importing Service Accounts or Service Key Providers."
Operations	Create, update, or delete.
	Resources are only scheduled for deletion when the JAR being imported is a full project JAR and there are project resources in the importing system that are not present in the imported JAR file.
References	The number of objects that this resource references.

7. Clear the check boxes next to the resources that you do not want to import.

8. Click **Import**.

After you finish

If applicable, continue in Section 29.1.2, "Specifying the Importing Security Settings". Or, continue in Section 29.1.4, "Reviewing the Import Summary."

29.1.2 Specifying the Importing Security Settings

If the JAR file contains resources with user name and password data that were encrypted when it was exported, the Import Resources - Security Settings page is displayed.

1. Enter the password that was used to encrypt the data (see Section 29.2, "Exporting Resources").

If you do not know the password that was used to encrypt the resource data, click the **Prev** button and clear the check box that is next to the resource. You can import all other non-encrypted resources without knowing the password.

2. Click **Finish Import**.

The Processing Configuration Data pop-up window is displayed. You should wait until this window closes before you do anything else on the Administration Console. The window closes automatically when the system finishes importing the objects. The Import Resources page is displayed, which includes a message that you have successfully deployed the data.

After you finish

Continue in Section 29.1.4, "Reviewing the Import Summary."

29.1.3 Importing Service Accounts or Service Key Providers

> **Note:** In earlier Oracle Service Bus versions, service key providers were called proxy service providers.

If the JAR file was created by AquaLogic Service Bus 3.0 or later and contains service accounts or service key providers, you can import these resources along with the user names, passwords, local-user to remote-user mappings, and alias to key-pair bindings that they contain.

For each service key provider, Oracle Service Bus imports the alias to key-pair binding into the PKI credential mapping provider. If this data was encrypted during export, you must supply the password that was used to encrypt the data. If you do not know the password, you can import all other non-encrypted resources.

If you import a service account or service key provider and a corresponding resource of the same name already exists in your domain, the imported resource will overwrite the one already in your domain, even if the one already in your domain has been modified during the current session, *unless* you specify to preserve security settings during import, performed in Section 29.1.1, "Selecting Projects and Resources to Import."

29.1.4 Reviewing the Import Summary

The import summary information is shown in Table 29–3.

Table 29–3   Import Summary Information

Property	Description
Status	An indicator of whether the resource was imported successfully.
	A WSDL may be temporarily marked as invalid until all of its dependencies are imported. When the import action is completed, however, the configuration service re-validates the files marked invalid.
Name	The name of the resource.
Path	The path of the resource. The path is the project name and the name of the folder in which the resource resides.
Type	Displays the Oracle Service Bus resource types.

Table 29–3 (Cont.) Import Summary Information

Property	Description
Diagnostic Message	Displays error or informational messages if they exist for this resource.

1. Click **Import Another**.

   You can import another JAR file, or you can click **View Changes** under Change Center to view the import details. See Section 3.4, "Viewing Configuration Changes."

2. When you have finished, click **Activate** under Change Center.

29.2 Exporting Resources

Use the Export Resources page to **Export Projects** or **Export Resources**. If you choose to export a full project, one or more projects can be selected for export. You can still expand the projects to see folders and resources but selection is only at the project level. Also, **Include Dependencies** cannot be selected.

You can export projects and resources whether you are working within a session or outside of a session. If you export within a session, the resources are session resources and the configuration may be incomplete or have conflicts. If you export outside of a session, the resources are activated core resources.

Oracle Service Bus cannot export the users, groups, or roles that you create in the Security Configuration module of the Oracle Service Bus Administration Console. Neither can it export credential maps or other security-provider data that you create in the WebLogic Server Administration Console. Instead, use the WebLogic Server Administration Console to export this data. See "Migrating Security Data" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

1. Select **System Administration** > **Export Resources**.

   The Export Resources page displays the list of objects in your configuration. The name, type, last modified time, and references for each object are displayed.

   For services, the last modified timestamp reflects the modifications made to the service definition, as well as to the service transport or pipeline configuration. It does not, however, reflect the changes made to the alerts defined for that service.

2. Select whether to **Export Projects** or **Export Resources**.

   > **Note:** Exporting projects might cause resource deletion when you import the full project JAR file. For more information, see Section 29.1, "Importing Resources."

3. Select the projects or resources you want to export. If you selected **Export Projects**, you cannot clear the check boxes for individual resources.

   a. Expand the project folders. The name, type, last modified time, and references for each resource contained in the project are displayed.

   b. Select the check boxes associated with the projects or resources you want to export.

   c. Clear the check boxes associated with the projects or resources that you do not want to export.

4. If you selected **Export Resources**, select or clear the **Include Dependencies** check box. Use this option to export any other resources that this resource references.

5. Click **Export** to create a JAR file and export it.

After you finish

If applicable, continue in

29.2.1 Specifying the Exporting Security Settings

If your export includes a service account or a service key provider, the Export Resources - Security Settings page is displayed.

For each service account, Oracle Service Bus exports the user name and password or the local-user to remote-user map (depending on which data was stored in the service account). For each service key provider, Oracle Service Bus exports the alias to key-pair binding from the PKI credential mapping provider; it does not export private keys, certificates, or other data from the key stores. Key store data must be exported using tools that the key store vendor provides.

If you want Oracle Service Bus to encrypt the user name and password data that is in the service account, service key provider, UDDI Registry, JNDI Provider, or SMTP provider resources before exporting it the to JAR file, do the following:

1. Select the **Protect Sensitive Data** check box.

2. Enter and confirm a password.

    When you or someone else attempts to import this JAR, Oracle Service Bus will not import the resources with encrypted user name and password data unless you specify this password. You can import all of the other non-encrypted resources in the JAR without specifying the password.

3. Click **Finish Export** button.

4. When you click Export or Finish Export, the Processing Configuration Data pop-up window is displayed. Wait until this window closes before you do anything else on the Administration Console. The window is closed automatically when the File Download dialog box is displayed.

5. In the File Download dialog box, click **Open** to open the exported JAR file or click **Save** to save the JAR file to your desktop.

30

UDDI

This chapter describes how to configure and use UDDI registries with Oracle Service Bus (OSB) for UDDI import and publish using the OSB Administration Console.

Universal Description, Discovery and Integration (UDDI) registries are used in an enterprise to share Web Services. UDDI provides a framework in which to classify your business, its services, and the technical details about the services you want to expose.

Publishing a service to a registry requires knowledge of the service type and the data structure representing that service in the registry. A registry entry has certain properties associated with it and these property types are defined when the registry is created. You can publish your service to a registry and make it available for other organizations to discover and use. Proxy services developed in Oracle Service Bus can be published to a UDDI registry. Oracle Service Bus can interact with any UDDI version 3.0-compliant registry.

See also "UDDI" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

30.1 Viewing Configured UDDI Registries

Use this page to designate a default UDDI registry (see Section 30.3, "Setting Up a Default UDDI Configuration") and to view UDDI registries. Additionally, you can search for registries that you have previously configured.

1. In the UDDI Registries panel, enter the name of the registry you want to find in the **Name** field, or enter wildcard characters; use * for multiple characters and ? for single characters. The search returns all the records that satisfy the name criteria.

2. Click **Search**.

3. Click **View All** to remove the search filters and view all the registries that are configured to work with Oracle Service Bus.

   Table 30–1 displays UDDI registry configuration settings.

Table 30–1   UDDI Registry Configuration Settings

Property	Description
Registry Name	The name of the registry. The name assigned to a registry when it is first created. You cannot edit the name of a registry after the entry is saved.
Inquiry URL	The URL of the Inquiry API endpoint used for locating and importing services.

Table 30–1   (Cont.) UDDI Registry Configuration Settings

Property	Description
Publish URL	The URL of the Publish API endpoint used for publishing services.
Security URL	The URL of the Security API endpoint used for getting an authentication token so that you can publish to the registry.
Subscription URL	The URL of the Subscription API endpoint used for subscribing to registry changes, creating a registry subscription, and listening for changes to imported services.
User Name	The user name which is required for user authentication.
Password (Confirm Password)	The password which is required for user authentication.
Load tModels into registry	Loads the tModels into the selected registry. This option only has to be selected once per registry.
Enable Auto-Import	Auto-synchronizes services with the UDDI registry.

30.2 Configuring UDDI Registries

This section includes the following tasks:

- Section 30.2.1, "Adding UDDI Registries"
- Section 30.2.2, "Editing UDDI Registries"

30.2.1 Adding UDDI Registries

Use this page to add and configure UDDI registries. After registries are configured, you can then publish Oracle Service Bus proxy services to them or import business services from them to be used in an Oracle Service Bus proxy service.

You must be in an active session to add a registry.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **UDDI Registries**.

3. In the UDDI Registries panel, click **Add Registry**.

4. To configure the registry, complete the following steps. Asterisks denote required fields. See Table 30–1 for descriptions of the properties that must be set when adding a registry.

   a. In the **Name** field, enter a name for the registry.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

   b. In the **Inquiry URL** field, enter an inquiry URL in the format:
   `http://host:port/APPLICATION_SERVER_CONTEXT/uddi/inquiry`

   c. In the **Publish URL** field, enter a publish URL in the format
   `http://host:port/APPLICATION_SERVER_CONTEXT/uddi/publishing`

   d. In the **Security URL** field, enter a security URL in the format
   `http://host:port/APPLICATION_SERVER_CONTEXT/uddi/security`

   e. In the **Subscription URL** field, enter a subscription URL in the format

   `http://host:port/APPLICATION_SERVER_CONTEXT/uddi/subscription`

By default, the value for *APPLICATION_SERVER_CONTEXT* in the installer is `registry`.

f. In the **Username** field, enter the user name to log in to the registry console.

g. In the **Password** /(**Confirm Password**) field enter the password to log in to the registry console.

h. Select the **Load tModel into registry** check box to publish the Oracle Service Bus tModels to this registry.

This field is only required when publishing proxy services to this registry.

i. Select the **Enable Auto Import** check box to auto-synchronize imported services with the UDDI registry. Any service that is imported with this option selected will be kept in synchrony with the UDDI registry.

Note: Auto-synchronization is a background process; you cannot reverse it using the session **Undo** function. Undoing an auto-synchronization change is not permanent as the service will be re-synchronized in the next synchronization cycle. If you want an imported service to stay out of synchrony with the UDDI registry, you have to detach the service to avoid further updates from the registry. See Section 30.6, "Detaching Services."

---

5. Click **Validate** to validate the URLs you have provided.

6. Click **Save** to configure the registry with the settings provided and save it in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

30.2.2 Editing UDDI Registries

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **UDDI Registries**.

3. In the UDDI Registries panel, from the list of available registries, click the name of the registry you want to edit.

You can also search for a specific registry using the **Search** option. See Section 30.1, "Viewing Configured UDDI Registries."

The UDDI Registry Configuration page displays the configuration properties that define the registry and the following registry-specific information, shown in Table 30–2.

Table 30–2   UDDI Registry Information

Property	Description
Last Modified By	The user who made the last configuration change to this UDDI registry resource.
Last Modified On	The date and time that the last configuration change was made. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."

Table 30–2   (Cont.) UDDI Registry Information

Property	Description
References	The number of objects that this UDDI registry references. If such references exist, click the numeric link to view a list of the references. See Section 4.22, "Viewing References to Resources."
Referenced By	The number of objects that reference this UDDI registry. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of the registry.

4. At the bottom of the page, click **Edit**.

5. On the UDDI Configuration-Edit Registry page, edit the registry configuration parameters.

   These properties are described in Table 30–1. You cannot edit the name of the registry.

6. Click **Save** to commit the updates in the current session.

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

30.3 Setting Up a Default UDDI Configuration

You can designate one of the configured registries as the default UDDI registry for the domain. See Section 30.2, "Configuring UDDI Registries."

To use the auto-publish functionality, you must first set a default registry. See Section 30.8, "Using Auto-Publish."

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **UDDI Registries**.

3. In the UDDI Default Configuration panel, click **Select Default**.

4. From the **Default Registry Name** list, select the name of the registry you want to set as the default registry.

5. To set a default business entity, choose an entity from the **Business Entity** list.

6. Click **Make Default**.

7. To change the default registry, click **Clear Selection** and repeat the above steps.

30.4 Importing Business Services From a UDDI Registry

You can import the following business service types from a UDDI registry into Oracle Service Bus:

- WSDL services over HTTP transport.

- Oracle Service Bus proxy services that are published to a UDDI registry. This feature is primarily used in multi-domain Oracle Service Bus deployments where proxy services from one domain need to discover and route to proxy services in another domain.

If you only have one registry configured, Oracle Service Bus automatically connects to that registry and retrieves all the business entities and populates the search form. If

you have more than one registry configured, the import wizard prompts you to first select the registry from which you want to import services.

On import, the inquiry URL is used to locate a specific service as a registry has several different types of services.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Import From UDDI**.

3. From the **Import Registry Name** list, select the name of the registry from which you want to import the service, then click **Next**.

   When a single registry is configured in Oracle Service Bus, the initial registry selection page is not displayed as this is the default registry. The registry selection page is only displayed when two or more registries are defined.

   Use the Import From UDDI page to search for specific business services and import them. You can search for a service by business entity, by pattern, or by using a combination of both.

4. Select the business entity name (for example, document Services) from the **Entity Name** list to search by business entity.

5. In the **Service Name** field, enter the pattern (for example, a%) for which you want to search. A list of business services matching the entered search criteria is displayed.

6. Select the service(s) that you want to import, then click **Next**.

   If you are unable to find a desired service, it may be because you do not belong to the security group with permissions to view its records.

7. Select the binding templates from which you want to create business services, then click **Next**.

   If a selected service has multiple binding templates, then each of these binding templates results in a business service. In this case you will be prompted to further narrow your selection among the binding templates you want to import.

8. On the Import UDDI: Select Import Location page select a project from the **Project** list, select a folder where you want to import your services and associated resources, then click **Next**.

9. The Import UDDI: Review and Import Services page displays a list of all of the resources (business services, MFLs, schemas, WSDLs, and so on) that will be created in the system. A warning message is displayed for any resource that cannot be imported.

   By default, all the items in the list are marked for import.

   a. Clear the check box next to a specific resource if you already have the resource in your system and want to manually resolve the dependencies.

   b. You can resolve dependencies from the View Conflicts page in the Change Center after the import process is complete. For more information on viewing and resolving conflicts, seeSection 3.9, "Viewing and Resolving Conflicts."

10. Click **Import** to start the import process.

    A pop-up window is displayed indicating the progress of the import. When the import is complete, the Import Summary page displays the result of the import. A success message is displayed at the top of the page if the import was successful. If one or more resources could not be imported, an error message is displayed.

- Go to the Status column for the resource to view the cause of the error.

- To view and resolve additional conflicts, click **View Conflicts**.

11. To import more services, click **Import Another.**

12. To make changes to your selections, click **Back**.

30.5 Using Auto-Import Status

You can use the Auto-Import Status page to synchronize changes to a service with those present in the registry. Upon any changes to a service in the registry, Oracle Service Bus provides notification of the change on the Auto-Import Status page which lists all out-of-sync services. You can then synchronize the service in the Oracle Service Bus Administration Console with the corresponding service in the UDDI registry.

When configuring a registry, you can select the **Enable Auto-Import** option which auto-synchronizes imported services with the UDDI registry. Any service that is imported with this option selected will be kept in synchrony with the UDDI registry automatically. See Section 30.2, "Configuring UDDI Registries."

If there is any failure during auto-synchronization, it will be reported on the Auto-Import Status page where you can update it manually.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Auto-Import Status**.

3. On the Auto-Import Status page, select the type of service you want to display from the **View** list.

4. Select the check boxes next to the proxy services you want to synchronize with the corresponding services in the registry.

5. Click **Synchronize**.

6. On the next page, click **Finish**.

30.6 Detaching Services

When you do not want the service in the Oracle Service Bus Administration Console synchronized with the corresponding service in the registry, you can avoid synchronization by detaching it from the registry in one of the following ways:

- "Detaching a Service by Editing its Configuration"

- "Detaching a Service From the Auto-Import Status Page"

Detaching a Service by Editing its Configuration

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select the **Project Explorer**.

3. Expand the project folder and select the business service you want to detach from the corresponding service in the registry.

4. At the bottom of the View a Business Service page, click **Edit**.

5. Enable the **Detach from Registry** option. The business service will be detached from the corresponding service in the UDDI registry.

Detaching a Service From the Auto-Import Status Page

You can use this Detach option only when you get notification about the change on the Auto-Import Status page.

1. On the **System Administration** > **Auto-Import Status** page, select the check boxes next to the business services you want to detach from the corresponding services in the UDDI registry.

   Services are shown on this page only when there is a change in the original service present in the registry. Not every service is available on this page.

2. Click **Detach**.

30.7 Publishing Proxy Services to a UDDI Registry

You can publish your service to a registry and make it available for other organizations to discover and use. All proxy services developed in Oracle Service Bus can be published to a UDDI registry. You can select the business entity under which you want to publish your service and you can publish a number of services at a time.

Before you can publish to a registry, you must have an account with that registry. Oracle Service Bus supports interoperability with version 3.0-compliant UDDI registries. The configuration described in this section assumes you are using Oracle Service Registry.

For information about setting up a user account in Oracle Service Registry, see the Oracle Service Registry product documentation, accessible from http://www.oracle.com/technetwork/middleware/registry/overview/index.html.

Unpublishing a service from a registry is done from the Oracle Service Registry installation.

1. Select **System Administration** > **Publish to UDDI**.

   You can only publish when you are not in a session. Exit your session to access the registries list. All the registries to which you have access are displayed in the list of selected registries.

2. From **Publish Registry Name**, select the name of the registry to which you want to publish, then click **Next**.

   When a single registry is configured in Oracle Service Bus, the initial registry selection page is not displayed as this is the default registry. The registry selection page is only displayed when two or more registries are defined.

   The Publish to UDDI: Select individual Services and Publish page is displayed, showing a list of Oracle Service Bus proxy services and their locations in the project folder. Starting with the topmost project level folder, the details for name, type, and description are shown for all services.

3. Select the service(s) that you want to publish.

   a. Expand the **Project** folder to see the proxy services defined. By default the folder and its contents are marked for publishing.

   b. You can select individual items to publish.

4. From the **Publish Services to Business Entity** list, select the business entity to which you want to add the service in the registry. This is the business entity under which the service will be classified in the registry.

5. Click **Publish** to publish the services.

A confirmation message is displayed indicating that the service was published successfully.

6. Click **Publish Another** to return to the Publish to UDDI: select registry page.

30.8 Using Auto-Publish

You can use auto-publish to automatically publish proxy services to a registry. To do so, you must enable the **Publish to Registry** option on the Create/Edit a Proxy Service-General Configuration page. See Section 20.2.1, "General Configuration Page." The **Publish to Registry** option only appears when a default registry exists. See Section 30.3, "Setting Up a Default UDDI Configuration."

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select the **Project Explorer**.

3. Expand the project folder and select the proxy service you want to edit.

4. At the bottom of the View a Proxy Service page, click **Edit**.

5. Select **Publish to Registry**, and click **Save** to commit the updates in the current session.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

   The service is configured to be published to the default registry.

Notes: When you have successfully published a service to a registry, the option to select the service for auto-publishing to a registry is no longer available.

If the service is not successfully published it can be re-published. To re-publish a service, select the service on the Auto-Publish Status page and click **Publish**.

If the **Publish to Registry** option is enabled, the proxy services are published as soon as they are created or edited, and the session is activated. You can use the **Publish to Registry** option with all proxy services, except those using Local Transport.

31

Global Resources

This chapter describes how to add and configure global resources in the Oracle Service Bus (OSB) Administration Console, such as JNDI Providers, SMTP Servers, and Proxy Servers. Global resources can be reused by OSB services.

This chapter includes the following sections:

- Section 31.1, "Viewing JNDI Providers"
- Section 31.2, "Adding JNDI Providers"
- Section 31.3, "Editing JNDI Providers"
- Section 31.4, "Deleting JNDI Providers"
- Section 31.5, "Viewing SMTP Servers"
- Section 31.6, "Locating SMTP Servers"
- Section 31.7, "Adding SMTP Servers"
- Section 31.8, "Configuring a Default SMTP Server"
- Section 31.9, "Editing SMTP Servers"
- Section 31.10, "Deleting SMTP Servers"
- Section 31.11, "Viewing Proxy Servers"
- Section 31.12, "Locating Proxy Servers"
- Section 31.13, "Adding Proxy Servers"
- Section 31.14, "Editing Proxy Servers"
- Section 31.15, "Deleting Proxy Servers"

31.1 Viewing JNDI Providers

Use this page to view a list of JNDI Providers and the information shown in Table 31–1.

Additionally, you can search for JNDI Providers that you have previously configured.

1. To restrict the number of items in the list or locate specific JNDI Providers:

   - Click the **Open** icon to display additional search filters. If you specify multiple search criteria, only results that match all the criteria will be returned.

   - Enter the name of the JNDI Provider you want to find in the **Name** field, and click **Search** to search for a specific entry.

You can enter wildcard characters (? for a single character; * for multiple characters) to perform a more general search.

- Enter the **Provider URL** of the JNDI Provider you want to find, and click **Search**.

2. Click **View All** to remove the search filters and display all the JNDI Provider resources that are configured to work with Oracle Service Bus.

Table 31–1 displays UDDI registry configuration settings.

Table 31–1   JNDI Provider Information

Property	Description
Name	The unique name assigned to this JNDI Provider. The name is a link to the View JNDI Provider - Configuration page. See Section 31.3, "Editing JNDI Providers."
Provider URL	The URL of the JNDI Provider. See Section 31.2, "Adding JNDI Providers."
JNDI Cache	Enabled or Disabled. When enabled, the JNDI context and JNDI objects are cached locally which improves performance when doing an object lookup. Oracle recommends that you keep the JNDI cache enabled.
Actions	■ Click the **Reset** icon to reset the JNDI context to discard the JNDI connection and locally cached objects. ■ Click the **Launch Test Console** icon to test whether the JNDI Provider can successfully establish a connection and obtain a JNDI context.
Options	■ Click the **Rename** icon to rename the JNDI Provider. ■ Click the **Delete** icon to delete a specific JNDI Provider. See Section 31.4, "Deleting JNDI Providers."

31.2 Adding JNDI Providers

You add and configure a JNDI Provider resource and make it available in Oracle Service Bus as a system resource. You must be in an active session.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **JNDI Providers**.

3. Click **Add**.

4. In the **Name** field, enter a name for the JNDI Provider resource. This is a required field.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Description** field, enter a short description for the JNDI Provider.

6. Keep the **JNDI Cache** option Enabled (default) or select Disabled.

   When enabled, the JNDI context and JNDI objects are cached locally which improves performance when doing an object lookup. Oracle recommends that you keep the JNDI cache enabled.

7. In the **Provider URL** field, enter the URL for the JNDI Provider in the format: `protocol://host:port`

   You can use any protocol, for example, `http`, `https`, `t3`, `t3s`, `iiop`, `iiops`.

This is a required field.

8. In the case of clusters, the JNDI Provider URL should be configured with a comma-separated list of Managed Servers, in the format:

   protocol://<hostname>:<ms1port>, <hostname>:<ms2port>

9. For **JNDI Request Timeout**, enter the number of milliseconds before a request connection times out. Zero (0) means no time limit.

10. If access to the target JNDI Provider requires a user name and password, enter a user name in the **User Name** field, and the associated password in the **Password** and **Confirm Password** fields.

    These fields are optional, and required only if the JNDI tree is secured.

11. Click **Save** to create and save the JNDI Provider resource in the current session.

12. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.3 Editing JNDI Providers

Use this page to view the details of a JNDI Provider and edit the configuration, if required. Additionally, you can reset the JNDI Provider cache and to test the connection to the JNDI Provider.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the JNDI Provider. See Section 31.1, "Viewing JNDI Providers."

3. Click the JNDI Provider name.

   The View JNDI Provider - Configuration page displays the JNDI Provider information shown in Table 31–2.

Table 31–2   JNDI Provider Information

Property	Description
Last Modified By	The user who created or modified this JNDI Provider resource, or imported it into the configuration.
Last Modified On	The date and time that the user created or modified this JNDI Provider resource, or imported it into the configuration. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this JNDI Provider references. If such references exist, click the numeric link to view a list of the references. See Section 4.22, "Viewing References to Resources."
Referenced By	The number of objects that reference this JNDI Provider. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	A description of this JNDI Provider.
JNDI Cache	Enabled or Disabled. When enabled, the JNDI context and JNDI objects are cached locally which improves performance when doing an object lookup. Oracle recommends that you keep the JNDI cache enabled.
Provider URL	The URL for the JNDI Provider.

Table 31–2 (Cont.) JNDI Provider Information

Property	Description
JNDI Request Timeout	The JNDI request timeout in milliseconds. The default of zero (0) means no timeout.
User Name and password	The user name used to access a secure JNDI Provider, with the ability to change the password.

4. Click **Edit** to display the Edit JNDI Provider - [provider-name] page.

   The **Name** field is not editable.

5. In the **Description** field, you can edit the short description of the JNDI Provider.

6. In the **Provider** URL field, you can edit the URL of the JNDI Provider.

7. You can edit the **User Name** and **Password** fields.

   If the JNDI Provider was configured with a user name and password, then the User Name will not be editable. You can edit the **New Password** and **Confirm Password** fields. If the JNDI was *not* configured with a user name and password, then the **User Name** field will be editable, and you can enter a new user name.

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.4 Deleting JNDI Providers

To delete a JNDI provider:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **JNDI Providers**.

3. In the Summary of JNDI Providers page, select a JNDI Provider. You can also search for a specific JNDI Provider using the **Search** option. See Section 31.1, "Viewing JNDI Providers."

4. Click the **Delete** icon in the Options field of the JNDI Provider you want to delete. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.5 Viewing SMTP Servers

Use this page to designate a default SMTP Server (see Section 31.8, "Configuring a Default SMTP Server") and to view a list of SMTP Servers.

SMTP Server resources are used while configuring alert destination resources and Email transport-based business services. See Section 6.2, "Adding Alert Destinations".

Table 31–3 displays SMTP Server information.

Table 31–3   SMTP Server Information

Property	Description
Name	The unique name assigned to this SMTP Server resource. The name is a link to the View SMTP Server - Configuration page. See Section 31.9, "Editing SMTP Servers."
Server URL	The URL that points to the SMTP Server. See Section 31.7, "Adding SMTP Servers."
Options	■ Click the **Rename** icon to rename this SMTP Server resource. ■ Click the **Delete** icon to delete a specific SMTP Server. See Section 31.10, "Deleting SMTP Servers."

31.6 Locating SMTP Servers

To locate SMTP servers:

1.   Select **System Administration** > **SMTP Servers**.

2.   To locate specific SMTP Servers:

■   Resort the list. Click an underlined column name. Ascending and descending arrows indicate the sort order. Click the column name to change the sort order.

■   Scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

31.7 Adding SMTP Servers

Use the Summary of SMTP Servers page to add and configure SMTP Server resources and make them available in Oracle Service Bus as a system resource. You must be in an active session to configure or reconfigure SMTP Server resources.

1.   If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2.   Select **System Administration** > **SMTP Servers**.

3.   Click **Add**.

4.   In the **Name** field, enter a name for the SMTP Server resource. This is a required field.

Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5.   In the **Description** field, enter a short description for the SMTP Server resource.

6.   In the **Server URL** field, enter the URL that points to the SMTP Server. This is a required field.

7.   In the **Port Number** field, enter a port number for the SMTP Server (the default port is 25). This is a required field.

8.   If access to the target SMTP Server requires a user name and password, enter a user name in the **User Name** field, and the associated password in the **Password** and **Confirm Password** fields.

These fields are optional, and required only if the SMTP Server is secured.

9.   Click **Save** to create and save the SMTP Server resource in the current session.

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.8 Configuring a Default SMTP Server

You can designate one of the configured SMTP Servers as the default server for the domain. See Section 31.7, "Adding SMTP Servers."

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **SMTP Servers**.

3. In the Default SMTP Server panel, click **Select Default**.

4. From the **Default SMTP Server** list, select an SMTP Server.

5. Click **Make Default**.

   The Summary of SMTP Servers page is displayed with the selected SMTP Server shown as the default SMTP Server for the domain.

6. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.9 Editing SMTP Servers

Use this page to view the details of an SMTP Server and edit the configuration, if required.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the SMTP Server. See Section 31.6, "Locating SMTP Servers."

3. Click the SMTP Server name.

   The View SMNP Servers - Configuration page displays the information shown in Table 31–4.

Table 31–4   SMTP Server Information

Property	Description
Last Modified By	The user who created or modified this SMTP Server resource.
Last Modified On	The date and time that the user created or modified this SMTP Server resource. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this SMTP Server references.If such references exist, click the numeric link to view a list of the references. See Section 4.22, "Viewing References to Resources."
Referenced By	The number of objects that reference this SMTP Server. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	The description of this SMTP Server, if provided.
Server URL	The URL that points to the location of the SMTP Server.
Port Number	The port number for this SMTP Server. This is port 25 by default.
User Name	The user name used to access a secure SMTP Server.

4. Click **Edit** to display the Edit SMTP Server - [server-name] page.

   The **Name** field is not editable.

5. In the **Description** field, you can edit the short description of the SMTP Server.

6. In the **Server URL** field, you can edit the URL of the SMTP Server.

7. In the **Port Number** field, you can edit the port number.

8. You can edit the **User Name** and **Password** fields.

9. Click **Save** to commit the updates in the current session.

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

If you rename an SMTP resource, the new name is automatically updated in the any email business services that reference the SMTP resource in the endpoint URI.

31.10 Deleting SMTP Servers

To delete an SMTP server:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **SMTP Servers**.

3. On the Summary of SMTP Servers page, select an SMTP Server. See Section 31.6, "Locating SMTP Servers."

4. Click the **Delete** icon in the Options field of the SMTP Server resource you want to delete. The SMTP Server resource is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.11 Viewing Proxy Servers

Use this page to view a list of Proxy Servers and the information shown in Table 31–5.

Proxy Server resources are used while configuring the HTTP transport for business services. For more information, see Section 19.2.12, "HTTP Transport Configuration Page."

Table 31–5 displays Proxy Server information.

Table 31–5   Proxy Server Information

Property	Description
Name	The unique name assigned to this Proxy Server resource. The name is a link to the View Proxy Server - Configuration page. See Section 31.14, "Editing Proxy Servers."
Options	■ Click the **Rename** icon to rename this Proxy Server resource. ■ Click the **Delete** icon to delete a specific Proxy Server. See Section 31.15, "Deleting Proxy Servers."

31.12 Locating Proxy Servers

To locate proxy servers:

1. Select **System Administration** > **Proxy Servers**.

2. To locate specific Proxy Servers:

   - Resort the list. Click an underlined column name. Ascending and descending arrows indicate the sort order. Click the column name to change the sort order.

   - Scroll through the pages. Use the page controls above or below the table. Go to a page by selecting the page number or by using the arrow buttons to go to the next, previous, first, or last page.

31.13 Adding Proxy Servers

Use the Summary of Proxy Servers page to add and configure Proxy Server resources and make them available in Oracle Service Bus as a system resource. You must be in an active session to configure or reconfigure Proxy Server resources.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Proxy Servers**.

3. Click **Add**.

4. In the **Name** field, enter a name for the Proxy Server resource. This is a required field.

   Follow the Section 2.3, "Resource Naming Restrictions" for naming guidance.

5. In the **Description** field, enter a short description for the Proxy Server resource.

6. In the Host-Port Parameters section, enter the following information:

   a. In the **Server Host** field, enter the host name or IP address of the Proxy Server. This is a required field.

      The Server Host name for the Oracle Service Bus proxy server must be identical to the server host name of the actual proxy server.

   b. In the **Clear Text Port** field, enter the Proxy Server clear-text port number.

   c. In the **SSL Port** field, enter the Proxy Server SSL port number. You must enter either a clear text or SSL port number.

   d. Click **Add**.

   You can configure multiple Proxy Servers for each Proxy Server resource. This enables Oracle Service Bus to perform load balancing and offer fault tolerance features for the Proxy Server resource.

7. If the Proxy Server performs proxy authentication, enter a user name in the **User Name** field, and the associated password in the **Password** and **Confirm Password** fields.

   These fields are optional, and required only if the Proxy Server is secured.

8. Click **Save** to create and save the Proxy Server resource in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.14 Editing Proxy Servers

Use this page to view the details of a Proxy Server and edit the configuration, if required.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Locate the Proxy Server. See Section 31.12, "Locating Proxy Servers."

3. Click the Proxy Server name.

   The View Proxy Servers - Configuration page displays the information shown in Table 31–6.

Table 31–6   Proxy Server Information

Property	Description
Last Modified By	The user who created or modified this Proxy Server resource.
Last Modified On	The date and time that the user created or modified this Proxy Server resource. Click the date and time link to view the change history of this resource. See Section 4.23, "View Change History Page."
References	The number of objects that this Proxy Server references.If such references exist, click the numeric link to view a list of the references. See Section 4.22, "Viewing References to Resources."
Referenced By	The number of objects that reference this Proxy Server. If such references exist, click the numeric link to view a list of the objects. See Section 4.22, "Viewing References to Resources."
Description	The description of this Proxy Server, if provided.
Host-Port Parameters	Details about each Proxy Server in the Proxy Server resource, including the following information: ■ Server Host: The host name or IP address of the Proxy Server. The Server Host name for the Oracle Service Bus proxy server must be identical to the server host name of the actual proxy server. ■ Clear Text Port: The Proxy Server clear-text port number. ■ SSL Port: The Proxy Server SSL port number.
User Name	The user name used for proxy authentication.

4. Click **Edit** to display the Edit Proxy Server - [server-name] page.

   The Name field is not editable.

5. In the **Description** field, you can edit the short description of the Proxy Server.

6. In the Host-Port Parameters section, you can add a new Proxy Server by entering the following information:

   a. In the **Server Host** field, enter the host name or IP address of the Proxy Server. This is a required field.

      The Server Host name for the Oracle Service Bus proxy server must be identical to the server host name of the actual proxy server.

   b. In the **Clear Text Port** field, enter the Proxy Server clear-text port number.

   c. In the **SSL Port** field, enter the Proxy Server SSL port number. You must enter either a clear text or SSL port number.

   d. Click **Add**.

   You can configure multiple Proxy Servers for each Proxy Server resource. This enables Oracle Service Bus to perform load balancing and offer fault tolerance features for the Proxy Server resource.

   You can also edit or delete an existing Proxy Server configuration by doing the following:

   - Click the **Edit** icon in the **Options** field of the Proxy Server you want to edit. Edit the **Server Host** field, the **Clear Text Port** field, and the **SSL Port** field, and click **Update**.

   - Click the **Delete** icon in the **Options** field of the Proxy Server you want to delete.

7. You can edit the **User Name** and **Password** fields, used for proxy authentication.

8. Click **Save** to commit the updates in the current session.

9. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

31.15 Deleting Proxy Servers

To delete a proxy server:

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Proxy Servers**.

3. On the Summary of Proxy Servers page, select a Proxy Server. See Section 31.6, "Locating SMTP Servers."

4. Click the **Delete** icon in the Options field of the Proxy Server resource you want to delete. The Proxy Server resource is deleted in the current session. A Deletion Warning icon is displayed when other resources reference this resource. You can delete the resource with a warning confirmation. This might result in conflicts due to unresolved references to the deleted resource.

5. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

32

Customization

This chapter provides details on customizing an Oracle Service Bus (OSB) environment. Instructions include finding and replacing the values of environment variables associated with different resources, and creating customization files that invoke automated changes to resources and environment values when moving an OSB configuration from one environment to another.

This chapter includes the following sections:

- Section 32.1, "Finding and Replacing Environment Values"
- Section 32.2, "Creating Customization Files"
- Section 32.3, "Executing Customization Files"

32.1 Finding and Replacing Environment Values

Use the Find and Replace page to search for environment values that differ between domains. Environment values are certain predefined fields in the configuration data whose values are very likely to change when you move your configuration from one domain to another (for example, from test to production). Environment values represent entities such as URLs, URIs, file and directory names, server names, emails, and such. Also, environment values can be found in alert destinations, proxy services, business services, SMTP Server and JNDI Provider resources, and UDDI Registry entries.

This page behaves differently based on whether or not you are in a session. If you are in a session, you can find and replace environment values. However, if you are outside a session, you can find environment values only; **Replace All** is disabled.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Find & Replace**.

3. In the **Find Value** field, enter the environment value that you want to locate.

4. To display a list of environment values in your configuration that contain the value you entered, click **Find**.

   A list of relevant values is displayed, which includes the Owner, Type, and the Environment Value columns.

5. To locate only items changed in your current session, select the **Only Items Changed In Current Session** check box.

6. Select the variable type for which to search, from the **Find in Variable Type** list, described in Table 32–1.

Table 32–1   Variable Types

Variable Type	Found In	Description
Alert SNMP Trap	Alert Destination	SNMP trap (enable or disable) for an alert destination.
Alert to Log	Alert Destination	Alert Logging (enable or disable) for an alert destination.
Alert To Reporting Data	Alert Destination	Alert to reporting data (enable or disable) for an alert destination.
Email Archive Directory	Email proxy	Archive directory for an email proxy service.
Email Destination URI	Alert Destination	URI in an email alert destination.
Email Download Directory	Email proxy	Download directory for an email proxy service.
Email Error Directory	Email proxy	Error directory for an email proxy service.
FTP Archive Directory	FTP proxy	Archive directory for an FTP proxy service.
FTP Download Directory	FTP proxy	Download directory for an FTP proxy service.
FTP Error Directory	FTP proxy	Error directory for an FTP proxy service.
File Archive Directory	File proxy	Archive directory for a file proxy service.
File Error Directory	File proxy	Error directory for a file proxy service.
File Stage Directory	File proxy	Stage directory for a file proxy service.
IMAP Move Folder	Email proxy	IMAP Move directory for an email proxy service.
JCA Always Use WSDL Flag	JCA proxy and business service	Determines whether connection factory properties, activation spec properties (proxy services), and interaction spec properties (business services) are always used from the WSDL.
JCA Connection Mode	JCA proxy and business service	Determines how a service connects to an associated JCA adapter: managed or non-managed mode.
JCA Overwrite Connection Authentication Flag	JCA proxy and business service	Determines whether the authentication credentials in the JCA adapter connection factory are overridden in a development/test environment (non-managed connection mode).
JMS Alert Destination URI	Alert Destination	URI in a JMS alert destination.
JMS Managed Server	JMS business service	Managed Server entry in the table of Managed Server vs. destination of a JMS business service with a response correlation pattern as Message ID pattern.
JMS Queue Connection Factory	JMS proxy and business service	A response queue connection factory of a JMS proxy or business service with the response correlation pattern as Message ID.
JMS Response Destination	JMS business service	Destination entry in the table of Managed Server vs. destination of a JMS business service with a response correlation pattern as Message ID pattern.

Table 32–1   (Cont.) Variable Types

Variable Type	Found In	Description
JMS Response URI	JMS proxy and business service	The URI of the response queue for JMS proxy or business services using response correlation pattern of JMS Correlation ID.
JNDI Provider URL	JNDI Provider	JNDI Provider URLs.
MQ Connection Pool Size	MQ Connection Resource	The size of the MQ connection pool.
MQ Connection Timeout	MQ Connection Resource	The time interval after which unused connections are destroyed.
MQ Dead Letter URI	MQ proxy	The URI of the dead letter queue to which request messages are redirected after a pipeline retries a message a specified number of times.
MQ Host Name	MQ Connection Resource	The host name of the MQ queue manager.
MQ Port Number	MQ Connection Resource	The port number of the MQ queue manager listener.
MQ Queue Manager Channel Name	MQ Connection Resource	The queue manager server connection channel name.
MQ Queue Manager Name	MQ Connection Resource	The name of the MQ queue manager.
MQ Response URI	MQ proxy and business service	Proxy or business service URI.
MQ Unrecognized Response URI	MQ business service	The URI of the queue to which unrecognized response messages are sent.
MQ Version	MQ Connection Resource	The version of WebSphere MQ being used.
Managed Server for Polling	Email, File, FTP, or SFTP proxy service in a clustered domain	Managed Server for polling in a clustered domain.
Proxy Server Host	Proxy Server Resource	The host name of the proxy server.
Proxy Server Port	Proxy Server Resource	The port number of the proxy server.
Proxy Server SSL Port	Proxy Server Resource	The SSL port of the proxy server.
SFTP Archive Directory	SFTP proxy	Archive directory for a SFTP proxy service. If direct-streaming is on, the archive directory is present on the remote SFTP server; otherwise, it is present locally.
SFTP Download Directory	SFTP proxy	Download directory for a SFTP proxy service.
SFTP Error Directory	SFTP proxy	Error directory for a SFTP proxy service. If direct-streaming is on, the error directory is present on the remote SFTP server; otherwise, it is present locally.
SMTP Server URL	SMTP Server	SMTP Server URLs.

Table 32–1   (Cont.) Variable Types

Variable Type	Found In	Description
Secure Connections to JMS Server	JMS proxy or business service	If true, Oracle Service Bus connects to the JMS server and JNDI tree using SSL (t3s); otherwise, connections occur over a clear text (t3) channel.
Service Retry Count	Business service	The number of times endpoint URIs are retried; in other words, the number of failover attempts.
Service Retry Iteration Interval	Business service	The length of time that a business service waits before iterating over the entire set of URIs again.
Service URI	Proxy or business service	Proxy or business service URI.
Service URI Weight	Business service	The individual weights assigned to business service URIs.
Split-Join Work Manager	Split-Join	Work Manager name (Dispatch Policy) for a split-join resource.
Tuxedo Access Point Map	Proxy or business service	Name and address of the local access point(s) per Managed Server; there is one location per URI.
Tuxedo Access Point Name	Business service	Name of the remote WTC access point associated with the URI.
Tuxedo Network Address	Business service	Network address of the remote WTC access point associated with the URI.
UDDI Auto Import	UDDI Registry	Auto-synchronize property of an imported business service from a UDDI Registry. This property is per registry.
UDDI Auto Publish	Proxy service	Auto-publish property of a proxy service to a UDDI Registry.
UDDI Inquiry URL	UDDI Registry	Inquiry URL for a UDDI Registry.
UDDI Publish URL	UDDI Registry	Publish URL for a UDDI Registry.
UDDI Security URL	UDDI Registry	Security URL for a UDDI Registry.
UDDI Subscription URL	UDDI Registry	Subscription URL for a UDDI Registry.
WS Error Queue URI	WS business service	The URI of the JMS queue for storing error messages.
Work Manager	Proxy or business services	Work Manager name (Dispatch Policy) in all proxy and business services.

7. To locate environment values located in a particular project, select the project name from the **Located in Project** list.

8. In the **Replace with** field, enter the new environment value.

9. To replace the original environment value with the new value, click **Replace All**.

All occurrences of the environment value you entered in the **Find Value** field are replaced with the environment value you entered in the **Replace with** field in the current session.

> **Note:** Certain environment values are complex XML objects that cannot be found and replaced using the Administration Console. However, you can still set these environment values directly by using the ALSBConfigurationMBean from a script. For detailed information about ALSBConfigurationMBean, see the Oracle Service Bus API Reference. In addition to setting them through the API, you can set complex type environment values using customization files. See Section 32.3, "Executing Customization Files."

10. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

Related Topics

For detailed information about EnvValueTypes, see the Oracle Service Bus API Reference. The Java types and location values of these environment values are specified in the API reference.

32.2 Creating Customization Files

The Create Customization File page provides a convenient way to generate a customization file for a set of resources or projects that you select. You can then use this file as a starting point for making your desired modifications by specifying the actual values for an environment.

You can use customization files to make changes to environment values as well as to change references within resources. Customization files can include customizations for all the environment values found in the resources you selected, including complex environment values types defined inside the `EnvValueTypes` class. In addition, it includes a reference customization type for changing resource references inside resources with dependencies.

The customization schema (`Customization.xsd`), which describes the customization types, is available at the following location in your Oracle Service Bus installation:

OSB_ORACLE_HOME\modules\com.bea.common.configfwk_*version*.jar.

To create a customization file:

1. Select **System Administration** > **Create Customization File**.

   The Create Customization File page displays the list of objects in your configuration. The name and type of each object is displayed.

2. Select the projects or resources you want to include in the customization file.

   a. Expand the project folders. The name and type for each resource contained in the project is displayed.

   b. Select the check boxes associated with the projects or resources you want to include in the customization file.

   c. Clear the check boxes associated with the projects or resources that you do not want to include in the customization file.

3. Click **Create File** to create a customization XML file.

4. In the File Download dialog box, click **Open** to open the customization file or click **Save** to save the XML file to your desktop.

5. In an XML editor, use the customization schema in conjunction with the base customization file you created to make the necessary modifications to resources and environment variables.

Your base customization file may already be populated with environment variables. Table 32–1 shows which environment variables are automatically included with different resources.

32.3 Executing Customization Files

Use this page to execute a customization XML file that has been previously saved on your system. Customization files provide a convenient way to modify your environment during deployment, from development to staging, from staging to production, or during design time.

You must be in a session to execute a customization file.

1. If you have not already done so, click **Create** to create a new session or click **Edit** to enter an existing session. See Section 3.1, "Using the Change Center."

2. Select **System Administration** > **Execute Customization File**.

3. Select a customization file to execute.

   a. In the **File Name** field, click **Browse** to locate the directory where the customization file is stored.

   b. Select the XML file, then click **Open**.

4. Click **Next**. The following customization details are displayed, shown in Table 32–2.

Table 32–2   Customization Details

Property	Description
Operation	■ Find & Replace—modify environment values by doing string substitutions. ■ Map References—maps existing references found in resources to other references. ■ Set Environment Values—modify environment values for resources.
Apply To	Name of the service, project, folder, or resource to be customized. The name is a link to the Configuration Details page of the service or resource, or to the project or folder in the Project Explorer.
Description	A description of each customization.

5. To customize only resources, services, projects, or folders changed in the current session, select the **Only Items Changed In Current Session** check box.

If you limit customizations to only resources modified in the current session, the Apply To column gets updated with the resources modified in current session. If you limit customizations of a project or folder, then the resources modified in current session *within* that project or folder are displayed in Apply To column.

6. Click **Execute** to commit the updates in the current session.

To view customization details, within the session, select **View Changes** under Change Center, and click the **Customization** task. See Section 3.4, "Viewing Configuration Changes."

7. To end the session and deploy the configuration to the runtime, click **Activate** under Change Center.

33

Test Console

This chapter describes how to test services using the Oracle Service Bus (OSB) Test Console from the OSB Administration Console.

The Oracle Service Bus Test Console is a browser-based test environment used to validate and test the design of your system. You can configure the object of your test (proxy service, business service, or resource), execute the test, and view the results in the Test Console. In some instances, you can trace through the code and examine the state of the message at specific trace points. To learn more about the Test Console, see "Using the Test Console" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

Only users in the `IntegrationAdmin` and `IntegrationDeployer` roles can use the Test Console. For more information about roles, see "Roles in Oracle Service Bus" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

> **Note:** If you receive an error saying the Test Console service is not running, try setting the Admin server listen address to a specific valid address, such as localhost. In the Oracle WebLogic Server Console, go to **Environment** > **Servers** > *admin_server_name* > **Configuration** > **General** to set the Listen Address.

This chapter contains the following sections:

- Section 33.1, "Testing Services"
- Section 33.2, "Testing Transformations"
- Section 33.3, "Performing XQuery Testing"
- Section 33.4, "Understanding How the Runtime Uses the Transport Settings in the Test Console"

33.1 Testing Services

This section includes the following topics:

- Section 33.1.1, "Testing Proxy Services"
- Section 33.1.2, "Configuring Proxy Services Test Data"
- Section 33.1.3, "Viewing Proxy Services Test Results"
- Section 33.1.4, "Tracing Proxy Services"
- Section 33.1.5, "Testing Business Services"

- Section 33.1.6, "Configuring Business Services Test Data"

33.1.1 Testing Proxy Services

You can test the following types of proxy services: any XML, any SOAP, Messaging, XML, and SOAP. You can test SOAP proxy services with Web Service Security (WSS) policies. See Web Service Security in Section 33.1.2, "Configuring Proxy Services Test Data."

Note: When the Test Console invokes a proxy with HTTP custom token authentication, the authentication check is not done.

Caution: Testing proxy services with the direct call option enabled bypasses some important security steps, including access control. Oracle recommends that you not use the test framework in production systems. For information on undeploying the Test Console, see Section 40.11, "Undeploying the Test Console."

---

1. Click **Activate** under Change Center to enable the test feature in the Test Console.

   You can test proxy services from the Resource Browser or Project Explorer.

2. Select **Resource Browser** > **Proxy Services** to display the Summary of Proxy Services page.

3. Under Actions, click the **Launch Test Console** icon associated with the proxy service you want to test.

   The Test Console opens the Proxy Service Testing page. For example, using the examples provided with the product, click the icon associated with the `LoanGateway1` proxy service.

Note: In a clustered domain, you cannot use the Test Console to test any configured business service or proxy service which routes to a business service.

---

4. For SOAP and XML services, select the WSDL operation you want to test.

5. Configure the test data for the proxy service. This must be the data that the proxy service expects from the client.

   By default, both test configuration options, **Direct Call** and **Include Tracing**, are enabled. You can clear the **Direct Call** option, which also clears the **Include Tracing** option. By doing so, testing is performed using the indirect call method where the message is sent through the transport layer.

   You can use the Direct Call option (leave **Direct Call** selected) and disable Execution tracing; simply clear the **Include Tracing** check box.

   To see message tracing, clear **Direct Call**.

> **Note:** To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:
>
> - Minimum severity to log: Info
> - Log file: Info
> - Standard out: Info
>
> For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

6. Click **Execute**. The Proxy Service Testing page displays the results. For information about interpreting the test results, see Section 33.1.3, "Viewing Proxy Services Test Results."

7. To run the test again, click **Back**. Repeat steps 5 and on.

33.1.2 Configuring Proxy Services Test Data

Table 33–1 lists the configuration options for testing proxy services. The fields for accepting input to the request document are based on the service type.

Table 33–1 Proxy Services Configuration Options

Section	Options/Fields
Name	The name of the proxy service being tested.
Available Operations	Operations associated with the proxy service.
Test Console Actions	- Execute - Run the test. - Reset - Reset the input values. - Close - Close the Test Console and do not run the test.
Test Configuration	Set the testing configuration options. ■ Direct Call - Send a message to the proxy service without using the Oracle Service Bus transport layer. The input data to the Test Console must be that which is sent from the client to the proxy service. The opposite of the direct call is an indirect call. It can be invoked by clearing the **Direct Call** option. It is performed by default for business services. The indirect call sends messages through the transport layer. In this case, the input data to the Test Console must be that which is being sent from a proxy service to the invoked service. ■ Include Tracing - Tracing shows the state of the message as it passes through the system.

Table 33–1   (Cont.) Proxy Services Configuration Options

Section	Options/Fields
Request Document	The input fields generate the request message that is sent to the proxy service. Click **Execute** to run the test with the values you entered. The Test Console displays the request message and the service's response message and metadata.
	Input in the Request Document section is service-type specific. The service types and a description of the input required by each are listed in the following sections.
	■ any XML - The service requests input in the form of a payload.The payload is the content of the message being sent. The content is expected to be an XML document. You can browse to a file or enter the message content in the text box.
	■ any SOAP - The service requests input in the form of a payload. The payload is the content of the message being sent. The content is expected to be the SOAP envelope. You can browse to a file or enter the message content in the text box.
	■ Messaging - Messaging services define four possible input types: `none`, `XML`, `Binary` or `Text`. The service requests a single input—either file-based or text-based, except for the type `none` that does not require any input.
	Oracle recommends entering binary input from a file. Data entered in the text area is converted to binary input using the system encoding.
	Data entered from files for text services must be converted to text. The encoding input field is used to specify the encoding to apply during the conversion. The system encoding is used if this field is not configured. By default, the encoding field is initialized with the encoding value configured on the proxy service endpoint.
	■ XML - The service requests input in the form of a payload.The payload is the content of the message being sent. The content is expected to be an XML document. You can browse to a file or enter the message content in the text box.
	If your proxy service is a WSDL-based service with multiple operations defined, the Test Console generates and provides a sample document to use when testing the service. You can use this sample data directly, edit it and then run the test, or provide your own test data.
	All operations are listed at the top of the page.
	■ SOAP Document - For a SOAP document, the SOAP envelope is usually composed of zero or more headers and one body payload. The Form and XML tabs provide alternative ways to specify the content.
	The Form tab contains a SOAP Header field and a SOAP Body field. The content of the SOAP Header field is expected to be a SOAP Header tag (this allows for the definition of more than one header). The SOAP Body field contains the data that is actually sent as part of the message. The content is expected to be an XML document. Both the header and the body are used to generate the SOAP envelope.
	■ SOAP RPC - For SOAP RPC, the SOAP envelope is composed of zero or more headers, and zero or more arguments.
	The Form and XML tabs provide alternative ways to specify the content.
	The Form tab contains a single input for SOAP headers, and one input field for each argument (the name of the input field corresponds to the name of the argument). The content of the SOAP Header field is expected to be a SOAP Header tag (this allows for the definition of more than one header).
	The WSDL is used to detect the type of each argument. A single-line input field is used for primitive types; a multi-line input field is used for XML types. A sample document is automatically generated to facilitate testing.
	The headers and arguments are used by the Test Console to generate the SOAP envelope.
	The XML tab contains a single input field. The content of this field is expected to be the SOAP envelope being sent. The payload (XML input) can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test.

Table 33–1   (Cont.) Proxy Services Configuration Options

Section	Options/Fields
Web Service Security	This section is available only for SOAP services when the selected operation has a Web Service Security (WSS) policy.
	■ Service Provider - The test service gets all client-side PKI (key-pair) credentials for Web Service security operations (digital signature and encryption) from this service provider. This field is optional. Certain scenarios require a service provider, depending on the request/response policy for the operation. For more information, see "Testing Services with Web Service Security" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
	■ Username - The user name in Web Service security username tokens generated from the test service. This field is optional. This user name is only needed in some scenarios where the operation's request policy specifies an identity assertion.
	Do not confuse this field with the transport security context username field.
	This must be a valid user name and password in the local security realm. An invalid user name or invalid password causes a client-side error on the test service.
	■ Password - The password in Web Service security user name tokens generated from the test service.
Transport	The Transport section is collapsed by default. The fields and values depend on the test configuration.
Authentication	■ Username - The user name for setting the security context used by the test service when invoking the proxy service.
	If the proxy service routes the message to a business service that expects a SAML token, this is the identity that will be represented by the token. For more information, see "Using SAML with Oracle Service Bus" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.
	Do not confuse this field with the Web Service Security (WSS) username field. This must be a valid user name and password in the local security realm. An invalid user name or invalid password will cause a client-side error on the test service.
	Note: When the Test Console invokes a proxy with HTTP custom token authentication, the authentication check is not done.
	■ Password - The associated password for the Username.
Invocation Mode	Request/Response - This option is only displayed when testing any SOAP or any XML proxy service. Clear the **Request/Response** check box for one-way service invocations.
Message Metadata	See Table 33–10
Transport Headers	See Table 33–10

> **Note:**   The *secured* SOAP message is displayed printed with extra white spaces. Because white spaces can affect the semantics of the document, this SOAP message cannot always be used as the literal data. For example, digital signatures are white space-sensitive and can become invalid.

33.1.3 Viewing Proxy Services Test Results

Table 33–2 describes proxy service testing results. Tracing is only enabled if you have selected both the **Direct Call** and the **Include Tracing** options.

Table 33–2   Testing Results for Proxy Services

Section	Description
Proxy Service Name	The name of the proxy service that is being tested.

Table 33–2   (Cont.) Testing Results for Proxy Services

Section	Description
Test Console Actions	**Back** displays the previous browser page.
	Close closes the Test Console.
Request Document	The request message sent to the proxy service by the Test Console.
	This section is initially collapsed if the Test Console did not modify the request message. This section is initially expanded for SOAP services configured using the Form tab, or if WS-Security has been applied.
	If WS-Security has been applied, this section will contain two SOAP messages—the first message is the *clear text* message; the second is the *secured* SOAP message.
Response Document	The message response.
	For a SOAP service with a WS-Security response, this section contains two SOAP messages. The first SOAP message is the *secured* message as received by the client. The second SOAP message is the corresponding *clear text* message.
Response Metadata	The metadata retuned with the message response.
Section 33.1.4, "Tracing Proxy Services"	Tracing shows the state of the message as it passes through the system. When the **Direct Call** option is not selected, tracing is not performed. For more information on tracing, see Section 33.1.4, "Tracing Proxy Services"

33.1.4 Tracing Proxy Services

Tracing is enabled when you test proxy services using a direct call. The **Include Tracing** check box is selected by default with the **Direct Call** option. If you do not want tracing, clear the **Include Tracing** check box. When you enable tracing, the test results include the details of the trace. Use tracing to track problems in the system and to isolate them for correction. The trace information shows the path taken through the request and response paths in the code.

1. Click **Activate** under Change Center to enable the test feature in the Administration Console.

2. Select **Resource Browser** > **Proxy Services** to display the Summary of Proxy Services page.

3. Under Actions, click the **Launch Test Console** icon associated with the service you want to test. The Test Console opens the Proxy Service Testing page.

4. Configure the test data for the proxy service. You must have the **Direct Call** and **Include Tracing** options selected to enable tracing. See Section 33.1.2, "Configuring Proxy Services Test Data."

5. Click **Execute**. The Proxy Service Testing page displays the test results for the service and the tracing information.

6. Scroll down to the Invocation Trace section.

    This section displays a graphical representation the message flow. You can trace the message through the service and view the state of the message at pre-selected points in the trace. The trace points are automatically set.

7. Click the **+** plus sign to expand the message flow to view more detail.

    While viewing the trace you can also view the message flow in the Oracle Service Bus Administration Console. This helps you relate the trace to the stages and

actions in the message flow. You can modify the message flow and run the trace again to view the output.

33.1.5 Testing Business Services

You can test the following types of business services: any XML, any SOAP, Messaging, XML, and SOAP. You can test SOAP business services with Web Service Security policies. For more information, see "Testing Services with Web Service Security" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.

> **Note:** In a clustered domain, you cannot use the Test Console to test any configured business service or proxy service which routes to a business service.

1. Click **Activate** under Change Center to enable the test feature in the Administration Console.

2. Select **Resource Browser** > **Business Services** to display the Summary of Business Services page.

3. Under Actions, click the **Launch Test Console** icon associated with the service you want to test. The Test Console opens the Business Service Testing page. For example, using the tutorials provided with the product, click the icon associated with loanSaleProcessor.

4. For SOAP and XML services, select an operation from the **Available Operations** field.

5. Configure the test data for the business service (the input data should be the message being sent by the proxy service to the business service). The Direct Call and Include Tracing options that are available when testing proxy services are not available for business services. By default, business services are tested using the **Direct Call** option, meaning that the messages pass through the transport layer.

   To see message tracing, clear **Direct Call**.

   > **Note:** To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:
   >
   > - Minimum severity to log: Info
   >
   > - Log file: Info
   >
   > - Standard out: Info
   >
   > For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

6. Click **Execute**.

   The Business Service Testing page displays the results. For more information, see Section 33.1.7, "Viewing Business Services Test Results."

33.1.5.1 Testing Attachments in Business Services

This section describes how to use the Test Console to test attachments on business services.

For illustration purposes, this section assumes that a WSDL-based HTTP business service uses a "submitAttachment" operation to send a ZIP file as an attachment in a SOAP message. Modify the settings as appropriate for different types of binary attachments.

1. In the Test Console for the business service, go to the XML page and enter the following SOAP Envelope for the test message:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Header/>
    <env:Body>
       <m:submitAttachment xmlns:m="http://www.alsb.com/SOAPwithAttachment/">
          <submitAttachment>
             <fileName>c:\yourfile.zip</fileName>
          </submitAttachment>
          <zipFile href="cid:zipFile"/>
       </m:submitAttachment>
    </env:Body>
</env:Envelope>
```

2. In the Attachment section of the Test Console:

   a. **Content-Type**: application/zip (as defined in the WSDL)

   b. **Content-ID**: zipFile (must match the value of href="cid:zipFile" in the test message)

   c. **File**: Click **Browse**, and select the ZIP file you specified in the SOAP Envelope.

   d. Click **Add**.

3. Click **Execute** to send the test message with the attachment.

To confirm success of the sent attachment, check the server console for a message similar to the following, which is logged by the submitAttachment operation:

```
WS called - received the following properties:
submitAttachment is:
      com.alsb.soapwithattachment.SubmitAttachmentRequestType@e2abb0
zipFile is: javax.activation.DataHandler@175cf0a
```

33.1.6 Configuring Business Services Test Data

Table 33–3 lists the configuration options for testing business services. The fields for accepting input to the request document are based on the service type.

Table 33–3   Business Services Configuration Options

Section	Options/Fields
Name	The name of the business service being tested.
Available Operations	Operations associated with the business service.
Test Console Actions	■ Execute - Run the test. ■ Reset - Reset the input values. ■ Close - Close the window and do not run the test.

Table 33–3   (Cont.) Business Services Configuration Options

Section	Options/Fields
Request Document	The input fields generate the request message that is sent to the business service. Click **Execute** to run the test with the values you entered. The Test Console displays the request message and the service's response message.
	Input in the Request Document section is service-type specific. The service types and a description of the input required by each are listed in the following sections.

- any XML - The service requests input in the form of a payload. The payload is the content of the message being sent. The content is expected to be an XML document. You can browse to a file or enter the message content in the text box.

- any SOAP - The service requests input in the form of a payload. The payload is the content of the message being sent. The content is expected to be the SOAP envelope. You can browse to a file or enter the message content in the text box.

- Messaging - Messaging services define 4 possible input types: none, XML, Binary or Text. The service requests a single input either file-based or text-based, except for the type none that does no require any input.

  Oracle recommends entering binary input from a file. Data entered in the text area are converted to binary using the system encoding.

  Data entered from files for text services must be converted to text. The encoding input field is used to specify the encoding to apply during the conversion. The system encoding is used if this field is not configured. By default, the encoding field is initialized with the encoding value configured on the proxy service endpoint.

- XML - The service requests input in the form of a payload. The payload is the content of the message being sent. The content is expected to be an XML document. You can browse to a file or enter the message content in the text box.

  This is a WSDL-based service with multiple operations defined. Oracle provides a sample document to use in testing this service. All operations are listed at the top of the page with an arrow highlighting the selected operation.

- SOAP Document - For a SOAP document, the SOAP envelope is usually composed of zero or more headers and one body payload.

  The Form and XML tabs provide alternative ways to specify the content.

  The Form tab contains a SOAP Header field and a SOAP Body field. The content of the SOAP Header field is expected to be a SOAP Header tag (this allows for the definition of more than one header). The SOAP Body field contains the data that is actually sent as part of the message. The content is expected to be an XML document. Both the header and the body are used to generate the SOAP envelope.

  The XML tab contains a single input field. The content of this field is expected to be the SOAP envelope being sent.

- SOAP RPC - For SOAP RPC, the SOAP envelope is composed of zero or more headers, and zero or more arguments.

  The Form and XML tabs provide alternative ways to specify the content.

  The Form tab contains a single input for SOAP headers, and one input field for each argument (the name of the input field corresponds to the name of the argument). The content of the SOAP Header field is expected to be a SOAP Header tag (this allows for the definition of more than one header).

  The WSDL is used to detect the type of each argument. A single-line input field is used for primitive types; a multi-line input field is used for XML types. A sample document is automatically generated to facilitate testing.

  The headers and arguments are used by the Test Console to generate the SOAP envelope.

  The XML tab contains a single input field. The content of this field is expected to be the SOAP envelope being sent. The payload (XML input) can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test.

Table 33–3   (Cont.) Business Services Configuration Options

Section	Options/Fields
Web Service Security	This section is available only for SOAP services when the selected operation has a Web Service Security policy.
	■ Service Provider - The test service gets all client-side PKI (key-pair) credentials for Web Service security operations (digital signature and encryption) from this service provider. This field is optional. Certain scenarios require a service provider, depending on the request/response policy for the operation. For more information, see "Testing Services with Web Service Security" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
	■ Username - The user name in Web Service security username tokens generated from the test service. This field is optional. This user name is only needed in some scenarios where the operation's request policy specifies an identity assertion.
	Do not confuse this field with the transport security context username field. This must be a valid user name and password in the local security realm. An invalid user name or invalid password will cause a client-side error on the test service.
	In some scenarios, this user name and password may also be used when the test service generates a SAML assertion.
	■ Password - The password in Web Service security username tokens generated from the test service.
Transport	The Transport section is collapsed by default. The fields and values depend on the test configuration.
Authentication	■ Username - The user name for setting the security context used by the test service when invoking the business service. If the business service expects a SAML token, this identity may be propagated in the SAML token. See "Using SAML with Oracle Service Bus" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. This must be a valid user name and password in the local security realm. An invalid user name or invalid password will cause a client-side error on the test service.
	■ Password - The associated password. For more information, see Username.
	■ Service Key Provider - This field is used when testing an HTTPS business service with CLIENT-CERT authentication.The service provider must have an associated SSL client credential. The test service will use that credential during the SSL handshake.
Invocation Mode	Request\Response - This option is only displayed when testing an any SOAP or any XML business service. Clear the **Request/Response** check box for one-way service invocations.
Message Metadata	See Table 33–10.
Transport Headers	See Table 33–10.

33.1.7 Viewing Business Services Test Results

Table 33–4 describes business service testing results.

Table 33–4   Testing Results for Business Services

Section	Description
Business Service Name	The name of the business service.
Test Console Actions	Click **Back** to display the previous browser page.
	Click **Close** to close the Test Console window.

Table 33–4 (Cont.) Testing Results for Business Services

Section	Description
Request Document	The request message sent to the business service by the Test Console.
	This section is initially collapsed if the Test Console did not modify the request message. This section is initially expanded for SOAP services configured using the Form tab, or if WS-Security has been applied.
	If WS-Security has been applied, this section will contain two SOAP messages. The first message is the *clear text* message. The second is the *secured* SOAP message.
Response Document	The message response.
	For a SOAP service with a WS-Security response, this section will contain two SOAP messages. The first SOAP message is the *secured* message as received by the client. The second SOAP message is the corresponding *clear text* message.
Response Metadata	The metadata returned with the message response.

Note: The *secured* SOAP message is displayed pretty printed, for example, with extra white spaces. This SOAP message cannot always be used as the literal data as white spaces can affect the semantics of the document. For example, digital signatures are white space-sensitive and can become invalid.

33.2 Testing Transformations

You can test transformations after activating a session or during a session to ensure that the resources operate with the expected behavior. You must activate the session to test the runtime, otherwise the testing is done at design time using your local changes.

You can test the following transformations:

- Section 33.2.1, "Testing MFL Transformations"

- Section 33.2.2, "Testing XSLT Transformations"

- Section 33.2.3, "Testing XQuery Transformations"

33.2.1 Testing MFL Transformations

A Message Format Language (MFL) document is a specialized XML document used to describe the layout of binary data. MFL resources support two transformations. Each transformation only accepts one input and provides a single output: XML to binary, or binary to XML.

Table 33–5 describes MFL configuration options.

Table 33–5 Configuring MFL Options

Section	Description
Name	The name of the resource being tested.
Supported transformations	To select a specific transformation, select the transformation name.

Table 33–5   (Cont.) Configuring MFL Options

Section	Description
Test Console Actions	■ Execute - Apply the transformation. ■ Reset - Reset the input field (for XML to binary, the sample XML document is reset). ■ Close - Cancel the current operation.
MFL Transformation Configuration	■ XML Input - Required for XML to binary transformations: The XML schema for the MFL document can be inferred. A sample XML document is automatically entered in the text field. The XML input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test. ■ Binary Input - Required for binary to XML transformations: The binary input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test.

You can test the design time or the runtime.

1. Click **Activate** to test the runtime. Do not activate the session to test the design time.

2. Select **Resource Browser** > **MFLs** to display the Summary of MFL Files page.

3. Under Actions, click the **Launch Test Console** icon associated with the resource you want to test. The Test Console opens the Resource Testing page.

4. Select the transformation you want to test.

5. Configure the test data for the MFL resource. For more information, see Table 33–5.

6. Click **Execute**. The **Resource Testing** page displays the results.

7. To retest, click **Back**. You can close the Test Console, modify, and retest the resource.

33.2.2 Testing XSLT Transformations

eXtensible Stylesheet Language Transformations (XSLT) describe XML-to-XML mappings in Oracle Service Bus. To test an XSLT resource you must supply the input XML document and the Test Console returns the output XML document. XSLT transformations may have additional named parameters. All parameters required by the transformation are displayed on the configuration page. Default values are available but you can override them.

Table 33–6 describes XSLT configuration options.

Table 33–6   Configuring XSLT Options

Section	Description
Name	The name of the resource being tested.
Test Console Actions	■ Execute - Apply the transformation. ■ Reset - Reset the input field(s). ■ Close - Cancel the current operation.

Table 33–6   (Cont.) Configuring XSLT Options

Section	Description
Input and Parameters	The input document and parameters for testing the XSLT resource.
	■ XML Input - The XML input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test. XML input is required.
	■ <param_name> ([] as XML) - param_name is a named XSLT parameter.
	There are two types of input: XML and primitive (String, integer, float, and so on). The default input type is String. Select the check box associated with the parameter name to identify a parameter of type XML.

You can test the design time or the runtime.

1. Click **Activate** to test the runtime. Do not activate the session to test the design time.

2. Select **Project Explorer** > **XSLTs** to display the Summary of XSLTs page.

3. Under Actions, click the **Launch Test Console** icon associated with the resource you want to test. The Test Console opens the Resource Testing page.

4. Configure the test data for the resource in the Input and Parameters section of the page. For more information, see Table 33–6.

5. Click **Execute**. The Resource Testing page displays the results.

6. To retest, click **Back**. You can close the Test Console, modify, and retest the resource.

33.2.3 Testing XQuery Transformations

XQuery maps can describe XML-to-XML, XML to non-XML, and non-XML to XML mappings. An XQuery transformation can take multiple inputs and returns one output. Each input corresponds to an XQuery external variable declared in the XQuery resource. The value of an XQuery input variable can be a primitive value (String, integer, date, and so on), an XML document, or a sequence of these types. The output value can be a primitive value (String, integer, date, and so on), an XML document, or a sequence of these types.

> **Note:**   The Test Console does not support sequences on input.

Table 33–7 describes XQuery configuration options.

Table 33–7   Configuring XQuery Options

Section	Description
Name	The name of the resource being tested.
Test Console Actions	■ Execute - Execute the transformation.
	■ Reset - Reset the input field(s).
	■ Close - Close the Test Console.

Table 33–7   (Cont.) Configuring XQuery Options

Section	Description
Variables	This section contains one input field for each of the XQuery external variables.
	<param_name> ([] as XML) - `param_name` is a XQuery variable name in the XQuery resource.
	In the Test Console, a single-line edit box is displayed if the type is a simple type. A multi-line edit box is displayed if the data is XML.
	A combination input (`<param_name> ([] as XML)`) is used when the variable is not typed. You must declare the variable type. Select the check box to identify a parameter of type XML.
	An XML input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test.
	Input in the Test Console is rendered based on the type to make it easier to understand the type of data you must enter. When untyped, the default type is XML.

You can test the design time or the runtime.

1. Click **Activate** to test the runtime. Do not activate the session to test the design time.

2. Select **Project Explorer** > **XQueries** to display the Summary of XQueries page.

3. Under Actions, click the **Launch Test Console** icon associated with the resource you want to test. The Test Console opens the Resource Testing page.

4. Configure the test data for the resource in the Variables section of the page. For more information, see Table 33–7.

5. Click **Execute**. The Resource Testing page displays the results.

6. To retest, click **Back**. You can close the Test Console, modify, and retest the resource.

33.3 Performing XQuery Testing

You can edit and test an action in the message flow using the following Editors: XQuery Expression Editor, XQuery Condition Editor, and XPath Expression Editor. Testing takes the same form for both the XQuery Expression and Condition Editors. However, the scenario is different for the XPath Expression Editor because it takes only one input.

> **Note:**   You must disable the pop-up blockers in your browser for XQuery testing to work. If you have toolbars in the IE browser, you may need to disable the pop-up blockers from under the **Options** menu as well as for all toolbars that are configured to block them.

This section includes the following topics:

- Section 33.3.1, "Using the XQuery Expression and XQuery Condition Editors"
- Section 33.3.2, "Using the XPath Expression Editor"

33.3.1 Using the XQuery Expression and XQuery Condition Editors

You use XQuery expressions to create the data content for the message context variables (or part of a message context variable) during the execution of the message flow. You can use the Test Console directly in the XQuery Expression Editor to test the definition of the expression.

Similarly, you use XQuery conditions to evaluate Boolean conditions in the message flow. You can use the Test Console directly in the XQuery Condition Editor to test the definition of the condition.

An XQuery can take multiple inputs and returns one output. Each input corresponds to an XQuery unbound variable defined in the XQuery. The value of an XQuery input can be a primitive value (String, integer, date, and so on), an XML document, or a sequence of these types. The output value can be a primitive value (String, integer, date, and so on), an XML document, or a sequence of these types.

> **Note:** The Test Console does not support sequences on input.

Table 33–8 describes XQuery configuration options.

Table 33–8   Configuring XQuery Testing

Section	Description
Name	The type of expression being tested.
Test Console Actions	■ Execute - Apply the transformation. ■ Reset - Reset the input field(s). ■ Close - Cancel the current operation.
Variables	This section contains one input field for each of the XQuery unbound variables. <param_name> ([] as XML) - param_name is the name of the corresponding XQuery unbound variable. In the Test Console, a single-line edit box is displayed if the type is a simple type. A multi-line edit box is displayed if the data is XML. A combination input (<param_name> ([] as XML)) is used when the variable is not typed.You must declare the variable type. Select the check box to identify a parameter of type XML. An XML input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test. Input in the Test Console is rendered based on the type to make it easier to understand the type of data you must enter. The default type is XML.

1. Access the Test Console when editing an action in the message flow of a pipeline.

    ■ To access the XQuery Expression Editor, see Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

    ■ To access the XQuery Condition Editor, see Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

    The XQuery Testing Expression page is displayed. All input variables requested are displayed on the page.

2. Configure the test data for the XQuery in the Variables section of the page. For more information, see Table 33–8.

3. Click **Execute**. The testing page displays the results.

4. Once you have completed a test, you can click **Back** to execute a new test. To execute a new test after making changes to the XQuery, you must close and reopen the Test Console for the changes to take effect.

33.3.2 Using the XPath Expression Editor

You use XPath expressions to select a subset of an XML message context variable. You can use the Test Console in the XPath Expression Editor to test the definition of the XPath expression. An XPath expression takes a single XML document as input and generates a sequence of XML documents, primitives types, or both as output.

Table 33–9 describes XPath expression configuration options.

Table 33–9   Configuring XPath Options

Section	Description
Name	The type of expression being tested.
Test Console Actions	■ Execute - Apply the transformation. ■ Reset - Reset the input field. ■ Close - Cancel the current operation.
Variables	This section contains a single input field corresponding to the XML document against which this XPath expression is being tested. The XML input can be file-based or text-based. Referencing a file for input takes precedence over textual input. Browse and select the file you want to use in your test.

1. Access the Test Console when editing an action in the message flow of a pipeline. To access the XPath Expression Editor, see Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

2. Configure the test data for the XPath expression in the Variables section of the page. For more information, see Table 33–9.

3. Click **Execute**. The testing page displays the results.

4. Once you have completed a test, you can use the **Back** button to execute a new test. To execute a new test after making changes to the XPath expression, you must close and reopen the Test Console for the changes to take effect.

33.4 Understanding How the Runtime Uses the Transport Settings in the Test Console

Section 33.1.2, "Configuring Proxy Services Test Data" and Section 33.1.6, "Configuring Business Services Test Data" describe how you configure the values of the transport headers, transport metadata, and transport-related security data for outbound requests when you test proxy services or business services in the Test Console. However, some specifications you can make in the Test Console are not honored at runtime. That is, the values of certain headers or metadata are overwritten, or ignored by the Oracle Service Bus at runtime when the test is executed. The headers and metadata for which there are limitations are described in Table 33–10.

Table 33–10   Limitations to Transport Header and Metadata Values You Specify in the Test Console When Testing a Service

Transport	Service Type	Description of Limitation	Transport Headers Affected
HTTP(S)[1]	Proxy Services	All transport headers and other fields you set are preserved at runtime. This is true whether or not the **Direct Call** option is set.	All
HTTP(S)	Business Services	The Oracle Service Bus runtime overrides any values you set for these parameters.	Content-Length Content-Type relative-URI client-host client-address
JMS	Proxy Services	When the **Direct Call** option is used, all transport headers and other fields you set are preserved at runtime.	All
JMS	Proxy Services	When the **Direct Call** option is not used, the same limitations apply as those for a transport header action configuration.	See the limitations for JMS transport headers described in "Limitations to Transport Header Values You Specify in Transport Header Actions" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
JMS	Business Services	The same limitations apply as those for a transport header action configuration.	See the limitations for JMS transport headers described in "Limitations to Transport Header Values You Specify in Transport Header Actions" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Service Bus*.
Email	Proxy Services	No limitations. Any transport headers and other fields you set are honored at runtime. This is true whether or not **Direct Call** is specified.	None
Email	Business Services	The Oracle Service Bus runtime overrides any values you set for these parameters	Content-Type
File	Proxy Services	No limitations. Any transport headers and other fields you set are honored at runtime.[2]	None
File	Business Services	No limitations	None
FTP	Proxy Services	No limitations. Any transport headers and other fields you set are honored at runtime.	None
FTP	Business Services	No limitations	None

[1] When you test proxy services, the Test Console never sends a HTTP request over the network, therefore transport-level access control is not applied.

[2] For example, FileName (Transport metadata)—the value you assign is appended to the output file name. For example, 1698922710078805308-b3fc544.1073968e0ab.-7e8e-{$FileName}.

Part II

User Guide

This part provides high-level information about using Oracle Service Bus. Use this information in conjunction with specific procedural information in other Oracle Service Bus guides. Chapters include:

- Chapter 34, "Introduction to the User Guide"

- Chapter 36, "Configuring Proxy Services and Business Services"

- Chapter 37, "Modeling Message Flow in Oracle Service Bus"

- Chapter 38, "Improving Service Performance with Split-Join"

- Chapter 39, "Message Context"

- Chapter 40, "Using the Test Console"

- Chapter 41, "UDDI"

- Chapter 42, "Extensibility Using Java Callouts and POJOs"

- Chapter 43, "XQuery Implementation"

34

Introduction to the User Guide

This part provides information on using and configuring Oracle Service Bus. It is intended for those responsible for messaging and SOA, specifically enterprise architects, application architects, and developers.

Use this guide as a practical high-level guide for working with Oracle Service Bus. While sometimes providing procedural information, this guide does not provide detailed information on how to configure resources using the Oracle Service Bus Administration Console or the Eclipse plug-ins. For information on using the Oracle Service Bus Administration Console, see Part I, "Console Help for Oracle Service Bus". For information on using the Oracle Service Bus Eclipse plug-ins, see "IDE Help for Oracle Service Bus" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Information for operations specialists such as monitoring, reporting, and tracing is presented in Part III, "Operations Guide".

For information about Oracle Service Bus transport providers used in proxy and business services, see "Transports" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

For security implementation details, see "Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

For deployment details, see the *Oracle Fusion Middleware Deployment Guide for Oracle Service Bus*.

35

Starting Oracle Service Bus Servers

This chapter describes different ways to start Oracle Service Bus servers.

This chapter contains the following sections.

- Section 35.1, "Starting Derby in Development Environments"
- Section 35.2, "Starting the Oracle Service Bus Admin Server"
- Section 35.3, "Starting Oracle Service Bus Managed Servers"
- Section 35.4, "Verifying Server Startup"

For information on server startup not contained in this chapter, see "Starting and Stopping Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

35.1 Starting Derby in Development Environments

There are specific issues to account for when starting Oracle Service Bus development servers that use the default Apache Derby database. For more information, see "Running Multiple Development Servers with Derby" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

35.2 Starting the Oracle Service Bus Admin Server

This section describes different options for starting and Oracle Service Bus Admin server.

35.2.1 Starting the Admin Server from the Command Line

Invoke the `startWebLogic` command from the root of the Oracle Service Bus domain: `startWebLogic.cmd` or `startWebLogic.sh`

A command window is displayed showing status information about Oracle WebLogic Server.

If you started the server in production mode, you are required to enter a valid user name and password to start the server.

35.2.2 Starting the Admin Server on Windows

This section describes how to start the Oracle Service Bus Admin server on Windows systems.

35.2.2.1 Starting the Admin Server from the Windows Start Menu

From the Windows Start menu, select **Start** > **All Programs** > **Oracle WebLogic** > **User Projects** > *domain_name* > **Start Server for Oracle Service Bus Domain**.

The *domain_name* is the name you assigned your Oracle Service Bus domain when you created it.

MW_HOME/user_projects is the default domain location.

35.2.2.2 Starting the Admin Server as a Windows Service

This section describes how to configure your environment in order to start an Oracle Service Bus domain as a Windows service. Starting the Admin server as a Windows service ensures the server starts after a system restart.

Because this procedure requires a modification to your Windows registry, you should back up your Windows registry before proceeding.

> **Note:** You can also start an Oracle Service Bus Admin server as a Windows service through Node Manager, as described in "Starting an Administration Server Using WLST and Node Manager" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

1. Set your Oracle Service Bus domain to start as a Windows service by running the *WL_HOME*\server\bin\installSvc.cmd script, as described in "Setting Up a WebLogic Server Instance as a Windows Service" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

2. Add the required environment variables to the Windows registry. Before proceeding, you must know the name of the Windows service you created. The service name is typically "beasvc *[DOMAIN_NAME]_[SERVER_NAME]*."

   a. Back up your Windows registry.

   b. Run the Windows regedit utility. Choose **Start** > **Run** in Windows. Enter regedit in the Run dialog box, and click **OK**.

   c. Locate the following registry key for your Windows service:

      HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\beasvc *[DOMAIN_NAME]_[SERVER_NAME]*

   d. Create a new Multi-String Value called `Environment`.

   e. Edit the new Environment value by adding the following entry:

      `ALSB_HOME=C:\oracle11g\Oracle_OSB1`

      where C:\oracle11g is the root of your Oracle Fusion Middleware installation.

   f. **Windows XP 32-bit only**: In a command window, change directories to a location where you want to generate a temporary file, then run the following command: `set > env.txt`.

      Open the env.txt file, copy its entire contents, and paste the contents into the Environment registry value you created, below the ALSB_HOME entry.

   g. Click **OK** in the multi-string value dialog box and close the Windows registry editor.

3. Before starting the service, be sure to start any databases used by the Oracle Service Bus domain.

35.2.3 Starting Servers Using Node Manager

See "Starting an Administration Server Using WLST and Node Manager" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

35.3 Starting Oracle Service Bus Managed Servers

Start each Oracle Service Bus Managed Server from a unique command window using the following syntax:

DOMAIN_HOME/bin/startManagedWebLogic.sh(.cmd) *osb_server*
http://*administration_server_host*:*administration_server_port*
-Dweblogic.management.username=*user_name*
-Dweblogic.management.password=*password*

where

- *osb_server* is the name of the Managed Server.

- http://*administration_server_host*:*administration_server_port* is an optional argument when the Admin server runs on a non-default port or resides on a different host than your Managed Servers (in a distributed environment).

- The -Dweblogic.management arguments let you optionally provide the user name and password you entered in the Oracle Fusion Middleware Configuration Wizard when you created the domain.

  If you do not provide the login credentials as command arguments, Oracle Service Bus prompts you for the user name and password before starting the server.

If you do not know the names of the Managed Servers that need to be started, do one of the following:

- View the following file: *DOMAIN_HOME*/startManagedWebLogic_readme.txt

- Access the Oracle WebLogic Server Administration Console at the following URL:

  http://*host*:*admin_server_port*/console

  After you log in, navigate to **Environment** > **Servers** to see the names of your Managed Servers.

For more information, see "Starting and Stopping Servers" in *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server*.

35.4 Verifying Server Startup

When a server has successfully started, the following message is printed to the console window:

```
<Server started in RUNNING mode>
```

36

Configuring Proxy Services and Business Services

This chapter describes the high-level aspects of configuring Oracle Service Bus business and proxy services, including the structure and use of WSDLs, service types and transports, and business service result caching with Oracle Coherence.

Oracle Service Bus proxy services and business services provide the means for managing services, transforming messages, and routing messages through the enterprise.

You can create and configure proxy services and business services using either the runtime environment (Oracle Service Bus Administration Console) or the development environment (Eclipse with Oracle Service Bus plug-ins). You can base proxy services or business services on existing Web Services Description Language (WSDL) resources, including those imported from a UDDI registry such as the Oracle Service Registry, and then further configure them in the Oracle Service Bus Administration Console or the plug-in.

This chapter contains the following sections:

36.1 Oracle Service Bus Proxy Services

Oracle Service Bus proxy services are definitions of intermediary Web services that Oracle Service Bus implements and hosts locally. Oracle Service Bus uses proxy services to route messages between business services (such as enterprise Web services and databases) and service clients (such as presentation applications or other business services).

A proxy service configuration includes its interface, transport settings, security settings, and a message flow definition. The message flow definition defines the logic that determines how messages are handled as they flow through the proxy service. If a

proxy service is based on a WSDL document, the configuration also includes a WSDL port or a WSDL binding. (See Section 36.4, "Using a WSDL to Define a Service.")

36.2 Oracle Service Bus Business Services

Oracle Service Bus business services are definitions of enterprise Web services to which Oracle Service Bus is a client. Those external Web services are implemented in and hosted by external systems, so Oracle Service Bus must know what to invoke, how to invoke it, and what to expect as a result. Oracle Service Bus business services model those interfaces so that Oracle Service Bus can invoke the external services.

A business service configuration includes its interface, transport settings, and security settings. (It does not include a message flow definition.) If the business service is based on a WSDL, the configuration also includes a WSDL port or a WSDL binding. (See Section 36.4, "Using a WSDL to Define a Service.")

36.3 How WSDL is Used in Oracle Service Bus

Oracle Service Bus defines some types of business services and proxy services using a WSDL, an XML-based specification for describing Web services. A WSDL document describes service operations, input and output parameters, and how a client application connects to the service. For the WSDL 1.1 specification, see the W3C Note, "W3C Web Services Description Language (WSDL) 1.1," at http://www.w3.org/TR/wsdl.

36.3.1 About Effective WSDLs and Generated WSDLs

In Oracle Service Bus, you can base a new proxy service or a new business service on an existing WSDL (called a "WSDL resource") and then override or add configuration properties. You can also create and configure new services that are not based on existing WSDLs.

36.3.1.1 Effective WSDLs

For WSDL-based services, Oracle Service Bus uses *effective WSDLs* instead of actual .wsdl files. An *effective WSDL* represents a service's WSDL properties as configured in Oracle Service Bus.

A source WSDL serves as a template for an effective WSDL. When you create a service based on a WSDL, Oracle Service Bus generates an effective WSDL by combining settings from the original WSDL and any transport configurations you set, plus any other configuration settings you add or change from the original WSDL. Settings from the original WSDL that are not used in the new configuration are omitted from the effective WSDL.

Oracle Service Bus can generate effective WSDLs for these types of proxy services and business services:

■ SOAP services created from a WSDL

■ XML services created from a WSDL

Effective WSDLs can be generated for those types of services using any transport that supports WSDL-based services, including HTTP, JMS, SB, and so on.

Oracle Service Bus cannot generate effective WSDLs for these types of proxy services and business services:

■ Any SOAP

- Any XML

- Messaging type services

Effective WSDLs have different characteristics for proxy services and business services and for services based on WSDL ports and services based on WSDL bindings. Those characteristics are discussed throughout this documentation. In particular, see Section 36.5, "Basing Services on WSDL Ports and on WSDL Bindings."

36.3.1.2 Generated WSDLs

A *generated WSDL* is an effective WSDL that Oracle Service Bus generates for transport-type services that were not created from a WSDL resource but which can be described using a WSDL. For example, a WSDL generated from an EJB-based service is a generated WSDL.

36.3.1.3 Accessing Effective WSDLs

There are three ways to access an effective WSDL:

- In a Web browser, enter the URL for an HTTP-based proxy service, appended with `?WSDL`.

  This works only for HTTP-transport-based services for which Oracle Service Bus can generate effective WSDLs.)

- In a Web browser, enter the fixed HTTP URL, for example:

  http://host:port/sbresource?PROXY/*project_path*/*proxy_service_name*

  or

  http://host:port/sbresource?BIZ/*project_path*/*business_service_name*

  This works for all services for which Oracle Service Bus can generate effective WSDLs.

- Export the WSDL from the Administration Console or from the plug-in. See:

  – Section 4.28, "Exporting a WSDL"

  – "Generating an Effective WSDL" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

  Exporting the WSDL generates a .zip file that contains the effective WSDL along with associated dependencies, including schemas and WS-Policies. Oracle Service Bus evaluates the dependencies, and the appropriate location is added to the `location` attribute of the WSDL `import` element.

  There is no `import` element for WS-policies. For WS-policies, the policy reference is retained, and the policy resource is included in the export.

  You cannot export a generated WSDL.

  For more information on accessing Oracle Service Bus resources with a URL, see Section 4.30, "Viewing Resources in a Web Browser."

36.3.2 WSDL Overview

A WSDL document describes a service, its location, its operations, and the way in which clients can communicate with it. This section provides a very brief introduction to WSDL, to provide context for discussing Oracle Service Bus features.

Table 36–1 summarizes the main elements used to define WSDL services.

Table 36–1   High-level WSDL Elements

Element	Description
<types>	Type definitions for message content.
<message>	Abstract definition of the data being exchanged. A message consists of parts, which describe the logical, abstract content of the message.
<portType>	Abstract collection of operations supported by the service.
<operation>	Abstract description of an action supported by the service.
<binding>	Concrete protocol and data format specification for a port type.
<port>	A single endpoint, consisting of a network address and a binding.
<service>	Collection of related ports, or endpoints.

WSDL specifies SOAP, HTTP, MIME, and Oracle Service Bus-specific binding extensions, which extend the WSDL binding mechanism to support protocol-specific or message format-specific features.

36.3.2.1 Types

The `<types>` element is a container for data type definitions. It uses a type system, such as XML Schema (XSD), to define the vocabulary of messages handled by this service. For example, a service that provides stock quotes might define an XML vocabulary, with the terms `TradePriceRequest` and `TradePrice`, as shown in Example 36–1.

Example 36–1   WSDL Types Example

```
<types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
            xmlns="http://www.w3.org/2001/XMLSchema">
        <element name="TradePriceRequest">
            <complexType>
                <all>
                    <element name="tickerSymbol" type="string"/>
                </all>
            </complexType>
        </element>
        <element name="TradePrice">
            <complexType>
                <all>
                    <element name="price" type="float"/>
                </all>
            </complexType>
        </element>
    </schema>
</types>
```

36.3.2.2 Message

The `<message>` element provides an abstract, typed definition of the data being communicated. A message consists of parts, each of which describes one logical, abstract unit of the message. A WSDL can define one or more messages, each of which may have one or more parts. For example, the WSDL fragment in Example 36–2 defines four message types, `sellerInfoMessage`, `buyerInfoMessage`, `response`, and `negotiationMessage`, each of which has one or more parts.

Example 36–2   WSDL Message Example

```
<message name="sellerInfoMessage">
    <part name="inventoryItem" type="xsd:string"/>
    <part name="askingPrice" type="xsd:integer"/>
</message>

<message name="buyerInfoMessage">
    <part name="item" type="xsd:string"/>
    <part name="offer" type="xsd:integer"/>
</message>

<message name="response">
    <part name="result" type="xsd:string"/>
</message>

<message name="negotiationMessage">
    <part name="item" type="xsd:string"/>
    <part name="price" type="xsd:integer"/>
    <part name="outcome" type="xsd:string"/>
</message>
```

36.3.2.3 Port Type

The `<portType>` element defines a set of operations supported by one or more endpoints (which are defined in the `<port>` element). The port type provides the public interface for the operations provided by the service. Each operation is defined in an `<operation>` element, each of which is an abstract description of an action supported by the service.

For example, the fragment in Example 36–3 defines a port type with one operation, `GetLastTradePrice`, which can handle an input message, `GetLastTradePriceInput`, and an output message, `GetLastTradePriceOuput`. The concrete descriptions of these messages are defined in the WSDL binding, as shown in the `<soap:operation>` subelement in Example 36–4.

Example 36–3   WSDL Port Type and Operation Example

```
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
```

36.3.2.4 Binding

The `<binding>` element specifies a concrete data format specification and a concrete transport protocol for a port type.

The fragment in Example 36–4 specifies the binding for the `StockQuotePortType` port type, which is provided as the value for the `type` attribute. The `<soap:binding>` subelement signifies that the binding is bound to the SOAP protocol format. In that subelement, the `style` attribute specifies that the data format is SOAP document style, and the `transport` attribute specifies that the transport protocol is HTTP.

Example 36–4   WSDL Binding Example

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document"
          transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
            <operation name="GetLastTradePrice">
                <soap:operation
                    soapAction="http://example.com/GetLastTradePrice"/>
                    <input>
                        <soap:body use="literal"/>
                    </input>
                    <output>
                        <soap:body use="literal"/>
                    </output>
            </operation>
    </binding>
```

36.3.2.5 Service and Port

The `<service>` element defines a collection of related endpoints, each of which is defined in a child `<port>` element. A port is defined as a binding associated with a network address. For example, the fragment shown in Example 36–5 defines two ports, `StockQuotePort`, and `StockQuotePortUK`. They both use the same binding, `tns:StockQuoteSoapBinding`, (which is concretely defined in `<binding>`) but have different network addresses: `http://example.com/stockquote` vs. `http://example.uk/stockquote`. These are alternative ports available for this service.

Example 36–5   WSDL service and port Example

```
<service name="StockQuoteService">
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.com:9999/stockquote"/>
    </port>
    <port name="StockQuotePortUK" binding="tns:StockQuoteSoapBinding">
        <soap:address location="http://example.uk:9999/stockquote"/>
    </port>
</service>
```

36.4 Using a WSDL to Define a Service

If a service has a well defined WSDL interface, it is recommended, although not required, that you use the WSDL to define the service.

There are three types of WSDLs you can define. They are:

- Section 36.4.1, "SOAP Document Wrapped Web Services"

- Section 36.4.2, "SOAP Document Style Web Services"

- Section 36.4.3, "SOAP RPC Web Services"

36.4.1 SOAP Document Wrapped Web Services

A document wrapped Web service is described in a WSDL as a Document Style Service. However, it follows some additional conventions. Standard document-oriented Web service operations take only one parameter or message part, typically an XML document. This means that the methods that implement the operations must also have only one parameter. Document-wrapped Web service operations, however, can take any number of parameters, although the parameter values will be wrapped into one complex data type in a SOAP message. This wrapped complex data type will be described in the WSDL as the single document for the operation.

36.4.2 SOAP Document Style Web Services

You can configure proxy services as SOAP style proxy services and configure business services as SOAP style business services.

Example 36–6 provides an example of a WSDL for a sample document style Web service using SOAP 1.1.

Example 36–6   WSDL for a Sample Document Style Web Service

```
<definitions name="Lookup"
targetNamespace="http://example.com/lookup/service/defs"
xmlns:tns="http://example.com/lookup/service/defs"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:docs="http://example.com/lookup/docs"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xs:schema targetNamespace="http://example.com/lookup/docs" elementFormDefault="qualified">
      <xs:element name="PurchaseOrg" type="xs:string"/>
      <xs:element name="LegacyBoolean" type="xs:boolean"/>
    </xs:schema>
  </types>
  <message name="lookupReq">
    <part name="request" element="docs:purchaseorg"/>
  </message>
  <message name="lookupResp">
    <part name="result" element="docs:legacyboolean"/>
  </message>
  <portType name="LookupPortType">
    <operation name="lookup">
      <input message="tns:lookupReq"/>
      <output message="tns:lookupResp"/>
    </operation>
  </portType>
  <binding name="LookupBinding" type="tns:lookupPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="lookup">
      <soap:operation/>
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
</definitions>
```

The service has an operation (equivalent to a method in a Java class) called `lookup`. The binding indicates that this is a SOAP 1.1 document style Web service.

When the WSDL shown in the preceding listing is used for a request, the value of the body variable (`$body`) that the document style proxy service obtains is displayed in Example 36–7.

> **Note:**   Namespace declarations have been removed from the XML in the listings that follow for the sake of clarity.

Example 36–7   Body Variable Value

```
<soap-env:body>
  <req:purchaseorg>Oracle</req:purchaseorg>
</soap-env:body>
```

In Example 36–7, soap-env is the predefined SOAP 1.1 namespace and req is the namespace of the PurchaseOrg element (http://example.com/lookup/docs).

If the business service to which the proxy service is routing uses the above WSDL, the value for the body variable ($body) given above is the value of the body variable ($body) from the proxy service.

The value of the body variable ($body) for the response from the invoked business service that the proxy service receives is displayed in Example 36–8.

> **Note:**   Namespace declarations have been removed from the XML in the listings that follow for the sake of clarity.

Example 36–8   Body Variable Value

```
<soap-env:body>
  <req:legacyboolean>true</req:legacyboolean>
</soap-env:body>
```

This is also the value of the body variable ($body) for the response returned by the proxy service using this WSDL.

There are many tools available (including Eclipse with Oracle Service Bus plug-ins) that take the WSDL of a proxy service (obtained by adding the ?WSDL suffix to the URL of the proxy service in the browser) and generate a Java class with the appropriate request and response parameters to invoke the operations of the service. This Java class can be used to invoke the proxy service that uses this WSDL.

36.4.3 SOAP RPC Web Services

You can configure proxy services as RPC style proxy services and configure business services as RPC style business services.

Example 36–9 provides an example of a WSDL for a sample RPC style Web service using SOAP 1.1.

Example 36–9   WSDL for a Sample RPC Style Web Service

```
<definitions name="Lookup"
targetNamespace="http://example.com/lookup/service/defs"
xmlns:tns="http://example.com/lookup/service/defs"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:docs="http://example.com/lookup/docs"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xs:schema targetNamespace="http://example.com/lookup/docs" elementFormDefault="qualified">
      <xs:complexType name="RequestDoc">
        <xs:sequence>
          <xs:element name="PurchaseOrg" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ResponseDoc">
        <xs:sequence>
```

```
            <xs:element name="LegacyBoolean" type="xs:boolean"/>
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
  </types>
  <message name="lookupReq">
    <part name="request" type="docs: RequestDoc"/>
  </message>
  <message name="lookupResp">
    <part name="result" type="docs: ResponseDoc"/>
  </message>
  <portType name="LookupPortType">
    <operation name="lookup">
      <input message="tns:lookupReq"/>
      <output message="tns:lookupResp"/>
    </operation>
  </portType>
  <binding name="LookupBinding" type="tns:lookupPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="lookup">
      <soap:operation/>
      <input>
        <soap:body use="literal" namespace="http://example.com/lookup/service"/>
      </input>
      <output>
        <soap:body use="literal" namespace="http://example.com/lookup/service"/>
      </output>
    </operation>
  </binding>
</definitions>
```

The service described in the preceding listing includes an operation (equivalent to a method in a Java class) called `lookup`. The binding indicates that this is a SOAP RPC Web service. In other words, the Web service's operation receives a set of request parameters and returns a set of response parameters. The `lookup` operation has a parameter called `request` and a return parameter called `result`. The namespace of the operation in the binding is:

```
http://example.com/lookup/service/defs
```

When the WSDL shown in Example 36–9 is used for a request, the value of the body variable (`$body`) that the SOAP RPC proxy service obtains is displayed in Example 36–10.

> **Note:** Namespace declarations have been removed from the XML in the listings that follow for the sake of clarity.

Example 36–10   Body Variable Value

```
<soap-env:body>
  <ns:lookup>
    <request>
      <req:purchaseorg>Oracle</req:purchaseorg>
    </request>
  </ns:lookup>
<soap-env:body>
```

Where `soap-env` is the predefined SOAP 1.1 name space, `ns` is the operation namespace (`<http://example.com/lookup/service>`) and, `req` is the namespace of the `PurchaseOrg` element (`http://example.com/lookup/docs`).

If the business service to which the proxy service routes the messages uses the WSDL shown in Example 36–10, the value for the body variable (`$body`), shown in Example 36–11, is the value of the body variable (`$body`) from the proxy service.

When this WSDL is used for a request, the value of the body variable (`$body`) for the response from the invoked business service that the proxy service receives is displayed in Example 36–10.

Example 36–11   Body Variable Value

```
<soap-env:body>
  <ns:lookupResponse>
    <result>
      <req:legacyboolean>true</req:legacyboolean>
    </result>
  </ns:lookupResponse>
<soap-env:body>
```

This is also the value of the body variable (`$body`) for the response returned by the proxy service using this WSDL.

There are many tools available (including Eclipse with Oracle Service Bus plug-ins) that take the WSDL of a proxy service (obtained by adding the `?WSDL` suffix to the URL of the proxy in the browser) and generate a Java class with the appropriate request and response parameters to invoke the operations of that service. You can use such Java classes to invoke the proxy services that use this WSDL.

The benefits of using a WSDL include the following:

- The system can provide metrics for each operation in a WSDL.

- Operational branching is possible in the pipeline. For more information, see Section 37.2, "Branching in Message Flows."

- For SOAP 1.1 services, the `SOAPAction` header is automatically populated for services invoked by a proxy service. For SOAP 1.2 services, the `action` parameter of the `Content-Type` header is automatically populated for services invoked by a proxy service.

- A WSDL is required for services using WS-Security. WS-Policies are attached to WSDLs.

- The system supports the `<url>?WSDL` syntax, which lets you dynamically obtain the WSDL of a HTTP proxy service. This is useful for a number of SOAP client generation tools, including Eclipse with Oracle Service Bus plug-ins. For more information, see Section 4.30, "Viewing Resources in a Web Browser."

- In the XQuery and XPath editors and condition builders, it is easy to manipulate the body content variable (`$body`) because the editor provides a default mapping of `$body` to the request message in the WSDL of a proxy service. See Chapter 39, "Message Context."

- The runtime contents of `$body` for a specific action can be different from the default mapping displayed in the editor. This is because Oracle Service Bus is not a programming language in which typed variables are declared and used. Instead, variables are untyped and are created dynamically at runtime when a value is assigned. In addition, the type of the variable is the type that is implied by its contents at any point in the message flow. To enable you to easily create XQuery

and XPath expressions, the design time editor lets you map the type for a given variable by mapping the variable to the type in the editor. To learn about using the XQuery and XPath editor to create expressions, see Section 37.11, "Working with Variable Structures."

36.5 Basing Services on WSDL Ports and on WSDL Bindings

When you create a service based on a WSDL resource, you must base the service on a WSDL port or on a WSDL binding:

- When you create a new service based on a *binding* in a WSDL resource, you are choosing the protocol and data format defined in the selected `<binding>` element in the WSDL resource.

- When you create a new service based on a *port* in a WSDL resource, you are choosing the binding and the network address defined in the `<port>` element.

When creating or modifying the service, you can change the transport, but you cannot override the data format.

The port and binding definitions from the original WSDL resource are modified in the effective WSDL depending on a number of factors, as described below.

36.5.1 Characteristics of Effective WSDLs Generated for Proxy Services

The following characteristics apply to effective WSDLs generated for proxy services:

- The effective WSDL has one and only one `wsdl:service` section.

- The `wsdl:service` section has one and only one `wsdl:port` section.

- For SOAP services, any existing `<wsdl:service>` definition is removed, and a new service definition containing a single `<wsdl:port>` is created.

- For SOAP binding over any of the supported transports the `wsdl:binding` section contains the standard WSDL SOAP binding elements along with a unique transport URI that identifies the transport.

- For XML binding over HTTP, the `wsdl:binding` section uses the standard binding elements specified in the WSDL 1.1 specification.

- For XML binding over any of the other supported transports the `wsdl:binding` section uses Oracle (Oracle Service Bus) proprietary WSDL XML binding elements.

- If the service is based on a binding:

  - If the service is generated from binding Y in the WSDL resource, the effective WSDL defines a new service and port (`<bindingname>QSService` and `<bindingname>QSPort`). None of the ports defined in the WSDL resource are included in the effective WSDL.

  - There may be multiple ports in that WSDL associated with that binding. Each port can use a different URL. Therefore, the effective WSDL uses the binding but generates an artificial port from the configuration on the service for that binding. All other ports will be removed.

- If the service is based on a port:

  - If the service is generated from port X in the WSDL resource, then port X is also defined in the effective WSDL. Any other ports defined in the WSDL resource are not included. Furthermore, if you base the proxy service on a

WSDL port, the effective WSDL uses that port name. The binding is determined from the port, and in turn, the port type is determined from the binding.

- The effective WSDL preserves any WS-Policies associated with the port defined in the resource WSDL.

- The transport address specified in the port definition in the resource WSDL is never used as the address for a proxy service in the effective WSDL:

  For HTTP services, you must specify a transport address when configuring the transport in the Administration Console or the plug-in. That address is used in the port definition in the effective WSDL.

  The URL specified as the transport address for a proxy service is always relative to a path in an Oracle Service Bus domain, because Oracle Service Bus always hosts proxy services.

- For SOAP-protocol-based WSDL services, the transport URI in the SOAP binding depends on the transport implementation. For standard transports (like HTTP and JMS), this value is as per the SOAP specification or another universally accepted value. For transports for which SOAP does not define a standard value, Oracle Service Bus sets one consisting of a predefined namespace with the transport ID appended at the end: `http://www.oracle.com/transport/2007/05/`.

- There is one service element in the effective WSDL, and the port address contains a URL whose syntax and semantic is defined by the transport selected in the binding.

36.5.1.1 Characteristics of Effective WSDLs Generated for Non-Transport-Type Business Services

The following characteristics apply to effective WSDLs generated for non-transport-type business services:

- The effective WSDL has one and only one `wsdl:service` section.

- The `wsdl:service section` may have more than one `wsdl:port` sections. This is generally true when load balancing is used and there are multiple end point addresses that can be used using one of the load-balance algorithms supported.

- For SOAP binding over any of the supported transports, the `wsdl:binding` section contains the standard WSDL SOAP binding elements along with a unique transport URI that identifies the transport.

- For XML binding over any of the supported transports, the `wsdl:binding` section contains the Oracle WSDL XML binding elements.

- The URL specified as the transport address is a remote location where the remote service is hosted.

- If the service is based on a port:

  - The effective WSDL defines a port with the same name as the port in the template on which the new service is based. If multiple endpoint addresses are configured for the business service, the effective WSDL generated from it defines ports for all the endpoints, with the names `<portname_in_template>_1, <portname_in_template>_2,...`

  - The binding for the new service is determined from the port, and the port type is in turn determined from the binding.

- The transport address URL is a remote location where the remote service is hosted.

- If the service is based on a binding:

  - The effective WSDL defines a new service and port, based on the port in the WSDL resource. None of the ports defined in the WSDL resource are included in the effective WSDL.

  - A binding in the WSDL resource may be associated with multiple ports. Each port can use a different URL and have a different WS-Policy attached to it. Therefore, the generated WSDL uses the binding but generates an artificial port for that binding with no WS-Policy.

  - For XML-based WSDL services, standard HTTP binding is used only if the service uses HTTP transport. For all other services created from an XML-based WSDL, the effective WSDL uses Oracle binding.

36.5.1.2 Characteristics of Effective WSDLs Generated for Transport-Type Business Services

Oracle Service Bus does not guarantee one and only one `wsdl:service` section in effective WSDLs generated for transport-type business services. Because the WSDL is generated by the transport, Oracle Service Bus does not generate nor does it clean up extra service-port sections. Oracle Service Bus does, however, evaluate dependencies and sets their references during export.

36.5.1.3 Generating Effective WSDLs in Clustered Domains

When generating the effective WSDL in a clustered domain, Oracle Service Bus rewrites the endpoint URL based on the server or cluster configuration. If a front-end HTTP host/port (or a front-end HTTPS host/port for HTTPS endpoints) has been specified, it will be used; otherwise, the Managed Server host or port will be used. It is strongly advised that a front-end HTTP or HTTPS host/port is assigned in clustered domains.

36.5.1.4 Examples of Proxy Services Based on a Port and on a Binding

Example 36–12 shows fragments of port and binding definitions in a WSDL resource.

Example 36–12   WSDL resource

```
<portType name="StockQuotePortType">
...
</portType>
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
...
</binding>
<service name="StockQuoteService">
   <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
     <soap:address location="http://example.com:9999/stockquote"/>
   </port>
   <port name="StockQuotePortUK" binding="tns:StockQuoteSoapBinding">
      <soap:address location="http://example.uk:9999/stockquote"/>
   </port>
</service>
```

36.5.1.4.1 Basing the Service on a Port If you create a proxy service based on the
`StockQuotePort` port in Example 36–12, the effective WSDL will look something like
the fragment in Example 36–13.

Example 36–13   Effective WSDL for a Proxy Service Based on a Port

```
<binding name="StockQuoteSoapBinding" type="ns:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
...
</binding>
<service name="StockQuoteService">
  <port name="StockQuotePort" binding="ns:StockQuoteSoapBinding">
    <soap:address location="http://host:port/project/proxyname"/>
  </port>
</service>
```

Notice the following about the above example:

- The service name, **StockQuoteService**, is the same in both the template and the
  effective WSDL.

- The binding is the same in both the template and the effective WSDL. In this
  example, it specifies that the service will use the HTTP transport protocol for
  SOAP document style messages. (The binding also specifies the same binding
  operation in both the template and the effective WSDL, but that is not shown in
  this example.)

- The second port defined in the WSDL resource, `StockQuotePortUK`, is not defined
  in the effective WSDL.

- The transport address (URI) defined in the WSDL resource's port,
  `http://example.com:9999/stockquote`, is different from the address generated in
  the effective WSDL's port, http://host:port/*project_path/proxy_service_name*. The
  effective WSDL uses the address specified for the transport configuration in the
  Oracle Service Bus Administration Console or Eclipse with Oracle Service Bus
  plug-ins.

36.5.1.4.2 Basing the Service on a Binding If you create a proxy service based on the
`StockQuoteBinding` binding in Example 36–12, the effective WSDL will look
something like the fragment in Example 36–14.

Example 36–14   Effective WSDL for a Proxy Service Based on a Binding

```
<binding name="StockQuoteSoapBinding" type="ns:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
...
</binding>
<wsdl:service name="StockQuoteSoapBindingQSService"
  <wsdl:port name="StockQuoteSoapBindingQSPort"
            binding="ns:StockQuoteSoapBinding">
    <soap:address location="http:/host:port/project/proxyname"/>
  </wsdl:port>
</wsdl:service>
```

Notice the following about the above example:

- The service and the port in the WSDL resource are different from the service and
  the port generated in the effective WSDL.

- The service name in the WSDL resource and the effective WSDL are different: **StockQuoteService** in the template and **StockQuoteSoapBindingQSService** in the effective WSDL.

- The binding is the same in both the WSDL resource and effective WSDL. In this example, it specifies that the service will use the HTTP transport protocol for SOAP document style messages.

- The binding also specifies the same binding operation in both the template and the effective WSDL, but that is not shown in this example.

- The transport address (URI) defined in the WSDL resource's port, `http://example.com:9999/stockquote`, is different from the address generated in the effective WSDL's port, `http://host:port/project/stockquote`.

36.5.2 Using Any SOAP or Any XML Service Types

To expose one port to clients for a variety of enterprise applications, use **Any SOAP** or **Any XML** service types. For Any SOAP, you must specify if it is SOAP 1.1 or SOAP 1.2.

36.5.3 Using the Messaging Service Type

If one of the request or response messages is non-XML, you must use the messaging service type.

Oracle Service Bus does not automatically perform "misunderstand" SOAP header checking. However, you can use XQuery conditional expressions and validate actions to explicitly perform this type of check. For more information on the validate action, see Section 21.24, "Adding Validate Actions" or "Validate Action Properties" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. For more information on conditional XQuery expressions, see Chapter 23, "Proxy Services: XQuery and XPath Editors" or "Condition Editor" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

You can use Oracle Service Bus to configure a validate action and use XQuery conditional expressions to perform validation checks explicitly in the message flow.

For more information on service types, see "General Configuration page" in Section 20.1, "Creating and Configuring Proxy Services."

36.6 Configuring Proxy Services

The following sections provide an overview of proxy service configuration. For specific instructions for configuring proxy services, see:

- Chapter 20, "Proxy Services: Creating and Managing"

- "Working with Proxy Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

36.6.1 Proxy Service Types and Transports

Each proxy service is of a certain type, and each can be used with the transport protocols appropriate for that type. Oracle Service Bus supports several standard transports plus custom transports. The service types are described in Table 36–2, including the standard transports available for each type.

Table 36–2   Proxy Service Types and Transports Supported by Oracle Service Bus

Service Type	Description	Standard Transport Protocols Supported for this Service Type
WSDL Web service	A SOAP or XML proxy service whose interface is described by a WSDL document.	HTTP JCA JMS Local SB WS
Transport Typed Service (no WSDL)	A service that uses a JEJB transport.	JEJB
Messaging service (no WSDL)	A messaging service where the request message and the response message can be of different data types, including binary, text, MFL, and XML.	EMAIL FILE FTP HTTP JMS LOCAL MQ SFTP TUXEDO
Any SOAP service (no WSDL)	A SOAP service that does not have an explicitly defined, concrete interface.	HTTP JMS LOCAL SB
Any XML service (non-SOAP, no WSDL)	An XML service that does not have an explicitly defined, concrete interface.	EMAIL FILE FTP HTTP JMS LOCAL MQ SB SFTP TUXEDO

> **Note:**   All service types can send and receive attachments using MIME.

36.6.2 Transport and Security Configuration for Proxy Services

You must configure transport settings for all proxy service types in the Oracle Service Bus Administration Console or Eclipse with the Oracle Service Bus plug-ins. Configuration details vary, depending on the transport type. For specific configuration settings, see Section 20.2.3, "Transport Configuration Page" and Section 20.2.4,

"Protocol-Specific Transport Configuration Pages."

The transport you select should support the transport mode (request/response, one-way, or both) required by the binding definition, and it should be configured accordingly.

For services exchanging messages in both modes, you must configure the binding layer to choose the transport mode accordingly (for any transport implementing the request/response as two asynchronous calls, for example, JMS). This occurs automatically when the service is a concrete type, as it is described in the binding definition. When it is not a concrete type, to configure the binding layer, you must set the mode in the $outbound variable.

Based on the transport and WSDL, the transport mode is automatically selected, but you can overwrite it in $inbound or $outbound.

You can specify security for proxy services, using service providers. A service provider is required if the proxy service routes messages to HTTPS services that require client-certificate authentication and may be required in some message-level security scenarios.

A service account can be created to provide authentication when connecting to a business service. It acts as an alias resource for the required username and password pair.

Oracle WebLogic Server can be used to directly manage security credentials for a business service requiring credential-level validation.

For more information, see Chapter 18, "Service Key Providers." See also Section 36.6.5, "Security-Related Validation for Proxy Services."

36.6.3 Configuration Settings For Each Proxy Service Type

Each proxy service type is modeled following the same pattern. Each service type must define these configurations:

- Binding definition

- Runtime configuration

- Runtime variables ($operation, $body, $header, $attachments)

Configuration properties specific to individual proxy service types are described in Table 36–3:

Table 36–3   Configuration Properties for Different Proxy Service Types

Proxy Service Type	Configuration Properties
WSDL Web service	**Binding Definition:** See Section 36.4, "Using a WSDL to Define a Service."
	Runtime Variables:
	■ For SOAP-based WSDL services, the variables are similar to Any SOAP service except that $operation is initialized based on the Operation Selection algorithm (See Section 20.2.17, "Operation Selection Configuration Page.".
	■ For XML-based WSDL services, the variables are similar to Any XML service except that the $operation will be initialized based on the Operation Selection algorithm.

Table 36–3   (Cont.) Configuration Properties for Different Proxy Service Types

Proxy Service Type	Configuration Properties
Transport Typed	Transport typed services have an empty binding definition. A WSDL is not specified.
	The Transport Typed Service is an inbound transport that lets EJBs make invocations through Oracle Service Bus. It is a self-described transport that generates a WSDL to describe its service interface. The JEJB transport features transaction and security context propagation.
Messaging Service	**Binding Definition**: The binding definition for messaging services consists of configuring the content-type of the messages that are exchanged. The content-type for the response does not need to be the same as for the request; therefore, the response is configured separately (for example, the service could accept an MFL message and return an XML acknowledgment receipt) and could also be set to None.
	As per their definition, messaging-based services do not have any WSDL definition. It is not possible to request a WSDL document for those services.
	Following are the content types to choose from for the request and response:
	■ None
	■ Binary
	■ Text
	■ MFL
	■ XML
	■ Java
	Note: If you are using a Reply action in the pipeline to propagate success or failure messages from a service to the calling client, select an option other than None. The None option blocks the reply.
	Runtime Variables:
	This service type is message-based. There is no concept of multiple "operations" as for Web services. Therefore, the $operation variable is left empty.
	The $body variable holds the incoming message wrapped in a <soap:Body> element. (For SOAP 1.1 proxies, $body uses the SOAP 1.1 namespace Body; for SOAP 1.2 proxies, it uses SOAP 1.2 namespace Body.)
	The $header variable is not applicable to this service type, and is set to its default value.
	The $attachments variable contains message attachments if there are any.
	To learn more about the message context variables, see Section 39.3, "Message-Related Variables" and Section 39.9, "Constructing Messages to Dispatch."

Table 36–3   (Cont.) Configuration Properties for Different Proxy Service Types

Proxy Service Type	Configuration Properties
Any SOAP service	**Binding Definition**: The only information this service type defines is that the service is receiving or sending SOAP messages—regardless of their WSDL binding definition. Therefore the binding configuration for this type is empty. In addition, as there is no binding configuration, the combination of this type and the content-type of the message is sufficient to determine whether there are attachments to the message. As per their definition, "any" services (SOAP or XML) do not have any WSDL definition. It is not possible to generate or view a WSDL document for those services. **Runtime Variables**: The $body and $header variables respectively hold the `<soap:Body>` and `<soap:Header>` of the incoming SOAP message. (For SOAP 1.1 proxies, $body and $header use SOAP 1.1 namespace Body and namespace; for SOAP 1.2 proxies, they use SOAP 1.2 namespace Body and namespace.) The $attachments variable contains the SOAP message attachments, if any. The $operation variable is not applicable to this service type as you do not define a port type. To learn more about the message context variables, see Section 39.3, "Message-Related Variables" and Section 39.9, "Constructing Messages to Dispatch."
Any XML service	**Binding Definition**: The only information this service type defines is that the service is receiving/sending XML messages—regardless of their WSDL binding definition. Therefore, the binding configuration for this type is empty. In addition, as there is no binding configuration, the combination of this type and the content-type of the message is sufficient to determine whether there are attachments to the message. As per their definition, "any" services (SOAP or XML) do not have any WSDL definition. It is not possible to request a WSDL document for those services. **Runtime Variables**: The $body variable holds the incoming XML message wrapped in a `<soap:Body>` element. (For SOAP 1.1 proxies, $body and $header use SOAP 1.1 namespace Body and namespace; for SOAP 1.2 proxies, they use SOAP 1.2 namespace Body and namespace.) The $attachments variable contains message attachments, if there are any. The $header variable is not applicable to this service type and is set to its default value. The $operation variable is not applicable to this service type as you do not define a port type. To learn more about the message context variables, see Section 39.3, "Message-Related Variables" and Section 39.9, "Constructing Messages to Dispatch."

36.6.4 Configuring Message Flow

A proxy service's message flow definition defines the logic that determines how messages are handled as they flow through the proxy service. A message flow definition transforms messages, as needed, to map the message data to the format

required by a business service (for outbound messages) or the originating client (for inbound messages). They then route the messages to the appropriate location. For information about configuring message flows, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

36.6.5 Security-Related Validation for Proxy Services

When you activate a session that contains changes to an active proxy service, Oracle Service Bus validates the changes to ensure that you have created all of the credentials that the proxy service's static endpoints require. For example, if you configured a proxy service to have a Web service as a static endpoint and the Web service requires a digital signature, Oracle Service Bus verifies that you have associated a service key provider with the proxy service and that the service key provider contains a key-pair binding that can be used as a digital signature.

If a session contains a change to the key-pair bindings of a service key provider, Oracle Service Bus validates the change against all of the proxy services that use the service key provider. For example, if you remove the encryption key-pair, Oracle Service Bus reports a validation error for any proxy service that references the service key provider and whose endpoint requires encryption.

The following criteria determine when Oracle Service Bus performs this security-related validation and the actions that it takes during validation:

- If a proxy service specifies a static route and operation, Oracle Service Bus determines which credentials the static route and operation require. If the proxy service is missing the required credentials, Oracle Service Bus will not commit the session until you add the missing credentials.

- If a proxy service specifies a static route but the operation is passed through from the inbound request, Oracle Service Bus determines which credentials the static route and each of the route's operations require. If the proxy service is missing the required credentials, Oracle Service Bus issues a validation warning but lets you commit the session.

- If a proxy service specifies a dynamic route and operation, Oracle Service Bus cannot validate the security requirements and you risk the possibility of runtime errors. For information about dynamic routing, see Section 37.8, "Using Dynamic Routing."

36.7 Configuring Business Services

The following sections provide an overview of business service configuration.

This section includes the following topics:

- Section 36.7.1, "Business Service Types and Transport"
- Section 36.7.2, "Configuration Settings for All Business Service Types"
- Section 36.7.3, "Configuration Settings For Each Business Service Type"
- Section 36.7.4, "Using Proxy Servers"
- Section 36.7.5, "Improving Performance by Caching Business Service Results"

For specific instructions for configuring business services, see:

- Chapter 19, "Business Services: Creating and Managing"
- "Working with Business Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

36.7.1 Business Service Types and Transport

Each business service is of a certain type, and each can be used with the transport protocols appropriate for that type. Oracle Service Bus supports several standard transports plus custom transports. The service types are described in Table 36–4, including the standard transports available for each type.

Table 36–4   Business Service Types and Transports Supported by Oracle Service Bus

Service Type	Description	Transport Protocols
WSDL Web service	A SOAP or XML business service whose interface is described by a WSDL document.	BPEL-10g DSP HTTP JCA JMS LOCAL[1] SB SOA-DIRECT WS
Transport Typed Service (no WSDL)	A service that uses an EJB, JEJB, or Flow (Split-Join) transport.	EJB FLOW JEJB
Messaging Type Service (no WSDL)	A messaging service where the request message and the response message can be of different data types, including binary, text, MFL, and XML.	EMAIL FILE FTP HTTP JMS LOCAL MQ SFTP TUXEDO
Any SOAP Service (no WSDL)	A SOAP service that does not have an explicitly defined, concrete interface.	DSP HTTP JMS LOCAL SB

Table 36–4   (Cont.) Business Service Types and Transports Supported by Oracle Service

Service Type	Description	Transport Protocols
Any XML Service (no WSDL)	An XML service that does not have an explicitly defined, concrete interface.	DSP
		EMAIL
		FILE
		FTP
		HTTP
		JMS
		LOCAL
		MQ
		SB
		SFTP
		TUXEDO[2]

[1]   Oracle recommends using the local transport for communication between two proxy services. For more information on local transport, see "Local Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

[2]   For a Tuxedo transport-based service, if the service type is XML, an FML32 buffer with an FLD_ MBSTRING field from a Tuxedo client will not be transformed to XML.

For information about configuring proxy and business services based on various transport protocols, see "Transports" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

36.7.2 Configuration Settings for All Business Service Types

Each business service type is modeled following the same pattern. Each service type must define these configurations:

- Binding definition

- Runtime configuration

- Runtime variables ($operation, $body, $header, $attachments)

The business service configuration properties described in Table 36–5, below, are common to all service types. Properties specific to individual service types are described in Section 36.7.3, "Configuration Settings For Each Business Service Type."

Table 36–5   Common Configuration Properties for Business Services

Property	Description
Resource Definition	The resource definition consists of:
	■ The service name (that is, project, path, and local name)
	■ An optional description for the service
	■ The service type

Table 36–5   (Cont.) Common Configuration Properties for Business Services

Property	Description
Transport Configuration	You can configure the following parameters for each business service:
	■ List of `<string URI, integer weight>` pairs—for example, `<http://www.example.com, 100>`. For a random-weighted list, the list should contain at least one element.
	■ Load-balancing algorithm—enumeration, one of round-robin, random, or random-weighted. If you select random-weighted, the weights are applicable for each URI.
	■ Retry Count
	■ Retry Iteration Interval
	■ Retry Application Errors
	The transport you select must be able to support the transport mode (that is, request/response, one-way or both) required by the binding definition, and be configured accordingly.
	For services exchanging messages in both modes (request/response and one-way), you must configure the binding layer so that it can select the transport mode accordingly. This occurs automatically when the service is a concrete type, as it is described in the binding definition. When it is not a concrete type, to configure the binding layer, you must use the routing options action in the message flow to set the mode for a route or publish.
	Based on the transport and WSDL or interface, the transport mode is automatically selected, but you can overwrite it using the routing options action for a route or publish action.
Result Caching	All business services let you determine if and how to cache the results of business service invocations. This feature, available in the Message Handling section of a service configuration, improves performance on services whose results seldom change. When you enable result caching, service results are returned to the client from the cache instead of invoking the external service to obtain results.
	For more information, see Section 36.7.5, "Improving Performance by Caching Business Service Results."

36.7.3 Configuration Settings For Each Business Service Type

Configuration properties specific to individual business service types are described in Table 36–6:

Table 36–6   Configuration Properties for Different Business Service Types

Property	Description
WSDL Web service	See Section 36.4, "Using a WSDL to Define a Service."

Table 36–6   (Cont.) Configuration Properties for Different Business Service Types

Property	Description
Any SOAP Service	**Binding Definition**: The only information this service type defines is that the service is receiving or sending SOAP messages—regardless of their WSDL binding definition. Therefore the binding configuration for this type is empty.
	In addition, as there is no binding configuration, the combination of this type and the content-type of the message is sufficient to determine whether there are attachments to the message.
	As per their definition, any services (SOAP or XML) do not have any WSDL definition.
	Runtime Variables:
	The $body and $header variables respectively hold the `<soap:Body>` and `<soap:Header>` of the SOAP message to the business service being routed to or published. (For SOAP 1.1 proxies, $body and $header use SOAP 1.1 namespace Body and namespace; for SOAP 1.2 proxies, they use SOAP 1.2 namespace Body and namespace.)
	The $attachments variable contains the SOAP message attachments if any.
	To learn more about the message context variables, see Section 39.3, "Message-Related Variables."
Transport Typed	Transport-typed services have an empty binding definition. A WSDL is not specified. Instead the transport automatically defines the WSDL for the service. A zip containing this WSDL can be exported from the Oracle Service Bus Administration Console or the Oracle Service Bus plug-ins for Eclipse. This WSDL will not have a port defined.
	The EJB/JEJB Transport-Typed Service is an outbound transport to access EJBs from Oracle Service Bus. It is a self-described transport that generates a WSDL to describe its service interface. The EJB and JEJB transport features include transaction and security context propagation.
	Business services built using an EJB or JEJB transport can be used for publish, service callout, and service invocation.
Any XML Service	**Binding Definition**: The only information this service type defines is that the service is receiving/sending XML messages—regardless of their WSDL binding definition. Therefore, the binding configuration for this type is empty.
	In addition, as there is no binding configuration, the combination of this type and the content-type of the message is sufficient to determine whether there are attachments to the message.
	As per their definition, any services (SOAP or XML) do not have any WSDL definition.
	Runtime Variables:
	The $body variable holds the incoming XML message wrapped in a `<soap:Body>` element. (For SOAP 1.1 proxies, $body uses SOAP 1.1 namespace Body; for SOAP 1.2 proxies, it uses SOAP 1.2 namespace Body.)
	The $attachments variable contains message attachments if there are any.
	The $header variable is not applicable to this service type and is set to its default value.
	To learn more about the message context variables, see Section 39.3, "Message-Related Variables."

Table 36–6 (Cont.) Configuration Properties for Different Business Service Types

Property	Description
Messaging Service	**Binding Definition**: The binding definition for messaging services consists of configuring the content-type of the messages that are exchanged. The content-type for the response does not need to be the same as for the request; therefore, the response is configured separately (for example, the service could accept an MFL message and return an XML acknowledgment receipt).
	By definition, messaging-based services do not have any WSDL definition. It is not possible to request a WSDL document for those services.
	The following content types are available for the request and response:
	- None
	- Binary
	- Text
	- MFL
	- XML
	- Java
	Runtime Variables:
	This service type is message based.
	The $body variable holds the incoming message wrapped in a `<soap:Body>` element. (For SOAP 1.1 proxies, $body uses SOAP 1.1 namespace Body; for SOAP 1.2 proxies, it uses SOAP 1.2 namespace Body.)
	The $header variable is not applicable to this service type, and is set to its default value.
	The $attachments variable contains message attachments if there are any.
	To learn more about the message context variables, see Section 39.3, "Message-Related Variables."

If a business service requires Web service security, make sure the WSDL you specify has the necessary WS-Policies attached when you create the business service. Furthermore, if the WS-Policy of the business service requires encryption, make sure the public certificate of the business service is embedded in the WSDL. If the business service is an Oracle WebLogic Server 9.0 or later Web service, you can retrieve its WSDL using the `http://<host>:<port>/<service url>?WSDL` URL, the public certificate will be automatically embedded for you if necessary. For more information, see Section 4.30, "Viewing Resources in a Web Browser."

36.7.4 Using Proxy Servers

You can configure business services to route messages through a proxy server. You do this by creating a proxy server resource that specifies one or more proxy servers together with the necessary credentials. You can then associate this proxy server resource with a business service. This instructs Oracle Service Bus to connect to the business service through the configured proxy server.

Adding multiple proxy servers to a resource enables Oracle Service Bus to perform load balancing and offer fault tolerance among the configured proxy servers. The credentials are used when opening a connection to the proxy server. If a particular proxy server is not reachable, Oracle Service Bus attempts to use the next proxy server in the configuration. If all proxy servers are unreachable, Oracle Service Bus tries to connect to the back end service directly. If that too fails, a fault is raised and sent back to the caller.

When configuring a proxy server, you can specify the clear text or SSL port number for the server, in addition to the host name or IP address. The port specified in the proxy server resource configuration corresponds to the actual physical port of the Web proxy server and not the port of the back end service. The end point configuration on the business service, however, corresponds to the actual end point of the back end service. This is true even in the case of SSL/TLS Tunneling using HTTP CONNECT command.

NOTE: While some Web proxy servers support HTTP CONNECT using both clear text and SSL sockets, other servers support HTTP CONNECT using only clear text sockets. For example, Oracle iPlanet Web Proxy Server 4.0 supports HTTP CONNECT using clear text and SSL sockets while Apache Web Server 2.2 in proxy server mode supports HTTP CONNECT using only clear sockets.

Typically, for SSL/TLS Tunneling support, you do not need to install server or Certificate Authority (CA) certificates on the physical proxy server. Instead, the Web proxy server uses the certificate received from the caller (Oracle Service Bus, in this case) to open a secure socket. Similarly, when the server hosting the business application requires a client certificate, the certificate of the server hosting the proxy service (Oracle Service Bus) is used for authentication.

36.7.5 Improving Performance by Caching Business Service Results

If you use business services that return somewhat static results that do not change often, you can configure those business services to cache results. When you enable result caching, the service returns results from cache rather than invoking the external service. This improves performance by reducing network overhead to access the external service. Result caching also helps improve scalability by reducing the load on the back-end servers that host the external service.

In this section, the term "result cache" refers to the cache itself that all business services share to store their respective results, and "cached result" refers to a single result in the result cache.

For business services that use result caching, you can control the time to live for the cached result. After the cached result expires, the next business service call results in invoking the back-end service to get the result. This result is then stored in the cache for future requests to access.

The result caching mechanism that Oracle Service Bus uses is Oracle Coherence (Enterprise Edition), which is included with Oracle WebLogic Server. If a license for Oracle WebLogic Server is available for a certain processor, Coherence can be used on that computer.

The Oracle Service Bus implementation of result caching includes global enable/disable, cache message variables, configuration fields on each business service, and cache options for service statistics, debugging, alert rules, and smart search.

This section describes the Oracle Service Bus implementation of result caching.

36.7.5.1 How Result Caching Works

Figure 36–1 illustrates a client invoking a business service and receiving a response that contains cached results.

> **Note:** Result caching works only with request/response operations.

Figure 36–1   Business Service Result Caching
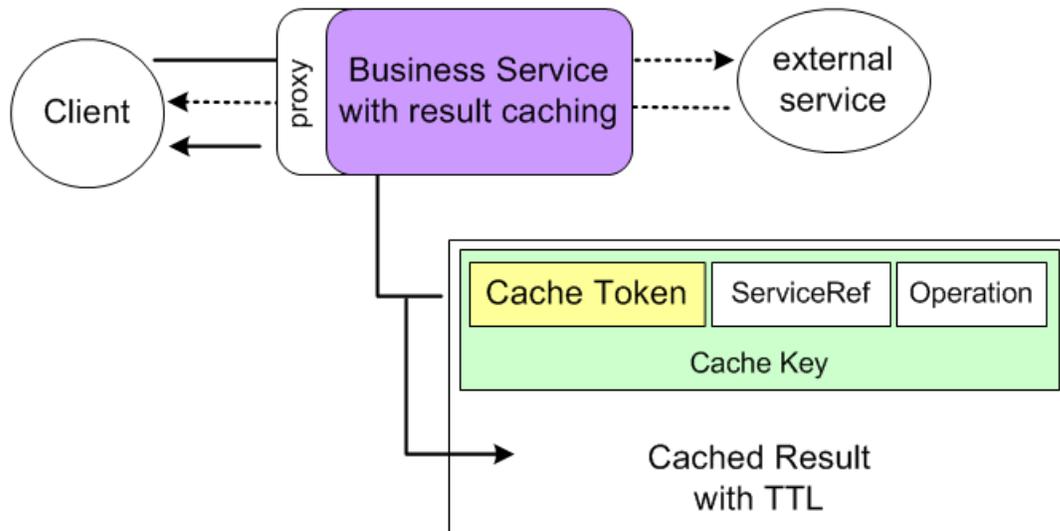
Each cached result is uniquely identified by a cache key that is made up of the ServiceRef (the unique identifier for the service which is the fully qualified path name of the service), the operation being invoked, and a cache token String. The cache token helps to uniquely identify a single cache result among other cache results for one business service. You control the value of the cache token. You can set the cache token either by configuring the Cache Token expression in the Result Caching configuration for the business service or by using the cache-token metadata element in $transportMetaData using the proxy service message flow.

If the business service locates cached results through a cache key, it returns those cached results to the client instead of invoking the external service directly.

In Figure 36–1, the solid arrows represent the message path between the client and a cached result. The dotted arrows show the message path if no cached result exists. If no cached result exists, the business service invokes the external service directly, returns the result to the client, and stores the result in cache. A result cache could be empty for a number of reasons, including: a first-time invocation where no cache yet exists, a caching error, or because the cache was flushed.

For cache expiration, cached results have a time-to-live (TTL) attribute. You can configure cache expiration either with the Expiration Time property in the Result Caching configuration on the business service or the cache-ttl element in $transportMetaData using the proxy service message flow. If Oracle Coherence finds that the TTL has expired, it flushes the cache, and the business service invokes the external service for a result. That result is then stored in the cache (if there is no error in the result), and the result is available in the cache so that it can be returned to the next request.

36.7.5.1.1   Flushing Cached Results   Oracle Service Bus/Coherence can flush an individual cached result, all cached results for a business service, or the entire result cache (all cached results for all Oracle Service Bus business services). The following events illustrate how Oracle Service Bus/Coherence flush cache:

■   Cache TTL has expired – Each cached result has its own TTL. When a TTL is reached, Oracle Coherence flushes that individual cached result.

- Disable result caching on a single business service – When you disable result caching on a business service, Oracle Service Bus triggers flushing of all cached results for that business service in Oracle Coherence.

- Update, Rename, or Delete a business service – If you update, rename, or delete a business services, Oracle Service Bus triggers flushing of all cached results for that business service from Oracle Coherence.

- Update dependent resource – When you update a dependent resource, such as a WSDL, Oracle Service Bus triggers flushing all cached results for that business service from Oracle Coherence. However, changes to the following dependent resources do not cause cache flushing: Service Provider, UDDI Registry, Alert Destination.

- Globally disable result caching – When you globally disable result caching, Oracle Service Bus triggers flushing the entire result cache (all cached results for all business services) from Oracle Coherence.

For more information on working with and configuring result caching, see:

- Section 27.2.11, "Enabling Result Caching Globally"

- Section 19.1, "Creating and Configuring Business Services"

- "Business Service Message Handling Configuration Page" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

36.7.5.2 Result Caching Best Practices

This section describes best practices for using result caching.

36.7.5.2.1 Security Best Practices with Result Caching Because cached results bypass the security of invoking an external service directly, do not use result caching with business services that provide security with a non-static service account or a WS-Security policy.

36.7.5.2.2 Production Best Practices with Result Caching Before deploying an Oracle Service Bus environment that will use result caching in production, you should plan and implement Coherence setup and configuration to allow for best performance, as described in the "Production Checklist" in the *Oracle Coherence Developer's Guide* at http://download.oracle.com/docs/cd/E15357_01/coh.360/e15723/deploy_checklist.htm#CHDJEJEI.

36.7.5.3 Result Cache Metadata

Following are the metadata used for result caching:

36.7.5.3.1 Request Metadata Following are the request metadata used with result caching: cache-token and cache-ttl, both String values. You can configure both in the business service configuration. You can also leave the cache token or TTL undefined in the business service and provide the cache token or TTL in the request with these metadata. When you set the cache token or TTL in the request, those values override any cache token or TTL you have defined in the business service configuration.

For more information, see Section 19.2.23, "Message Handling Configuration Page" for business service configuration.

36.7.5.3.2 Response Metadata Following are the response metadata used with result caching:

- cache-token – Contains the cache token that was used to retrieve content from the result cache or add content to the result cache after invoking the external service.

- cache-originated – Contains a boolean value, true or false. A value of true means the returned content came from the result cache. A value of false means the returned value came from invoking the external service.

36.7.5.4 Testing Result Caching

Result caching takes effect only when the business service configured with result caching is invoked (for example, with a Route or Service Callout) from a proxy service. Therefore, in order to test result caching, do not invoke the business service directly from the Test Console. Instead, use the Test Console to test the proxy service that invokes the business service.

You can, however, invoke Split-Join business services directly to test result caching.

36.7.5.5 Result Caching Advanced Configuration

In each Oracle Service Bus domain, you can modify how the domain uses Oracle Coherence for business service result caching. This section describes default settings and when you should change them.

36.7.5.5.1 Result Cache Setup and Tuning You can tune cache schemes for the Oracle Service Bus result cache by modifying *DOMAIN_HOME*/config/osb/coherence/osb-coherence-cache-config.xml. By default, a distributed cache scheme is used for the Oracle Service Bus result cache. For more information, see "Cache Configuration Elements" in the *Oracle Coherence Developer's Guide* at http://download.oracle.com/docs/cd/E15357_01/coh.360/e15723/appendix_cacheconfig.htm.

36.7.5.5.2 Working with Unicast and Multicast Oracle Service Bus provides its own default Oracle Coherence configuration for the servers in a domain. The configuration file, located at *DOMAIN_HOME*/config/osb/coherence/osb-coherence-override.xml, provides different properties that control server access to Oracle Coherence cache.

This section describes the default Oracle Coherence configuration settings for Oracle Service Bus single- and multi-server domains, and describes scenarios in which you need to modify the default settings.

Table 36–7 shows the default system property values in osb-coherence-override.xml for a single-server domain created with the Oracle Service Bus for developers template. These unicast settings restrict Oracle Coherence cache access to only the local server. With this configuration, nodes started on different servers do not join the same Oracle Coherence cluster to share cached information.

You can modify the system properties either in osb-coherence-override.xml (requires server restart) or by specifying the system properties on server startup.

Table 36–7 osb-coherence-override.xml Default Settings for an "Oracle Service Bus for developers" Domain Topology

Server Details	Element/Property Name in the File	System Property	Default Value
Unicast Address	<address>	OSB.coherence.localhost	127.0.0.1
Unicast Port	<port>	OSB.coherence.localport	7890

Table 36–7   (Cont.) osb-coherence-override.xml Default Settings for an "Oracle Service Bus for developers" Domain Topology

Server Details	Element/Property Name in the File	System Property	Default Value
Well-Known Address	<address>	OSB.coherence.wka	127.0.0.1
Well-Known Port	<port>	OSB.coherence.wka.port	7890
Network Time to Live (TTL)	<time-to-live>	OSB.coherence.ttl	0 For more information on this setting, see "Network" in the *Oracle Coherence Developer's Guide* at http://download.oracle.com/docs/cd/E14526_01/coh.350/e14509/appproductcheck.htm#COHDG305.

Table 36–8 shows the default system property values in osb-coherence-override.xml for a multi-server domain created with the Oracle Service Bus template. The default multicast values create an Oracle Coherence cluster that is shared by any Oracle WebLogic Server node on the same subnet created from the same template.

You can modify the system properties either in osb-coherence-override.xml (requires server restart) or by specifying the system properties on server startup.

Table 36–8   osb-coherence-override.xml Default Settings for an "Oracle Service Bus" Domain Topology

Server Details	Element/Property Name in the File	System Property	Default Value
Unicast Address	<address>	OSB.coherence.localhost	127.0.0.1
Unicast Port	<port>	OSB.coherence.localport	7890
Unicast well-known address placeholders	<address>	OSB.coherence.wka*N*	The file provides placeholders for 9 well-known addresses (with no address values) for you to configure and use as needed.
Unicast well-known address port placeholders	<port>	OSB.coherence.wka*N*.port	The file provides placeholders for 9 well-known address ports (with no port values) for you to configure and use as needed.
Multicast Listener Address	<address>	OSB.coherence.clusteraddress	228.8.8.8
Multicast Listener Port	<port>	OSB.coherence.clusterport	9888

The Multicast Address and Multicast Port properties determine how member nodes of the Oracle Coherence cluster are located. This combination must be unique for each Oracle Coherence cluster that is running within the same subnet.

> **Note:** Even though the Oracle Service Bus template is configured to
> use a multicast listener by default, a best practice is to configure the
> Oracle Coherence cluster to use a unicast listener with an explicit list
> of nodes for the Oracle Coherence cluster.
>
> To switch to unicast, delete the multicast values and enter well-known
> addresses for unicast listeners. For example:
>
> ```
> <well-known-addresses>
> <socket-address id="1">
> <address system-property="OSB.coherence.wka1">127.0.0.1</address>
> <port system-property="OSB.coherence.wka1.port">7890</port>
> </socket-address>
> ...
> ```
>
> For more information, see "well-known-addresses" at
> http://coherence.oracle.com/display/COH35UG/well-known-addre
> sses.
>
> You can also provide overrides using system properties, as described
> in "Configure Oracle Coherence for the Oracle Service Bus Result
> Cache" in the *Oracle Fusion Middleware High Availability Guide*.

The following configurations require manual modification of
osb-coherence-override.xml (or providing system property overrides as described in
the previous note) to ensure the desired sharing of an Oracle Coherence cluster among
multiple servers:

- If you want to switch from a multicast listener to a unicast listener with
  well-known addresses in a cluster, as described in the previous note.

- If you have multiple Oracle WebLogic Server clusters in the same subnet, modify
  the relevant Oracle Coherence address and port properties to ensure desired
  sharing of an Oracle Coherence cluster. Do not use the same address and port as
  those used for the Oracle WebLogic Server cluster.

- If you have multiple Admin Servers with Managed Servers in the same subnet,
  modify the relevant Oracle Coherence address and port properties to ensure
  desired sharing of an Oracle Coherence cluster.

- If you have any combination of Oracle WebLogic Server clusters and Admin
  Servers with Managed Servers in the same subnet, modify the relevant Oracle
  Coherence address and port properties to ensure desired sharing of an Oracle
  Coherence cluster.

- If multiple Oracle Coherence clusters are running in the same subnet, modify the
  Multicast Address and Multicast Port to specify which Oracle Coherence cluster a
  node should connect to.

36.7.5.5.3 Using an Out-of-Process Coherence Cache Server If you plan to use result
caching heavily with Oracle Service Bus and want to avoid using too much heap space
for result caching, you can set up a Coherence cache server to run on its own JVM
rather than sharing an Oracle Service Bus domain JVM. Running a Coherence cache
server outside of an Oracle Service Bus JVM—out of process—lets the Coherence cache
server use its own heap space without affecting the heap space Oracle Service Bus uses
to process messages.

Oracle Service Bus provides support for out-of-process Coherence caching with the
following JAR file: *OSB_ORACLE_HOME*/lib/osb-coherence-client.jar.

The following steps describe how to use an out-of-process Coherence cache server:

> **Note:** Any out-of-process Coherence cache server used with Oracle Service Bus must use the same version of Coherence as the version included with Oracle Service Bus.

1. Configure the Oracle Service Bus domain to support the out-of-process Coherence cache server.

    a. Disable local caching on each Oracle Service Bus node by adding the following argument to the Oracle Service Bus node startup:

        ```
        -Dtangosol.coherence.distributed.localstorage=false
        ```

    b. Set the Coherence cluster name with the following argument.

        ```
        -DOSB.coherence.cluster=cluster_name
        ```

2. Configure the out-of-process Coherence cache server startup:

    a. Make sure osb-coherence-client.jar is on the Coherence cache server classpath. For example:

        ```
        -classpath coherence.jar:OSB_ORACLE_HOME/lib/osb-coherence-client.jar
        ```

    > **Note:** The path in this example shows osb-coherence-client.jar in the default location in an Oracle Service Bus installation. In an out-of-process Coherence cache server that does not have Oracle Service Bus installed, copy this file from the Oracle Service Bus domain to the Coherence cache domain and reference it in the startup argument for all Coherence nodes.

    b. Use osb-coherence-override.xml by adding the following argument to the Coherence cache server startup:

        ```
        -Dtangosol.coherence.override=DOMAIN_
        HOME/config/osb/coherence/osb-coherence-override.xml
        ```

        This step ensures that the same networking configuration is used for both the Oracle Service Bus Coherence nodes and the out-of-process Coherence cache server. For more information on this file, see Section 36.7.5.5.2, "Working with Unicast and Multicast."

    > **Note:** The path in this example shows the osb-coherence-override.xml file in the default location in an Oracle Service Bus installation. In an out-of-process Coherence cache server that does not have Oracle Service Bus installed, copy this file from the Oracle Service Bus domain to the Coherence cache domain and reference it in the startup argument for all Coherence nodes.

    c. Use osb-coherence-cache-config.xml by adding the following argument to the Coherence cache server startup:

        ```
        -Dtangosol.coherence.cacheconfig=DOMAIN_
        HOME/config/osb/coherence/osb-coherence-cache-config.xml
        ```

See the previous note.

d. Add the following arguments to the Coherence cache server startup to set the host and port:

`-DOSB.coherence.localhost=`*HOSTNAME* and
`-DOSB.coherence.localport=`*PORT* where host:port is unique in the cluster.

e. If you use a unicast listener for cluster membership (recommended), and you do not include well-known listeners in the osb-coherence-override.xml file as described in Table 36–8, add as many well-known addresses as needed to the Coherence cache server startup using the following arguments:

`-DOSB.coherence.wka[1-9]` and `-DOSB.coherence.wka[1-9].port`

For more information on using unicast for Oracle Coherence with Oracle Service Bus, see "Configure Oracle Coherence for the Oracle Service Bus Result Cache" in the *Oracle Fusion Middleware High Availability Guide*.

f. Make sure the Coherence cluster name is the same for the out-of-process Coherence cache server as it is for the Oracle Service Bus server by using the following startup argument:

`-Dtangosol.coherence.cluster=OSB-cluster`

The Oracle Service Bus cluster is always called "OSB-cluster".

Starting the Servers

When using an out-of-process Coherence cache server, which involves disabling local caching on the Oracle Service Bus server as described at the beginning this procedure, be sure to start the out-of-process Coherence cache server before starting the Oracle Service Bus server.

One way to start the Coherence cache server is to use a Java command with the desired arguments. The following example startup command assumes osb-coherence-override.xml contains the proper configurations for host, port, and well-known addresses (unicast listener):

```
java
-classpath coherence.jar:C:/oracle/Oracle_OSB1/lib/osb-coherence-client.jar
-Dtangosol.coherence.override=C:/oracle/user_projects/domains/osb_
domain/config/osb/coherence/osb-coherence-override.xml
-Dtangosol.coherence.cacheconfig=C:/oracle/user_projects/domains/osb_
domain/config/osb/coherence/osb-coherence-cacheconfig.xml
-Dtangosol.coherence.cluster=OSB-cluster
com.tangosol.net.DefaultCacheServer
```

For more information on starting a Coherence cache server, see the Oracle Coherence documentation at http://download.oracle.com/docs/cd/E15357_01/index.htm.

36.7.5.5.4 More Information on Configuring and Using Oracle Coherence You can perform many other types of cache configuration flexibly, without changing application code, using the Oracle Coherence configuration framework. For example, you can use attributes modify cache type and behavior, and you can query cache. For more information, see:

- "Configure Oracle Coherence for the Oracle Service Bus Result Cache" in the *Oracle Fusion Middleware High Availability Guide*

- Oracle Coherence documentation at
  http://download.oracle.com/docs/cd/E15357_01/index.htm

36.8 Viewing Resource Details

Oracle Service Bus provides a resource servlet that is used to expose the resources registered in Oracle Service Bus through a URL. For more information, see Section 4.30, "Viewing Resources in a Web Browser."

37

Modeling Message Flow in Oracle Service Bus

This chapter describes the high-level aspects and concepts of creating and configuring message flows using the Oracle Service Bus Administration Console. Topics include message flow components, message transformation, routing and service callout, error handling, message context, and quality of service (QoS).

In Oracle Service Bus, a message flow defines the implementation of a proxy service. You can create and configure Oracle Service Bus proxy services in the Oracle Service Bus Administration Console or the Oracle Service Bus plug-ins for Eclipse. This section describes message flows and presents guidelines for designing them.

The following sections describe message flows in Oracle Service Bus:

- Section 37.1, "Message Flow Components"
- Section 37.2, "Branching in Message Flows"
- Section 37.3, "Configuring Actions in Stages and Route Nodes"
- Section 37.4, "Performing Transformations in Message Flows"
- Section 37.5, "Constructing Service Callout Messages"
- Section 37.7, "Handling Errors in Message Flows"
- Section 37.8, "Using Dynamic Routing"
- Section 37.9, "Accessing Databases Using XQuery"
- Section 37.10, "Understanding Message Context"
- Section 37.11, "Working with Variable Structures"
- Section 37.12, "Quality of Service"
- Section 37.13, "Using Work Managers with Oracle Service Bus"
- Section 37.14, "Content Types, JMS Type, and Encoding"
- Section 37.15, "Throttling Pattern"
- Section 37.16, "WS-I Compliance"
- Section 37.17, "Converting Between SOAP 1.1 and SOAP 1.2"

For instructions on creating and configuring message flows in the Oracle Service Bus Administration Console, see:

- Chapter 22, "Proxy Services: Message Flow"
- Chapter 21, "Proxy Services: Actions"
- Chapter 24, "Proxy Services: Error Handlers"

For instructions on creating and configuring message flows in Eclipse, see the following topics in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*:

- "Working with Proxy Services"

- "Working with Proxy Service Message Flows"

37.1 Message Flow Components

A message flow is composed of components that define the logic for routing and manipulating messages as they flow through a proxy service. Nodes are configured to route messages through the message flow, and stages and actions contain rules for processing and transforming messages.

Most of the processing logic in a message flow is handled in pipelines. A pipeline is a named sequence of stages representing a non-branching one-way processing path. Pipelines belong to one of the following categories:

- Request pipelines process the request path of the message flow.

- Response pipelines process the response path of the message flow.

- Error pipelines handle errors for stages and nodes in a message flow as well as errors at the level of the message flow (service).

To implement the processing logic of a proxy service, request and response pipelines are paired together in pipeline pair nodes. These pipeline pair nodes can be combined with other nodes into a single-rooted tree structure to control overall flow.

Table 37–1 describes the components available for defining message flows.

Table 37–1   Message Flow Components

Component Type	Summary
Start node	Every message flow begins with a start node. All messages enter the message flow through the start node, and all response messages are returned to the client through the start node. There is nothing to configure in a start node.
Pipeline pair node	A pipeline pair node combines a single request pipeline and a single response pipeline in one top-level element. A pipeline pair node can have only one direct descendant in the message flow. During request processing, only the request pipeline is executed when Oracle Service Bus processes a pipeline pair node. The execution path is reversed when Oracle Service Bus processes the response pipeline.
Stage	Request pipelines, response pipelines, and error handlers can contain stages, where you configure actions to manipulate messages passing through the pipeline.
	See also Section 37.3, "Configuring Actions in Stages and Route Nodes."
Error handler	An error handler can be attached to any node or stage, to handle potential errors at that location.
	See also Section 37.7, "Handling Errors in Message Flows."
Branch node	A branch node allows processing to proceed along exactly one of several possible paths. Operational branching is supported for WSDL-based services, where the branching is based on operations defined in the WSDL. Conditional branching is supported for conditions defined in an XPath-based switch table.
	See also Section 37.2, "Branching in Message Flows."

Table 37–1   (Cont.) Message Flow Components

Component Type	Summary
Route node	A route node performs request/response communication with another service. It represents the boundary between request and response processing for the proxy service. When the route node dispatches a request message, the request processing is considered complete. When the route node receives a response message, the response processing begins. The route node supports conditional routing as well as request and response transformations.
	Because a route node represents the boundary between request and response processing, it cannot have any descendants in the message flow.
	See also Section 37.3, "Configuring Actions in Stages and Route Nodes."

Figure 37–1 shows a high level view of components in a message flow definition.

Figure 37–1   Components of Message Flow
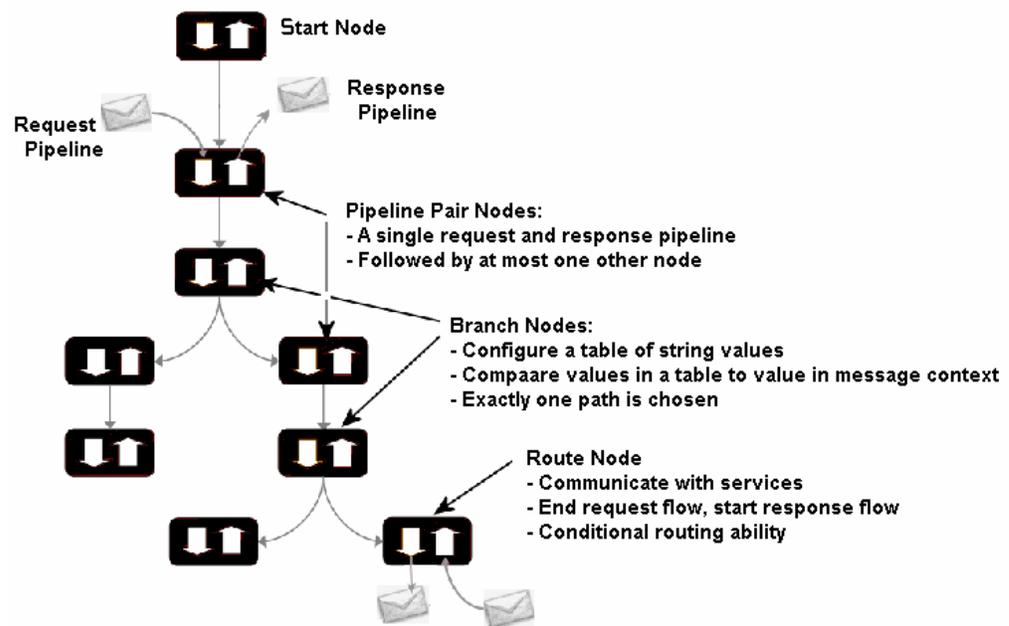
37.1.1 Building a Message Flow

The only components required in a message flow are a start node and a route node. No restrictions exist on what other components can be chained together in the message flow. You could create a single route node that contained all the logic for the flow. Or, you could link two pipeline pair nodes without a branch node in between. If you use branch nodes, each branch node could start with a different element. One branch could terminate with a route node, another could be followed by a pipeline pair, and yet another could have no descendant. (When a branch with no descendants is executed at runtime, response processing begins immediately.)

However, in general, a message flow is likely to be designed in one of the following ways:

■   For non-operational services (services that are not based on WSDLs with operations), the flow consists of a single pipeline pair at the root followed by a route node.

- For operational services, the flow consists of a single pipeline pair at the root, followed by a branch node based on an operation, with each branch consisting of a pipeline pair followed by a route node.

37.1.2 Message Execution

Table 37–2 and Table 37–3 briefly describe how request and response messages are processed in a typical message flow.

Table 37–2   Path of a Request Message During a Message Flow

Message Flow Node	What Happens During Message Processing?
Request Processing	Request Processing container.
Start node	Request processing begins at the start node.
Pipeline pair node	Executes the request pipeline only.
Branch node	Evaluates the branch table and proceeds down the relevant branch.
Route node	Performs the route along with any request transformations. In the message flow, regardless of whether routing takes place or not, the route node represents the transition from processing a request to processing a response. At the route node, the direction of the message flow is reversed. If a request path does not have a route node, the response processing is initiated in the reverse direction without waiting for any response.

Table 37–3   Path of a Response Message During a Message Flow

Message Flow Node	What Happens During Message Processing?
Response Processing	Skips any branch nodes and continues with the node that preceded the branch.
Route node	Executes any response transformations.
Branch node	Skips any branch nodes and continues with the node that preceded the branch.
Pipeline pair node	Executes the response pipeline.
Start node	Sends the response back to the client.

37.2 Branching in Message Flows

Two kinds of branching are supported in message flows: *operational* branching, configured in an operational branch node, and *conditional* branching, configured in a conditional branch node.

37.2.1 Operational Branching

When message flows define WSDL-based proxy services, operation-specific processing is required. When you create an operational branch node in a message flow, you can build branching logic based on the operations defined in the WSDL.

You must use operational branching when a proxy service is based on a WSDL with multiple operations. You can consider using an operational branch node to handle messages separately for each operation.

37.2.2 Conditional Branching

Use conditional branching to branch based on a specified condition, for example the document type of a message.

Conditional branching is driven by a lookup table with each branch tagged with simple, unique string values, for example, `QuantityEqualToOrLessThan150` and `QuantityMoreThan150`.

You can configure a conditional branch to branch based on the value of a variable in the message context (declared, for example, in a stage earlier in the message flow), or you can configure the condition to branch based on the results of an XPath expression defined in the branch itself.

At runtime, the variable or the expression is evaluated, and the resulting value is used to determine which branch to follow. If no branch matches the value, the default branch is followed. A branch node may have several descendants in the message flow: one for each branch, including the default branch. You should always define a default branch. You should design the proxy service in such a way that the value of a lookup variable is set before reaching the branch node.

For example, consider the following case using a lookup variable. A proxy service is of type any SOAP or any XML, and you need to determine the type of the message so you can perform conditional branching. You can design a stage action to identify the message type and then design a conditional branching node later in the flow to separate processing based on the message type.

Now consider the following case using an XPath expression in the conditional branch node. You want to branch based on the quantity in an order. That quantity is passed by a variable that can be retrieved from a known location in `$body`.

You could define the following XPath expression to retrieve the quantity:

```
declare namespace openuri="http://www.openuri.org/";
declare namespace com="example.com/demo/orders/cmnCust";
./openuri:processCust/com:cmnCust/com:Order_Items/com:Item/com:Quantity
```

The condition (for example, `<500`) is then evaluated in order down the message flow against the expression. Whichever condition is satisfied first determines which branch is followed. If no branch condition is satisfied, then the default branch is followed.

You can use conditional branching to expose the routing alternatives for the message flow at the top level flow view. For example, consider a situation where you want to invoke service A or service B based on a condition known early in the message flow (for example, the message type). You could configure the conditional branching in a routing table in the route node. However, that makes the branching somewhat more difficult to follow if you are just looking at the top level of the flow. Instead, you could use a conditional branch node to expose this branching in the message flow itself and use simple route nodes as the subflows for each of the branches.

Consider your business scenario before deciding whether you configure branching in the message flow or in a stage or route node. When making your decision, remember that configuring branches in the message flow can be awkward in the design interface if a large number of branches extend from the branch node.

37.3 Configuring Actions in Stages and Route Nodes

Actions provide instructions for handling messages in pipeline stages, error handler stages, and route nodes. The context determines which actions are available in the

Oracle Service Bus Administration Console or in the Oracle Service Bus plug-ins for Eclipse, as described in the following sections:

- Section 37.3.1, "Communication Actions"

- Section 37.3.2, "Flow Control Actions"

- Section 37.3.3, "Message Processing Actions"

- Section 37.3.4, "Reporting Actions"

See Also

- Chapter 21, "Proxy Services: Actions"

- "Working with Proxy Service Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

37.3.1 Communication Actions

Communication actions control message flow. Table 37–4 describes the communication actions.

Table 37–4   Communication Actions

Action	Use to...	Available in
Dynamic publish	Publish a message to a service specified by an XQuery expression.	■ Pipeline stage ■ Error handler stage ■ Route node
Publish	Identify a statically specified target service for a message and to configure how the message is packaged and sent to that service.	■ Pipeline stage ■ Error handler stage
Publish table	Publish a message to zero or more statically specified services. Switch-style condition logic is used to determine at runtime which services will be used for the publish.	■ Pipeline stage ■ Error handler stage
Routing options	Modify any or all of the following properties in the outbound request: URI, Quality of Service, Mode, Retry parameters, Message Priority.	■ Pipeline stage
Service callout	Configure a synchronous (blocking) callout to an Oracle Service Bus-registered proxy or business service. See Section 37.5, "Constructing Service Callout Messages."	■ Pipeline stage ■ Error handler stage
Transport headers	Set the header values in messages. See Section 37.3.5, "Configuring Transport Headers in Message Flows."	■ Pipeline stage ■ Error handler stage

37.3.2 Flow Control Actions

Flow controls actions implement conditional routing, conditional looping, and error handling. You can also use them to notify the invoker of success or to skip the rest of the actions in the stage. Table 37–5 describes the flow control actions.

Table 37–5   Flow Control Actions

Action	Use to...	Available in
For each	Iterate over a sequence of values and execute a block of actions	■ Pipeline stage ■ Error handler stage
If... then...	Perform an action or set of actions conditionally, based on the Boolean result of an XQuery expression.	■ Pipeline stage ■ Route node ■ Error handler stage
Raise error	Raise an exception with a specified error code (a string) and description.	■ Pipeline stage ■ Error handler stage
Reply	Specify that an immediate reply be sent to the invoker. The reply action can be used in the request, response or error pipeline. You can configure it to result in a reply with success or failure. In the case of reply with failure where the inbound transport is HTTP, the reply action specifies that an immediate reply is sent to the invoker.	■ Pipeline stage ■ Error handler stage
Resume	Resume message flow after an error is handled by an error handler. This action has no parameters and can only be used in error handlers.	■ Error handler stage
Skip	Specify that at runtime, the execution of this stage is skipped and the processing proceeds to the next stage in the message flow. This action has no parameters and can be used in the request, response or error pipelines.	■ Pipeline stage ■ Error handler stage

37.3.3 Message Processing Actions

The actions in this category process the message flow. Table 37–6 describes the message processing actions.

Table 37–6   Message Processing Actions

Action	Use to...	Available in
Assign	Assign the result of an XQuery expression to a context variable.	■ Pipeline stage ■ Error handler stage
Delete	Delete a context variable or a set of nodes specified by an XPath expression.	■ Pipeline stage ■ Error handler stage
Insert	Insert the result of an XQuery expression at an identified place relative to nodes selected by an XPath expression.	■ Pipeline stage ■ Error handler stage
Java callout	Invoke a Java method, or EJB business service, from within the message flow.	■ Pipeline stage ■ Error handler stage

Table 37–6 (Cont.) Message Processing Actions

Action	Use to...	Available in
MFL transform	Convert message content from XML to non-XML, or vice versa, in the message pipeline. An MFL is a specialized XML document used to describe the layout of binary data. It is an Oracle proprietary language used to define rules to transform formatted binary data into XML data, or vice versa.	■ Pipeline stage ■ Error handler stage
Rename	Rename elements selected by an XPath expression without modifying the contents of the element.	■ Pipeline stage ■ Error handler stage
Replace	Replace a node or the contents of a node specified by an XPath expression. The node or its contents are replaced with the value returned by an XQuery expression. A replace action can be used to replace simple values, elements and even attributes. An XQuery expression that returns nothing is equivalent to deleting the identified nodes or making them empty, depending upon whether the action is replacing entire nodes or just node contents. The replace action is one of a set of Update actions.	■ Pipeline stage ■ Error handler stage
Validate	Validate elements selected by an XPath expression against an XML schema element or a WSDL resource. You can validate global elements only; Oracle Service Bus does not support validation against local elements.	■ Pipeline stage ■ Error handler stage

37.3.4 Reporting Actions

You use the actions in this category to log or report errors and generate alerts if required in a message flow within a stage. Table 37–7 describes the reporting actions.

Table 37–7 Reporting Actions

Action	Use to...	Available in
Alert	Generate alerts based on message context in a pipeline, to send to an alert destination. Unlike SLA alerts, notifications generated by the alert action are primarily intended for business purposes, or to report errors, and not for monitoring system health. Alert destination should be configured and chosen with this in mind. If pipeline alerting is not enabled for the service or enabled at the domain level, the configured alert action is bypassed during message processing.	■ Pipeline stage ■ Error handler stage
Log	Construct a message to be logged and to define a set of attributes with which the message is logged.	■ Pipeline stage ■ Error handler stage
Report	Enable message reporting for a proxy service.	■ Pipeline stage ■ Error handler stage

37.3.5 Configuring Transport Headers in Message Flows

The transport header action is a communication type action, and it is available in pipeline stages and error handler stages.

37.3.5.1 Configuring Global Pass Through and Header-Specific Copy Options for Transport Headers

The following options are available when you configure a transport headers action:

- The **Pass all Headers through Pipeline** option specifies that at runtime, the transport headers action passes all headers through from the inbound message to the outbound message or vice versa. Every header in the source set of headers is copied to the target header set, overwriting any existing values in the target header set.

- The **Copy Header from Inbound Request** option and the **Copy Header from Outbound Response** options specifies that at runtime, the transport headers action copies the specific header with which this option is associated from the inbound message to the outbound message or vice versa.

Use the options in a way that best suits your scenario. Both options result in the headers in the source header set being copied to the target header set, overwriting any existing value in the target set. The **Pass all Headers through Pipeline** option is executed before the header-specific **Copy Header** options. In other words, for a given transport headers action configuration, if you select **Pass all Headers through Pipeline**, there is no need to select the **Copy Header** option for given headers.

However, you can select **Pass all Headers through Pipeline** to copy all headers, and subsequently configure the action such that individual headers are deleted by selecting **Delete Header** for specific headers.

> **Caution:** Because transport headers are specific to the transport types, it is recommended that the pass-through (or copy) options only be used to copy headers between services of the same transport type. Passing (or copying) headers between services of different transport types can result in an error if the header being passed is not accepted by the target transport. For the same reasons, be careful when you specify a header name using the Set Header option.

37.3.5.2 Understanding How the Runtime Uses the Transport Headers Settings

You can use transport header actions to configure the values of the transport headers for outbound requests (the messages sent out by a proxy service in route, publish, or service callout actions) and inbound responses (the response messages a proxy service sends back to clients). In general, the header values can be:

- Specified using an XQuery expression

- Passed through from the source to the target service

- Deleted while going from the source to the target service

The transport headers action lets you set, delete, or pass-through the headers in $inbound or $outbound. If you set or delete these headers and then log $inbound or $outbound, you can see the effects of your changes. However, when the message is sent out, the Oracle Service Bus binding layer may modify or remove some headers in $inbound or $outbound and the underlying transport may in turn ignore some of these headers and use its own values. An important distinction is that any modifications

done by the binding layer on a header are done directly to `$inbound` and `$outbound`, whereas modifications done by the transport affects only the message's *wire format*. For example, although you can specify a value for the outbound `Content-Length` header, the binding layer deletes it from `$outbound` when sending the message. Consequently, the modification is visible in the response path (for example, you can see the modified value if you log `$outbound`). If you set the `User-Agent` header in `$outbound`, the HTTP transport ignores it and use its own value—however, the value in `$outbound` is not changed.

Table 37–8 describes the transport headers that are ignored or overwritten at runtime and other limitations that exist for specific transport headers.

Table 37–8   Limitations to Transport Header Values You Specify in Transport Header Actions

Transport	Description of Limitation...	Outbound Request Header	Inbound Response Header
HTTP	Oracle Service Bus runtime may overwrite these headers in the binding layer when preparing the message for dispatch. If these headers are modified, `$inbound` and `$outbound` are updated accordingly.	Content-Type	Content-Type
HTTP	The underlying transport may ignore these headers and use different values when sending the message. Any changes done by the transport will not be reflected in `$inbound` or `$outbound`.	■ Accept ■ Content-Length ■ Connection ■ Host ■ User-Agent	■ Content-Length ■ Date ■ Transfer-Encoding
JMS	Can only be set when the request is with respect to a one-way service or a request/response service based on `JMSMessageID` correlation. If sending to a request/response service based on `JMSCorrelationID` correlation, these headers are overwritten at runtime.	JMSCorrelationID	JMSCorrelationID
JMS	Should be set to the message time-to-live in milliseconds. The resulting value in the message received is the sum of the time-to-live value specified by the client and the GMT at the time of the send or publish[1].	JMSExpiration	JMSExpiration
JMS	The Oracle Service Bus runtime sets these headers. In other words, any specifications you make for these headers at design time are overwritten at runtime.	■ JMSMessageID ■ JMSRedelivered ■ JMSTimestamp ■ JMSXDeliveryCount ■ JMSXUserID ■ JMS_IBM_PutDate[2] ■ JMS_IBM_PutTime 2 ■ JMS_IBM_PutApplType 2 ■ JMS_IBM_Encoding2 ■ JMS_IBM_Character_Set 2	■ JMSMessageID ■ JMSRedelivered ■ JMSTimestamp ■ JMSXDeliveryCount ■ JMSXUserID ■ JMS_IBM_PutDate 2 ■ JMS_IBM_PutTime 2 ■ JMS_IBM_PutApplType 2 ■ JMS_IBM_Encoding 2 ■ JMS_IBM_Character_Set 2

Table 37–8 (Cont.) Limitations to Transport Header Values You Specify in Transport Header Actions

Transport	Description of Limitation...	Outbound Request Header	Inbound Response Header
JMS	Because IBM MQ does not allow certain properties to be set by a client application, if you set these headers with respect to an IBM MQ destination, a runtime exception is raised.	■ JMSXDeliveryCount ■ JMSXUserID ■ JMSXAppID	■ JMSXDeliveryCount ■ JMSXUserID ■ JMSXAppID
JMS	These headers cannot be deleted when the **Pass all Headers through Pipeline** option is also specified.	■ JMSDeliveryMode ■ JMSExpiration ■ JMSMessageID ■ JMSRedelivered ■ JMSTimestamp ■ JMSXDeliveryCount	■ JMSDeliveryMode ■ JMSExpiration ■ JMSMessageID ■ JMSRedelivered ■ JMSTimestamp ■ JMSXDeliveryCount ■ JMSCorelationID—if the inbound message has the correlation ID set. For example, if the inbound response comes from a registered JMS business service
FTP and File	No limitations. In other words you can set or delete the header(s)[3] for File and FTP transports and your specifications are honored by the Oracle Service Bus runtime.	N/A	N/A
Email	The Oracle Service Bus runtime sets these headers. In other words, any specifications you make for these headers at design time are overwritten at runtime.	Content-Type	Content-Type
Email	Oracle Service Bus does not use these headers in outbound requests. If you set them dynamically (that is, if you set them in the $outbound headers section), Oracle Service Bus ignores them. These headers are received in $inbound. Date is the time the mail was sent by the sender. From is retrieved from incoming mail headers.	■ From (Name) ■ Date	N/A

[1] For example, if you set the JMSExpiration header to 1000, and at the time of the send, GMT is 1,000,000 (as a result of System.currentTimeMillis()), the resulting value of the JMSExpiration property in the JMS message is 1,000,1000

[2] Header names with the JMS_IBM prefix are to be used with respect to destinations hosted by an IBM MQ server

[3] For FTP and file proxies, there is an transport request header 'fileName'. The value of this request header is the name of the file being polled.

> **Note:** The same limitations around setting certain transport headers and metadata are true when you set the inbound and outbound context variables, and when you use the Oracle Service Bus Test Console to test your proxy or business services.

37.4 Performing Transformations in Message Flows

Transformation maps describe the mapping between two data types. Oracle Service Bus supports data mapping that uses the XQuery and the eXtensible Stylesheet Language Transformation (XSLT) standards. XSLT maps describe XML-to-XML mappings. XQuery maps can describe XML-to-XML, XML to non-XML, and non-XML to XML mappings.

The point in a message flow at which you specify a transformation depends on whether:

- The message format relies on target services—that is, the message format must be in a format acceptable by the route destination. This applies when the transformation is performed in a route node or in one of the publish actions.

  Publish actions identify a target service for a message and configure how the message is packaged and sent to that service. Oracle Service Bus also provides publish table actions. A publish table action consists of a set of routes wrapped in a switch-style condition table. It is a shorthand construct that allows different routes to be selected, based upon the results of a single XQuery expression.

- You perform the transformation on the response or request message regardless of the route destination. In this case, you can configure the transformations in the request or response pipeline stages.

37.4.1 Transformations and Publish Actions

When transformations are designed in publish actions, the transformations have a local copy of the $outbound variable and message-related variables ($header, $body, and $attachments). Any changes you make to an outbound message in a publish action affect only the published message. In other words, the changes you make in the publish action are rolled back before the message flow proceeds to any actions that follow the publish action in your message flow.

For example, consider a message flow that deals with a large purchase order, and you have to send the summary of the purchase order, through email, to the manager. The summary of the of the purchase order is created in the SOAP body of the incoming message when you include a publish action in the request pipeline. In the publish action, the purchase order data is transformed into a summary of the purchase order—for example, all the attachments in $attachments can be deleted because they are not required in the summary of the purchase order. After the publish action, the message in its state prior to the publish action continues through the message flow, as described in the following section.

37.4.1.1 Publish Action Behavior with Quality of Service

This section describes how the publish action behaves with different quality of service (QoS) settings.

Exactly-Once – When QoS is exactly-once, the publish action waits (blocking call) until the response from the target service is available, although the response itself is discarded. When the target is a business service, the publish action waits until the business service response is available. When the target is a proxy service, the publish action waits until the proxy service's response pipeline completes.

Best-Effort – When QoS is best-effort and the target service is a one-way proxy service or a one-way business service with retry count > 0, the publish action waits until the target service returns. With a one-way target service there is no response, but the publish action waits until the request is delivered.

If the target proxy or business service is request-response or a one-way business service with retry count = 0, the publish action does not wait for the response (non-blocking call).

37.4.2 Transformations and Route Nodes

You may need to route messages to one of two destinations, based on a WS-addressing header. In that case, content-based routing and the second destination require the newer version of the document in the SOAP body. In this situation, you can configure the route node to conditionally route to one of the two destinations. You can configure a transformation in the route node to transform the document for the second destination.

You can also set the control elements in the outbound context variable (`$outbound`) to influence the behavior of the system for the outbound message (for example, you can set the Quality of Service).

See Section 39.4, "Inbound and Outbound Variables" and Section 39.9, "Constructing Messages to Dispatch" for information about the sub-elements of the inbound and outbound variables and how the content of messages is constructed using the values of the variables in the message context.

See Also

- Chapter 39, "Message Context"

- Chapter 11, "XQuery Transformations" and Chapter 15, "XSL Transformations"

- "XQuery Mapper" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

37.5 Constructing Service Callout Messages

When Oracle Service Bus makes a call to a service through a service callout action, the content of the message is constructed using the values of variables in the message context. The message content for outbound messages is handled differently depending upon the type of the target service.

How the message content is created depends on the type of the target service and whether you choose to configure the SOAP body or the payload (parameters or document), as described in the following topics:

- Section 37.5.1, "SOAP Document Style Services"

- Section 37.5.2, "SOAP RPC Style Services"

- Section 37.5.3, "XML Services"

- Section 37.5.4, "Messaging Services"

37.5.1 SOAP Document Style Services

Messages for SOAP Document Style services (including EJB document and document-wrapped services), can be constructed as follows:

- The variable assigned for the request document contains the SOAP body.

- The variable assigned for the SOAP request header contains the SOAP header.

- The response must be a single XML document—it is the content of the SOAP body plus the SOAP header (if specified)

To illustrate how messages are constructed during callouts to SOAP Document Style services, consider a service callout action configured as follows:

- **Request Document Variable**: `myreq`

- **Response Document Variable**: `myresp`

- **SOAP Request Header**: `reqheader`

- **SOAP Response Header**: `respheader`

Assume also that at runtime, the request document variable, `myreq`, is bound to the following XML.

Example 37–1   Content of Request Variable (myreq)

```
<sayHello xmlns="http://www.openuri.org/">
    <intVal>100</intVal>
    <string>Hello Oracle</string>
</sayHello>
```

At runtime, the SOAP request header variable, `reqheader`, is bound to the following SOAP header.

Example 37–2   Content of SOAP Request Header Variable (reqheader)

```
<soap:Header xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing">
    <wsa:Action>...</wsa:Action>
    <wsa:To>...</wsa:To>
    <wsa:From>...</wsa:From>
    <wsa:ReplyTo>...</wsa:ReplyTo>
    <wsa:FaultTo>...</wsa:FaultTo>
  </soap:Header>
```

In this example scenario, the full body of the message sent to the external service is shown in Example 37–3 (the contents of the `myreq` and `reqheader` variables are shown in bold).

Example 37–3   Message Sent to the Service as a Result of Service Callout Action

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing">
        <wsa:Action>...</wsa:Action>
        <wsa:To>...</wsa:To>
        <wsa:From>...</wsa:From>
        <wsa:ReplyTo>...</wsa:ReplyTo>
        <wsa:FaultTo>...</wsa:FaultTo>
    </soap:Header>
    <soapenv:Body>
        <sayHello xmlns="http://www.openuri.org/">
            <intVal>100</intVal>
            <string>Hello Oracle</string>
        </sayHello>
    </soapenv:Body>
</soapenv:Envelope>
```

Based on the configuration of the service callout action described above, the response from the service is assigned to the `myresp` variable. The full response from the external service is shown in Example 37–4.

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.xmlsoap.
org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <env:Header/>
    <env:Body env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <m:sayHelloResponse xmlns:m="http://www.openuri.org/">
            <result xsi:type="xsd:string">This message brought to you by Hello
Oracle and the number 100
            </result>
        </m:sayHelloResponse>
    </env:Body>
</env:Envelope>
```

In this scenario, the `myresp` variable is assigned the value shown in Example 37–5.

Example 37–5   Content of Response Variable (myresp) as a Result of Service Callout Action

```
<m:sayHelloResponse xmlns:m="http://www.openuri.org/">
    <result ns0:type="xsd:string"
xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance">
This message brought to you by Hello Oracle and the number 100
    </result>
</m:sayHelloResponse>
```

37.5.2 SOAP RPC Style Services

Messages for SOAP RPC Style services (including EJB RPC services) can be constructed as follows:

- Request messages are assembled from message context variables.

  - The SOAP body is built based on the SOAP RPC format (operation wrapper, parameter wrappers, and so on).

  - The SOAP header is the content of the variable specified for the SOAP request header, if one is specified.

  - Part as element—the parameter value is the variable content.

  - Part as simple type—the parameter value is the string representation of the variable content.

  - Part as complex type—the parameter corresponds to renaming the root of the variable content after the parameter name.

- Response messages are assembled as follows:

  - The output content is the content of SOAP header, if a SOAP header is specified.

  - Part as element—the output content is the child element of the parameter; there is at most one child element.

  - Part as simple/complex type—the output content is the parameter itself.

To illustrate how messages are constructed during callouts to SOAP RPC Style services, look at this example with the following configuration:

- A message context variable `input1` is bound to a value 100.

- A message context variable `input2` is bound to a string value: `Hello Oracle`.

- A service callout action configured as follows:

  - **Request Parameter `intval`:** input1

  - **Request Parameter `string`:** input2

  - **Response Parameter `result`:** output1

In this scenario, the body of the outbound message to the service is shown in Example 37–6.

Example 37–6   Content of Outbound Message

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <sayHello2 xmlns="http://www.openuri.org/">
            <intVal>100</intVal>
            <string >Hello Oracle</string>
        </sayHello2>
    </soapenv:Body>
</soapenv:Envelope>
```

The response returned by the service to which the call was made is shown in Example 37–7.

Example 37–7   Content of Response Message From the helloWorld Service

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <env:Header/>
    <env:Body env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <m:sayHello2Response xmlns:m="http://www.openuri.org/">
            <result xsi:type="n1:HelloWorldResult" xmlns:n1="java:">
                <message xsi:type="xsd:string">
                This message brought to you by Hello Oracle and the number 100
                </message>
            </result>
        </m:sayHello2Response>
    </env:Body>
</env:Envelope>
```

The message context variable `output1` is assigned the value shown in Example 37–8.

Example 37–8   Content of Output Variable (output1)

```
<message ns0:type="xsd:string"
xmlns:ns0="http://www.w3.org/2001/XMLSchema-intance">
This message brought to you by Hello Oracle and the number 100</message>
```

37.5.3 XML Services

Messages for XML services can be constructed as follows:

- The request message is the content of the variable assigned for the request document.

- The content of the request variable must be a single XML document.

- The output document is the response message.

To illustrate how messages are constructed during callouts to XML services, take for example a service callout action configured as follows:

- **Request Document Variable**: `myreq`

- **Response Document Variable**: `myresp`

Assume also that at runtime, the request document variable, `myreq`, is bound to the following XML.

Example 37–9   Content of myreq Variable

```
<sayHello xmlns="http://www.openuri.org/">
    <intVal>100</intVal>
    <string>Hello Oracle</string>
</sayHello>
```

In this scenario:

- The outbound message payload is the value of the `myreq` variable, as shown in Table 37–9.

- The response and the value assigned to the message context variable, `myresp`, is shown in Example 37–10.

Example 37–10   Content of myresp Variable

```
<m:sayHelloResponse xmlns:m="http://www.openuri.org/">
     <result xsi:type="xsd:string">This message brought to you by Hello Oracle and
the number 100
     </result>
</m:sayHelloResponse>
```

37.5.4 Messaging Services

In the case of Messaging services:

- The request message is the content of the request variable. The content can be simple text, XML, or binary data represented by an instance of `<binary-content ref=.../>` reference XML.

- Response messages are treated as binary, so the response variable will contain an instance of `<binary-content ref= ... />` reference XML, regardless of the actual content received.

For example, if the request message context variable myreq is bound to an XML document of the following format: `<hello>there</hello>`, the outbound message contains exactly this payload. The response message context variable (myresp) is bound to a reference element similar to the following:

```
<binary-content ref=" cid:1850733759955566502-2ca29e5c.1079b180f61.-7fd8"/>
```

37.6 Handling Errors as the Result of a Service Callout

You can configure error handling at the message flow, pipeline, route node, and stage level. The types of errors that are received from an external service as the result of a

service callout include transport errors, SOAP faults, responses that do not conform to an expected response, and so on.

The `fault` context variable is set differently for each type of error returned. You can build your business and error handling logic based on the content of the `fault` variable. To learn more about `$fault`, see Section 39.6, "Fault Variable."

37.6.1 Transport Errors

When a transport error is received from an external service and there is no error response payload returned to Oracle Service Bus by the transport provider (for example, if an HTTP business service accepts response types other than XML or SOAP and therefore cannot receive HTTP response codes), the service callout action throws an exception, which in turn causes the pipeline to raise an error. The fault variable in a user-configured error handler is bound to a message formatted similarly to that shown in Example 37–11.

Example 37–11   Contents of the Oracle Service Bus fault Variable—Transport Error, no Error Response Payload

```
<con:fault xmlns:con="http://www.bea.com/wli/sb/context">
  <con:errorCode>BEA-380000</con:errorCode>
  <con:reason>Not Found</con:reason>
   <con:details>
   .......
   </con:details>
    <con:location>
       <con:node>PipelinePairNode1</con:node>
       <con:Pipeline>PipelinePairNode1_request</con:Pipeline>
    <con:Stage>Stage1</con:Stage>
    </con:location>
</con:fault>
```

In the case that there is a payload associated with the transport error—for example, when an HTTP 500 error code is received from the business service and there is XML payload in the response—a message context fault is generated with the custom error code: BEA-382502.

The following conditions must be met for a BEA-382502 error response code to be triggered as the result of a response from a service—when that service uses an HTTP or JMS transport:

- (HTTP) The response code must be any code 300 or greater.

- (JMS) The response must have a property set to indicate that it is an error response—the transport metadata status code set to1 indicates an error.

- The content type must be text/xml, application/*any_string*+xml, or multipart/related.

- If the service is AnySoap or WSDL-based SOAP, then it must have a SOAP envelope. The body inside the SOAP envelope must be XML format; it cannot be text.

- If the service type is AnyXML, or a messaging service of type text returns XML content with a non-successful response code (any code other than 200 or 202).

If the transport is HTTP, the `ErrorResponseDetail` element will also contain the HTTP error code returned with the response. The `ErrorResponseDetail` element in the fault contains error response payload received from the service. Example 37–12 shows an example of the `ErrorResponseDetail` element.

Example 37–12   Contents of the Oracle Service Bus fault Variable—Transport Error, with Error Response Payload

```
<ctx:Fault xmlns:ctx="http://www.bea.com/wli/sb/context">
    <ctx:errorCode>BEA-382502<ctx:errorCode>
    <ctx:reason> Service callout has received an error response from the
server</ctx:reason>
    <ctx:details>
        <alsb:ErrorResponseDetail xmlns:alsb="http://www.oracle.com/...">
            <alsb:detail> <![CDATA[
. . .
]]>
            </alsb:detail>
<alsb:http-response-code>500</alsb:http-response-code>
        </alsb:ErrorResponseDetail>
    </ctx:details>
    <ctx:location>. . .</ctx:location>
</ctx:Fault>
```

> **Note:** The XML schema for the service callout-generated fault is shown in Example 37–15.

37.6.2 SOAP Faults

In case an external service returns a SOAP fault, the Oracle Service Bus runtime sets up the context variable `$fault` with a custom error code and description with the details of the fault. To do so, the contents of the 3 elements under the `<SOAP-ENV:Fault>` element in the SOAP fault are extracted and used to construct an Oracle Service Bus fault element.

Take for example a scenario in which a service returns the following error.

Example 37–13   SOAP Fault Returned From Service Callout

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Application Error</faultstring>
      <detail>
        <message>That's an Error!</message>
        <errorcode>1006</errorcode>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The `<faultcode>`, `<faultstring>`, and `<detail>` elements are extracted and wrapped in an `<alsb:ReceivedFault>` element. The `faultcode` element in Example 37–15 contains a QName—any related namespace declarations are preserved. If the transport is HTTP, the `ReceivedFault` element will also contain the HTTP error code returned with the fault response.

The generated `<alsb:ReceivedFault>` element, along with the custom error code and the error string are used to construct the contents of the `fault` context variable, which in this example takes a format similar to that shown in Example 37–14.

Example 37–14   Contents of the Oracle Service Bus Fault Variable—SOAP Fault

```
<ctx:Fault xmlns:ctx="http://www.bea.com/wli/sb/context">
    <ctx:errorCode>BEA-382500<ctx:errorCode>
    <ctx:reason> service callout received a soap Fault response</ctx:reason>
    <ctx:details>
        <alsb:ReceivedFault xmlns:alsb="http://www.oracle.com/...">
            <alsb:faultcode
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">SOAP-ENV:Clien
            </alsb:faultcode>
            <alsb:faultstring>Application Error</alsb:faultstring>
            <alsb:detail>
                <message>That's an Error!</message>
                <errorcode>1006</errorcode>
            </alsb:detail>
          <alsb:http-response-code>500</alsb:http-response-code>
        </alsb:ReceivedFault>
    </ctx:details>
    <ctx:location> </ctx:location>
</ctx:Fault>
```

> **Note:** The unique error code BEA-382500 is reserved for the case
> when service callout actions receive SOAP fault responses.

When chaining local proxy services, SOAP fault details are not propagated from one pipeline to the next in the $fault variable. To propagate SOAP faults from proxy service to proxy service, use an error handler with a Reply with Failure action, as described in "Propagating SOAP Faults Between Proxy Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

37.6.3 Unexpected Responses

When a service returns a response message that is not what the proxy service runtime expects, a message context fault will be generated and initialized with the custom error code BEA-382501. The details of the fault include the contents of the SOAP-Body element of the response. If the transport is HTTP, the `ReceivedFault` element will also contain the HTTP error code returned with the fault response.

The XML schema definition of the service callout-generated fault details is shown in Example 37–15.

Example 37–15   XML Schema for the Service Callout-Generated Fault Details

```
<xs:complexType name="ReceivedFaultDetail">
    <xs:sequence>
        <xs:element name="faultcode" type="xs:QName"/>
        <xs:element name="faultstring" type="xs:string"/>
        <xs:element name="detail" minOccurs="0" >
            <xs:complexType>
              <xs:sequence>
                    <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" />
              </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax" />
            </xs:complexType>
        </xs:element>
        <xs:element name="http-response-code" type="xs:int" minOccurs="0"/>\
type="xs:int" minOccurs="0"/>
```

```
                </xs:sequence>
</xs:complexType>

<xs:complexType name="UnrecognizedResponseDetail">
        <xs:sequence>
            <xs:element name="detail" minOccurs="0" type="xs:string" />
        </xs:sequence>
</xs:complexType>

<xs:complexType name="ErrorResponseDetail">
        <xs:sequence>
            <xs:element name="detail" minOccurs="0" type="xs:string" />
        </xs:sequence>
</xs:complexType>
```

37.7 Handling Errors in Message Flows

Error handling can be defined at multiple levels, including the pipeline stage, the pipeline (request or response), or for the entire proxy service.

When an error occurs, the error handler at the stage level is invoked. If the stage-level error handler is not able to handle the error, the pipeline error handler is invoked. If the pipeline-level error handler also fails to handle the error, the service-level error handler is invoked. If the service-level error handler also fails, the error is handled by the system.

You define how errors are handled at the stage, pipeline, and proxy service levels by adding an action to the message flow. Typical error handling actions include Raise Error, Skip, Reply (with success), and Reply (with failure). For more information on defining error messages and error handling, see Chapter 24, "Proxy Services: Error Handlers." For more information about the actions you can specify, see Section 21, "Proxy Services: Actions."

Table 37–9 summarizes the scope of the error handlers at various levels in the message flow.

Table 37–9   Scope of Error Handlers

Level	Scope
Stage	Handles all the errors within a stage.
Pipeline	Handles all the errors in a pipeline, along with any unhandled errors from any stage in a pipeline.
Service	Handles all the errors in a proxy service, along with any unhandled errors in any pipeline in a service. **Note**: All WS-Security errors are handled at this level.
System	Handles all the errors that are not handled any where else in a pipeline.

> **Note:** There are exceptions to the scope of error handlers. For example, an exception thrown by a non-XML transformation at the stage level is only caught by the service-level error handler. Suppose a transformation occurs that transforms XML to MFL for an outgoing proxy service response message, it always occurs in the binding layer. As an example, if a non-XML output is missing a mandatory field at the stage level, only a service-level error handler can catch this error.

You can handle errors by configuring a test that checks if an assertion is true and use the reply action configured false. You can repeat this test at various levels. You can also have an error without an error handler at a lower level and handle it through an error handler at an higher level in message flow.

In general, it is easier to handle specific errors at a stage level of the message flow and use error handlers at the higher level for more general default processing of errors that are not handled at the lower levels. It is good practice to explicitly handle anticipated errors in the pipelines and allow the service-level handler to handle unanticipated errors.

> **Note:** You can only handle WS-Security related errors at the service level.

37.7.1 Generating the Error Message, Reporting, and Replying

A predefined context variable (the `fault` variable) is used to hold information about any error that occurs during message processing. When an error occurs, this variable is populated with information before the appropriate error handler is invoked. The `fault` variable is defined only in error handler pipelines and is not set in request and response pipelines, or in route or branch nodes. For additional information about `$fault`, see Section 39.2, "Predefined Context Variables."

In the event of errors for request/response type inbound messages, it is often necessary to send a message back to the originator outlining the reason why an error occurred. You can accomplish this by using a *Reply with Failure* action after configuring the message context variables with the response you want to send. For example, when an HTTP message fails, *Reply with Failure* generates the `HTTP 500` status. When a JMS message fails, *Reply with Failure* sets the `JMS_BEA_Error` property to true. The Oracle Service Bus error actions are discussed in Chapter 24, "Proxy Services: Error Handlers."

An error handling pipeline is invoked if a service invoked by a proxy service returns a SOAP fault or transport error. Any received SOAP fault is stored in `$body`, so if a *Reply with Failure* is executed without modifying `$body`, the original SOAP fault is returned to the client that invoked the service. If a reply action is not configured, the system error handler generates a new SOAP fault message. The proxy service recognizes that a SOAP fault is returned because an HTTP error status is set, or the JMS property `SERVER_Error` is set to true.

Some use cases require error reporting. You can use the report action in these situations. For example, consider a scenario in which the request pipeline reports a message for tracking purposes, but the service invoked by the route node fails after the reporting action. In this case, the reporting system logged the message, but there is no guarantee that the message was processed successfully, only that the message was successfully received.

You can use the Oracle Service Bus Administration Console to track the message to obtain an accurate picture of the message flow. This lets you view the original reported message indicating the message was submitted for processing, and also the subsequent reported error indicating that the message was not processed correctly. To learn how to configure a report action and use the data reported at runtime, see Chapter 21, "Proxy Services: Actions."

37.7.2 Example of Action Configuration in Error Handlers

This example shows how you can configure the report and reply actions in error handlers. The message flow shown in Figure 37–2 includes an error handler on the

`validate loan application` stage. The error handler in this case is a simple message flow with a single stage configured—it is represented in the Oracle Service Bus Administration Console as shown in Figure 37–2.

Figure 37–2   Error Handler Message Flow
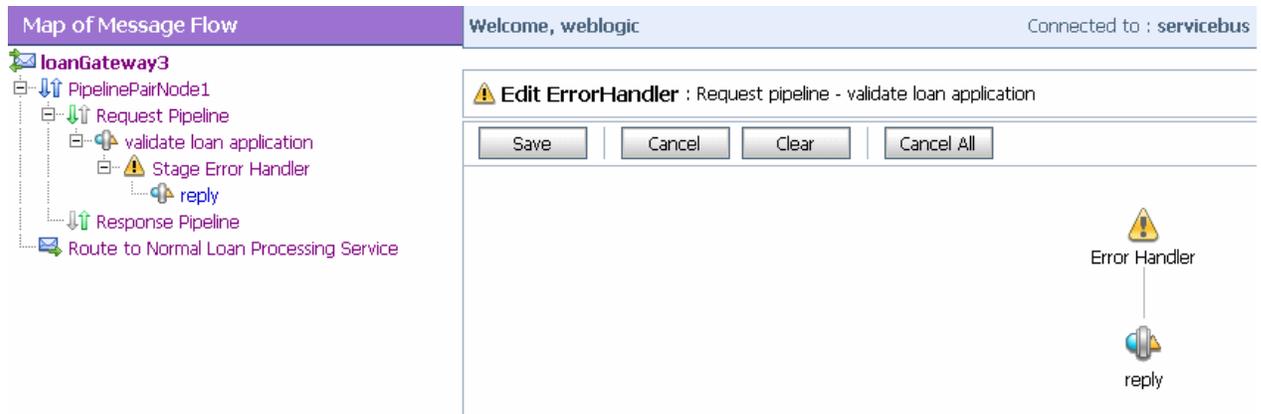
The stage is, in turn, configured with actions (replace, report, and reply) as shown in Figure 37–3.

Figure 37–3   Actions in Stage Error Handler

The actions control the behavior of the stage in the pipeline error handler as follows:

- **Replace**—The contents of a specified element of the body variable are replaced with the contents of the `fault` context variable. The body variable element is specified by an XPath expression. The contents are replaced with the value returned by an XQuery expression—in this case `$fault/ctx:reason/text()`

- **Report**— Messages from the reporting action are written to the Oracle Service Bus Reporting Data Stream if the error handler configured with this action is invoked. The JMS Reporting Provider reports the messages on the Oracle Service Bus Dashboard. Oracle Service Bus provides the capability to deliver message data to one or more reporting providers. Message data is captured from the body of the

message and from any other variables associated with the message, such as header or inbound variables. You can use the message delivered to the reporting provider for functions such as tracking messages or regulatory auditing.

When an error occurs, the contents of the fault context variable are reported. The key name is `errorCode`, and the key value is extracted from the fault variable using the following XPath expression: `./ctx:errorCode`. Key/value pairs are the key identifiers that identify these messages in the Dashboard at runtime.

To configure a report action and use the data reported at runtime, see Chapter 21, "Proxy Services: Actions."

- **Reply**— At runtime, an immediate reply is sent to the invoker of the loanGateway3 proxy service (see Figure 37–3) indicating that the message had a fault The reply is `With Failure`.

For configuration information, see Chapter 24, "Proxy Services: Error Handlers."

37.8 Using Dynamic Routing

When you do not know the service you need to invoke from the proxy service you are creating, you can use dynamic routing.

For any given proxy service, you can use one of the following techniques to dynamically route messages:

- In a message flow pipeline, design an XQuery expression to dynamically set the fully qualified service name in Oracle Service Bus and use the dynamic route or dynamic publish actions.

  > **Note:** Dynamic Routing can be achieved in a route node, whereas dynamic publishing can be achieved in a stage in a request pipeline or a response pipeline.

  With this technique, the proxy service dynamically uses the service account of the endpoint business service to send user names and passwords in its outbound requests. For example, if a proxy service is routing a request to Business Service A, then the proxy service uses the service account from Business Service A to send user names and passwords in its outbound request. See Section 37.8.1, "Implementing Dynamic Routing."

- Configure a proxy service to route or publish messages to a business service. Then, in the request actions section for the route action or publish action, add a Routing Options action that dynamically specifies the URI of a service.

  With this technique, to send user names and passwords in its outbound requests, the proxy service uses the service account of the statically defined business service, regardless of the URI to which the request is actually sent.

  For information on how to use this technique, see Section 37.8.1, "Implementing Dynamic Routing."

  > **Note:** This technique is used when the overview of the interface is fixed. The overview of the interface includes message types, port types, and binding, and excludes the concrete interface. The concrete interface is the transport URL at which the service is located.

For a working example of dynamic service invocation, see the Oracle Service Bus samples at
http://www.oracle.com/technetwork/middleware/service-bus/learnmore/index.html.

37.8.1 Implementing Dynamic Routing

You can use dynamic routing to determine the destination during the runtime of a proxy service. To achieve this you can use a routing table in an XML file to create an XQuery resource.

> **Note:** Instead of using the XQuery resource, you can also directly use the XML file from which the resource is created.

An XML file or the XQuery resource can be maintained easily. At runtime you provide the entry in the routing table that will determine the routing or publishing destination of the proxy service.The XML file or the XQuery resource contains a routing table, which maps a logical identifier (such as the name of a company) to the physical identifier (the fully qualified name of the service in Oracle Service Bus). The logical identifier, which is extracted from the message, maps on to the physical identifier, which is the name of the service you want to invoke.

> **Note:** To use the dynamic route action, you need the fully qualified name of the service in Oracle Service Bus.

In a pipeline the logical identifier is obtained with an XPath into the message.You assign the XML table in the XQuery resource to a variable. You implement a query against the variable in the routing table to extract the physical identifier based on the corresponding logical identifier. Using this variable you will be able to invoke the required service. The following sections describe how to implement dynamic routing.

- Section 37.8.1.1, "Sample XML File"
- Section 37.8.1.2, "Creating an XQuery Resource From the Sample XML"
- Section 37.8.1.3, "Creating and Configuring the Proxy Service to Implement Dynamic Routing"
- Section 37.8.1.4, "Guidelines for Implementing Identity-Based Routing"

37.8.1.1 Sample XML File

You can create an XQuery resource from the following XML file. Save this as sampleXquery.xml.

Example 37–16   Sample XML File

```
<routing>
    <row>
        <logical>Oracle</logical>
        <physical>default/goldservice</physical>
    </row>
    <row>
        <logical>ABC Corp</logical>
        <physical>default/silverservice</physical>
    </row>
```

```
</routing>
```

37.8.1.2 Creating an XQuery Resource From the Sample XML

To create an XQuery resource from the sample XML:

1. In an active session, select **Project Explorer** from the left navigation panel. The Project View page is displayed.

2. Select the project to which you want to add the XQuery resource.

3. In the **Project View** page, select the XQuery resource from the **Select Resource Type** list. The Create XQuery page is displayed.

4. In the **Resource Name** field, enter the name of the resource. This is a mandatory.

5. In the **Resource Description** field, provide the a description for the resource. This is optional.

6. In the XQuery field, provide the path to the XML you are using as an XQuery resource. Click **Browse** to locate the file. Optionally, you can copy and paste the XML in the XQuery field. This is mandatory.

7. Save the XQuery resource.

8. Activate the session.

37.8.1.3 Creating and Configuring the Proxy Service to Implement Dynamic Routing

To implement dynamic routing with a proxy service:

1. In an active session, select **Project Explorer** from the left navigation panel. The Project View page is displayed.

2. Select the project to which you want to add the proxy service.

3. In the **Project View** page, select the **Proxy Service** resource from the **Select Resource Type** list. The General Configuration page is displayed.

4. In the **Service Name** field of the **General Configuration** page, enter the name of the proxy service. This is mandatory.

5. To select the type of service, click the button adjacent to various types of services available under **Service Type**. For more information on selecting the service type, see Chapter 21, "Proxy Services: Actions."

6. Click **Finish**. On the **Summary** page, click **Save** to save the proxy service.

7. On the **Project View** page, click the Edit Message Flow icon against the newly created proxy service in the **Resource** table. The **Edit Message Flow** page is displayed.

8. Click the message flow to add a pipeline pair to the message flow.

9. Click **Request Pipeline** icon select **Add Stage** from the menu.

10. Click the Stage1 icon to and select **Edit Stage** from the menu. The **Edit Stage Configuration** page appears.

11. Click **Add Action** icon. Choose **Add an Action** item from the menu.

12. Choose the **Assign** action from **Message Processing**.

13. Click **Expression**. The **XQuery Expression Editor** is displayed.

14. Click **XQuery Resources**. The browser displays the page where you can import the XQuery resource. Click **Browse** to locate the XQuery resource.

15. Click **Validate** to validate the imported XQuery resource.

16. Save the imported XQuery resource on successful validation.

17. On the **Edit Stage Configuration** page, enter the name of the variable in the field.

This assigns the XQuery resource to this variable. The variable now contains the externalized routing table.

18. Add another **Assign** action.

19. Enter the following XQuery:

```
<ctx: route>
<ctx: service isProxy='false'>
{$routingtable/row[logical/text()=$logicalidentifier]/physical/text()}
</ctx: service>
</ctx: route>
```

In the above code, replace `$logicalidentifier` by the actual XPath to extract the logical identifier from the message (example from `$body`).

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

21. Save the XQuery on successful validation.

22. On the **Edit Stage Configuration** page, enter the name of the variable (for example, *routeresult*) in the field.

This extracts the XML used by the dynamic route action into this variable.

23. Click the message flow to add a route node to the end of the message flow.

24. Click the **Route Node** icon and select **Edit** from the menu.

25. Click the **Add Action** icon. Choose **Add an Action** item from the menu.

26. Choose the **Dynamic Route** action.

27. Click **Expression**. The XQuery Expression Editor is displayed.

28. Enter the variable; for example, *$routeresult*.

37.8.1.4 Guidelines for Implementing Identity-Based Routing

To enable dynamic message routing based on the identity of an authenticated user, Oracle Service Bus stores information such as username, group membership (/principals/group), and the name of the subject (/subject-properties/property/name in the following inbound context variables:

■ $inbound/ctx:security/ctx:transportClient/*

■ $inbound/ctx:security/ctx:messageLevelClient/*

For more information on these context variables, see Table 39–5, " Sub-Elements of the Security Element".

Using the guidance provided in Section 37.8, "Using Dynamic Routing," use XQuery or simple XML to map authenticated user identity characteristics to different endpoints using the desired mapping technique.

The following predefined OSB XQuery functions are also available to perform security checks in identity-based routing:

■ Section 43.2.1, "fn-bea:lookupBasicCredentials"

- Section 43.2.2, "fn-bea:isUserInGroup"

- Section 43.2.3, "fn-bea:isUserInRole"

For a working example of dynamic service invocation, see the Oracle Service Bus samples at
http://www.oracle.com/technetwork/middleware/service-bus/learnmore/index.html.

37.9 Accessing Databases Using XQuery

Oracle Service Bus provides read-access to databases from proxy services without requiring you to write a custom EJB or custom Java code and without the need for a separate database product like Oracle Data Service Integrator. You can use the `execute-sql()` function to make a simple JDBC call to a database to perform simple database reads. Any SQL query is legal, from a query that gets a single tax rate for the supplied location to a query that does a complex join to obtain an order's current status from several underlying database tables.

A database query can be used to get data for message enrichment, for routing decisions, or for customizing the behavior of a proxy service. Take for example a scenario in which an Oracle Service Bus proxy service receives "request for quote" messages. The proxy service can route the requests based on the customer's priority to one of a number of quotation business services (say, standard, gold, or platinum level services). The proxy service can then perform a SQL-based augmentation of the results that those services return—for example, based on the selected ship method and the weight of the order, the shipping cost can be looked up and that cost added to the request for quote message.

Section 43.2.5, "fn-bea:execute-sql()" describes the syntax for the function and provides examples of its use. The `execute-sql()` function returns typed data and automatically translates values between SQL/JDBC and XQuery data models.

You can store the returned element in a user-defined variable in an Oracle Service Bus message flow.

The following databases and JDBC drivers are supported using the `execute-sql()` function:

- The sample database provided by Oracle WebLogic Server.

  IBM DB2/NT 8

- Microsoft SQL Server 2000, 2005

- Oracle8*i*, Release 8.1.x

- Oracle9*i*, Oracle Database 10*g*

- Sybase Adaptive Server 12.5.2 and 12.5.3

- Oracle WebLogic Server Type 4 JDBC drivers

- Third-party drivers supported by Oracle WebLogic Server

Use non-XA drivers for datasources you use with the Section 43.2.5, "fn-bea:execute-sql()" function—the function supports read-only access to the datasources.

> **Caution:** In addition to specifying a non-XA JDBC driver class to use to connect to the database, you must ensure that you disable global transactions and two-phase commit. (Global transactions are enabled by default in the Oracle WebLogic Server Administration Console for JDBC data sources.) These specifications can be made for your data source through the Oracle WebLogic Server Administration Console. See "Create JDBC Data Sources" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

For complete information about database and JDBC drivers support in Oracle Service Bus, see *Oracle Fusion Middleware Supported System Configurations* at:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

Databases other than the core set described in the preceding listing are also supported. However, for the core databases listed above, the XQuery engine does a better recognition and mapping of data types to XQuery types than it does for the non-core databases—in some cases, a core database's proprietary JDBC extensions are used when fetching data. For the non-core databases, the XQuery engine relies totally on the standard type codes provided by the JDBC driver and standard JDBC resultset access methods.

When designing your proxy service, you can enter XQueries inline as part of an action definition instead of entering them as resources. You can also use inline XQueries for conditions in If...Then... actions in message flows. For information about using the inline XQuery editor, see Section 37.11.3, "Creating Variable Structure Mappings."

37.10 Understanding Message Context

The message context is a set of variables that hold message context and information about messages as they are routed through the Oracle Service Bus. Together, the `header`, `body`, and `attachments` variables, (referenced as `$header`, `$body` and `$attachments` in XQuery statements) represent the message as it flows through Oracle Service Bus. The canonical form of the message is SOAP. Even if the service type is not SOAP, the message appears as SOAP in the Oracle Service Bus message context.

37.10.1 Message Context Components

In a Message Context, `$header` contains a SOAP header element and `$body` contains a SOAP Body element. The Header and Body elements are qualified by the SOAP 1.1 or SOAP 1.2 namespace depending on the service type of the proxy service. Also in a Message Context, `$attachments` contains a wrapper element called attachments with one child attachment element per attachment. The attachment element has a body element with the actual attachment.

When a message is received by a proxy service, the message contents are used to initialize the header, body, and attachments variables. For SOAP services, the Header and Body elements are taken directly from the envelope of the received SOAP message and assigned to `$header` and `$body` respectively. For non-SOAP services, the entire content of the message is typically wrapped in a Body element (qualified by the SOAP 1.1 namespace) and assigned to `$body`, and an empty Header element (qualified by the SOAP 1.1 namespace) is assigned to `$header`.

Binary and MFL messages are initialized differently. For MFL messages, the equivalent XML document is inserted into the Body element that is assigned to `$body`. For binary

messages, the message data is stored internally and a piece of reference XML is inserted into the Body element that is assigned to `$body`. The reference XML looks like `<binary-content ref="..."/>`, where `"..."` contains a unique identifier assigned by the proxy service.

The message context is defined by an XML schema. You must use XQuery expressions to manipulate the context variables in the message flow that defines a proxy service.

The predefined context variables provided by Oracle Service Bus can be grouped into the following types:

- Message-related variables

- Inbound and outbound variables

- Operation variable

- Fault variable

For information about the predefined context variables, see Section 39.2, "Predefined Context Variables."

The `$body contains` message payload variable. When a message is dispatched from Oracle Service Bus you can decide the variables, whose you want to include in the outgoing message. That determination is dependent upon whether the target endpoint is expecting a SOAP or a non-SOAP message:

- For a binary, any text or XML message content inside the Body element in `$body` is sent.

- For MFL messages, the Body element in `$body` contains the XML equivalent of the MFL document.

- For text messages, the Body element in `$body` contains the text. For text attachments, the body element in `$attachments` contains the text. If the contents are XML instead of simple text, the XML is sent as a text message.

- For XML messages, the Body element in `$body` contains the XML. For XML attachments, the body element in `$attachments` contains the XML.

- SOAP messages are constructed by wrapping the contents of the header and body variables inside a `<soap:Envelope>` element. (The SOAP 1.1 namespace is used for SOAP 1.1 services, while the SOAP 1.2 namespace is used for SOAP 1.2 services.) If the body variable contains a piece of reference XML, it is sent.That is the referenced content is not substituted in the message.

For non-SOAP services, if the Body element of `$body` contains a binary-content element, then the referenced content stored internally is sent 'as is', regardless of the target service type.

For more information, see Chapter 39, "Message Context."

The types for the message context variables are defined by the message context schema (`MessageContext.xsd`). When working with the message context variables in the Oracle XQuery Mapper, you need to reference `MessageContext.xsd`, which is available in a JAR file, *OSB_ORACLE_HOME*/lib/sb-schemas.jar, and the transport-specific schemas, which are available at

OSB_ORACLE_HOME/lib/transports/

To learn about the message context schema and the transport specific schemas, see Section 39.10, "Message Context Schema."

37.10.2 Guidelines for Viewing and Altering Message Context

Consider the following guidelines when you want to inspect or alter the message context:

- In an XQuery expression, the root element in a variable is not present in the path in a reference to an element in that variable. For example, the following XQuery expression obtains the Content-Description of the first attachment in a message:

  ```
  $attachments/ctx:attachment[1]/ctx:content-Description
  ```

  To obtain the second attachment

  ```
  $attachments/ctx:attachment[2]/ctx:body/*
  ```

- A context variable can be empty or it can contain a single XML element or a string value. However, an XQuery expression often returns a sequence. When you use an XQuery expression to assign a value to a variable, only the first element in the sequence returned by the expression is stored as the variable value. For example, to assign the value of a WS-Addressing Message ID from a SOAP header (assuming there is one in the header) to a variable named idvar, the assign action specification is:

  ```
  assign data($header/wsa:messageID to variable idvar
  ```

  > **Note:** In this case, if two WS-Addressing MessageID headers exist, the idvar variable will be assigned the value of the first one.

- The variables $header, $body, and $attachments are never empty. However, $header can contain an empty SOAP Header element, $body can contain an empty SOAP Body element, and $attachments can contain an empty attachment element.

- In cases in which you use a transformation resource (XSLT or XQuery), the transformation resource is defined to transform the document in the SOAP body of a message. To make this transformation case easy and efficient, the input parameter to the transformation can be an XQuery expression. For example, you can use the following XQuery expression to feed the business document in the Body element of a message ($body) as input to a transformation:

  ```
  $body/* [1]
  ```

  The result of the transformation can be put back in $body with a replace action. That is replace the content of $body, which is the content of the Body element. For more information, see Chapter 11, "XQuery Transformations" and Chapter 15, "XSL Transformations."

- In addition to inserting or replacing a single element, you can also insert or replace a selected sequence of elements using an insert or replace action. You can configure an XQuery expression to return a sequence of elements. For example, you can use insert and replace actions to copy a set of transport headers from $inbound to $outbound. For information on adding an action, see Section 21.1, "Adding and Editing Actions in Message Flows." For an example, see Section 37.10.3, "Copying JMS Properties From Inbound to Outbound."

37.10.3 Copying JMS Properties From Inbound to Outbound

It is assumed that the interfaces of the proxy services and of the invoked business service may be different. Therefore, Oracle Service Bus does not propagate any

information (such as the transport headers and JMS properties) from the inbound variable to the outbound variable.

The transport headers for the proxy service's request and response messages are in `$inbound` and the transport headers for the invoked business service's request and response are in `$outbound`.

For example, the following XQuery expression can be used in a case where the user-defined JMS properties for a one-way message (an invocation with no response) need to be copied from inbound message to outbound message:

Use the transport headers action to set

```
$inbound/ctx:transport/ctx:request/tp:headers/tp:user-header
```

as the first child of:

```
./ctx:transport/ctx:request/tp:headers
```

in the outbound variable.

To learn how to configure the transport header action, see Section 21.7, "Adding Transport Header Actions."

37.11 Working with Variable Structures

This section includes the following topics:

- Section 37.11.2, "Using Variable Structures"

- Section 37.11.2, "Using Variable Structures"

- Section 37.11.3, "Creating Variable Structure Mappings"

37.11.1 Using the Inline XQuery Expression Editor

Oracle Service Bus lets you import XQueries that have been created with an external tool such as the Oracle XQuery Mapper. You can use these XQueries anywhere in the proxy service message flow by binding the XQuery resource input to an Inline XQuery, and binding the XQuery resource output to an action that uses the result as the input; for example, the assign, replace, or insert actions.

However, you can enter the XQuery inline as part of the action definition instead of entering the XQuery as a resource. You can also use Inline XQueries for the condition in an **If...Then...** action.

Use the Inline XQuery Expression Editor to enter simple XQueries that consist of the following:

- Fragments of XML with embedded XQueries.

- Simple variable paths along the child axis.

> **Note:** For more complex XQueries, it is recommended that you use the XQuery Mapper, especially if you are not familiar with XQuery.

Inline XQueries can be used effectively to:

- Create variable structures by using the Inline XQuery Expression Editor. See Section 37.11.2, "Using Variable Structures."

- Extract or access a business document or RPC parameter from the SOAP envelope elements in `$header` or `$body`.

- Extract or access an attachment document in `$attachments`.

- Set up the parameters of a service callout action by extracting it from the SOAP envelope.

- Insert the result parameter of a service callout action into the SOAP envelope.

- Extract a sequence from the SOAP envelope to drive a `for loop`.

- Update an item in the sequence in a `for loop` with an Update action.

> **Note:** You can also use the Inline XQuery Expression Editor to create variable structures. For more information, see Section 37.11.2, "Using Variable Structures."

37.11.1.1 Inline XQueries

Oracle Service Bus lets you import XQueries that have been created with an external tool such as the Oracle XQuery Mapper. You can use these XQueries anywhere in the proxy service message flow by binding the XQuery resource input to an inline XQuery, and binding the XQuery resource output to an action that uses the result as the action input; for example, the assign, replace, or insert actions. However, you can enter the XQuery inline as part of the action definition instead of entering the XQuery as a resource. You can also use inline XQueries for the condition in an **If...Then...** action.

The inline XQuery and XPath editors allow you to declare a variable's structure by mapping it to a type or element and then creating path expressions with a drag and drop action from the graphical representation of the structure. You can also enter the path expressions manually.

You can use this feature directly for all user-defined variables, as well as `$inbound`, `$outbound`, and `$fault`. However, you cannot use it directly to access XML attachments in `$attachments`, headers in `$header`, or documents and RPC parameters in `$body`, with one exception— you can use it directly to access documents and parameters in `$body` for request messages received by a WSDL proxy service.

To learn more about creating variable structures, see Section 37.11.3, "Creating Variable Structure Mappings."

To learn more about XQuery engine support and the relationship with Oracle functions and operators, see Chapter 43, "XQuery Implementation."

37.11.1.2 Uses of the Inline XQuery Expression Editor

You typically use the Inline XQuery Expression Editor to enter simple XQueries that consist of the following:

- Fragments of XML with embedded XQueries.

- Simple variable paths along the child axis.

> **Note:** For more complex XQueries, we recommend that you use the Oracle XQuery Mapper, an editor with drag-and-drop functionality. See "XQuery Mapper" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Examples of good uses of inline XQueries are:

- Extract or access a business document or RPC parameter from the SOAP envelope elements in `$header` or `$body`.

- Extract or access an attachment document in `$attachments`.

- Set up the parameters of a service callout by extracting it from the SOAP envelope.

- Fold the result parameter of a service callout into the SOAP envelope.

- Extract a sequence from the SOAP envelope to drive a `for loop`.

- Update an item in the sequence in a `for loop` with an Update action.

You can also use the Inline XQuery Expression Editor to create variable structures. For more information, see Section 37.11.2, "Using Variable Structures."

37.11.1.2.1 Best Practices for Type-Dependent Expressions To help ensure expected results when using type-dependent expressions, manually cast values to the desired types. For example, the following statement casts `counter` as an integer for the XQuery compiler, which ensures a single return value:

```
<Body><result>{$foo/item[xs:int($counter)]}</result></Body>
```

37.11.2 Using Variable Structures

You can use the Inline XQuery Expression Editor to create variable structures, with which you define the structure of a given variable for design purposes. For example, it is easier to browse the XPath variable in the Administration Console rather than viewing the XML schema of the XPath variable.

> **Note:** It is not necessary to create variable structures for your runtime to work. Variable structures define the structure of the variable or the variable path but do not create the variable. Variables are created at runtime as the target of the assign action in the stage.

In a typical programming language, the scope of variables is static. Their names and types are explicitly declared. The variable can be accessed anywhere within the static scope.

In Oracle Service Bus, there are some predefined variables, but you can also dynamically create variables and assign value to them using the assign action or using the loop variable in the for-loop. When a value is assigned to a variable, the variable can be accessed anywhere in the proxy service message flow. The variable type is not declared but the type is essentially the underlying type of the value it contains at any point in time.

> **Note:** The scope of the for-loop variable is limited and cannot be accessed outside the stage.

When you use the Inline XQuery Expression Editor, the XQuery has zero or more inputs and one output. Because you can display the structure of the inputs and the structure of the output visually in the Expression Editor itself, you do not need to open the XML schema or WSDL resources to see their structure when you create the Inline XQuery. The graphical structure display also enables you to drag and drop simple variable paths along the child axis without predicates, into the composed XQuery.

Each variable structure mapping entry has a label and maps a variable or variable path to one or more structures. The scope of these mappings is the stage or route node. Because variables are not statically typed, a variable can have different structures at different points (or at the same point) in the stage or route node. Therefore, you can map a variable or a variable path to multiple structures, each with a different label. To view the structure, select the corresponding label with a list.

> **Note:** You can also create variable structure mappings in the Inline XPath Expression Editor. However, although the variable or a variable path is mapped to a structure, the XPaths generated when you select from the structure are XPaths relative to the variable. An example of a relative XPath is `./ctx:attachment/ctx:body`.

37.11.3 Creating Variable Structure Mappings

The following sections describe how to create several types of variable structure mappings:

- Section 37.11.3.1, "Sample WSDL"

- Section 37.11.3.2, "Creating the Resources You Need for the Examples"

- Section 37.11.3.3, "Example 1: Selecting a Predefined Variable Structure"

- Section 37.11.3.4, "Example 2: Creating a Variable Structure That Maps a Variable to a Type"

- Section 37.11.3.5, "Example 3: Creating a Variable Structure that Maps a Variable to an Element"

- Section 37.11.3.6, "Example 4: Creating a Variable Structure That Maps a Variable to a Child Element"

- Section 37.11.3.7, "Example 5: Creating a Variable Structure that Maps a Variable to a Business Service"

- Section 37.11.3.8, "Example 6: Creating a Variable Structure That Maps a Child Element to Another Child Element"

37.11.3.1 Sample WSDL

This sample WSDL is used in most of the examples in this section. You need to save this WSDL as a resource in your configuration. For more information, see Section 37.11.3.2, "Creating the Resources You Need for the Examples."

Example 37–17   Sample WSDL

```
<definitions
    name="samplewsdl"
    targetNamespace="http://example.org"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:s0="http://www.oracle.com"
    xmlns:s1="http://example.org"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<types>
  <xs:schema
    attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    targetNamespace="http://www.oracle.com"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
        <xs:element name="PO" type="s0:POType"/>
        <xs:complexType name="POType">
          <xs:all>
            <xs:element name="id" type="xs:string"/>
            <xs:element name="name" type="xs:string"/>
          </xs:all>
        </xs:complexType>
        <xs:element name="Invoice" type="s0:InvoiceType"/>
        <xs:complexType name="InvoiceType">
          <xs:all>
            <xs:element name="id" type="xs:string"/>
            <xs:element name="name" type="xs:string"/>
          </xs:all>
        </xs:complexType>
</xs:schema>
</types>
<message name="POTypeMsg">
    <part name="PO" type="s0:POType"/>
</message>
<message name="InvoiceTypeMsg">
    <part name="InvReturn" type="s0:InvoiceType"/>
</message>

<portType name="POPortType">
    <operation name="GetInvoiceType">
      <input message="s1:POTypeMsg"/>
      <output message="s1:InvoiceTypeMsg"/>
    </operation>
</portType>
<binding name="POBinding" type="s1:POPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetInvoiceType">
      <soap:operation soapAction="http://example.com/GetInvoiceType"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </operation>
</binding>
</definitions>
```

37.11.3.2 Creating the Resources You Need for the Examples

To make use of the examples that follow, you save the sample WSDL as a resource in your configuration and create the sample business service and proxy service that use the sample WSDL.

The instructions that follow tell how to accomplish the tasks in the Oracle Service Bus Administration Console:

- Section 37.11.3.2.1, "Save the WSDL as a Resource"

- Section 37.11.3.2.2, "Create a Proxy Service That Uses the Sample WSDL"

- Section 37.11.3.2.3, "Build a Message Flow for the Sample Proxy Service"

- Section 37.11.3.2.4, "Create a Business Service That Uses the Sample WSDL"

37.11.3.2.1 Save the WSDL as a Resource Perform the following steps:

1. In the left navigation pane in the Oracle Service Bus Administration Console, under **Change Center**, click **Create** to create a new session for making changes to the current configuration.

2. In the left navigation pane, click **Project Explorer**.

3. In the **Project View** page, click the project to which you want to add the WSDL.

4. In the **Project View** page, in the **Create Resource** field, select **WSDL** under **Interface**.

5. In the **Create a New WSDL Resource** page in the **Resource Name** field, enter `SampleWSDL`. This is a required field.

6. In the **WSDL** field, copy and paste the text from the sample WSDL into this field.

   This is a required field.

7. Click **Save**. The new WSDL `SampleWSDL` is included in the list of resources and saved in the current session. You must now create a proxy service that uses this WSDL, see Section 37.11.3.2.2, "Create a Proxy Service That Uses the Sample WSDL."

37.11.3.2.2 Create a Proxy Service That Uses the Sample WSDL Perform the following steps:

1. In the left navigation pane, click **Project Explorer**.

2. In the **Project View** page, select the project to which you want to add the proxy service.

3. In the **Project View** page, in the **Create Resource** field, select **Proxy Service** under **Service**.

4. In the **Edit a Proxy Service - General Configuration** page, in the **Service Name** field, enter `ProxywithSampleWSDL`. This is a required field.

5. In the **Service Type** field, which defines the types and packaging of the messages exchanged by the service:

   a. Select **WSDL Web Service** from under **Create a New Service**.

   b. Click **Browse**. The **WSDL Browser** is displayed.

   c. Select `SampleWSDL`, then select `POBinding` in the **Select WSDL Definitions** pane.

   d. Click **Submit**.

6. Keep the default values for all other fields on the **General Configuration** page, then click **Next**.

7. Keep the default values for all fields on the **Transport Configuration** pages, then click **Next**.

8. In the **Operation Selection Configuration** page, make sure **SOAP Body Type** is selected in the **Selection Algorithm** field, then click **Next**.

9. Review the configuration data that you have entered for this proxy service, then click **Save**. The new proxy service `ProxywithSampleWSDL` is included in the list of resources and saved in the current session.To build message flow for this proxy service, see Section 37.11.3.2.3, "Build a Message Flow for the Sample Proxy Service."

37.11.3.2.3 Build a Message Flow for the Sample Proxy Service Perform the following steps:

1. In the **Project View** page, in the **Actions** column, click the **Edit Message Flow** icon for the `ProxywithSampleWSDL` proxy service.

2. In the **Edit Message Flow** page, click the `ProxywithSampleWSDL` icon, then click **Add Pipeline Pair**. PipelinePairNode1 is displayed, which includes request and response pipelines.

3. Click the **Request Pipeline** icon, then click **Add Stage**. The Stage **Stage1** is displayed.

4. Click **Save**. The basic message flow is created for the `ProxywithSampleWSDL` proxy service.

37.11.3.2.4 Create a Business Service That Uses the Sample WSDL Perform the following steps:

1. In the left navigation pane, click **Project Explorer**. The Project View page is displayed.

2. Select the project to which you want to add the business service.

3. From the **Project View** page, in the **Create Resource** field, select **Business Service** from under **Service**. The **Edit a Business Service - General Configuration** page is displayed.

4. In the **Service Name** field, enter `BusinesswithSampleWSDL`. This is a required field.

5. In the **Service Type** field, which defines the types and packaging of the messages exchanged by the service, do the following:

   a. Select **WSDL Web Service** from under **Create a New Service**.

   b. Click **Browse**. The **WSDL Browser** is displayed.

   c. Select `SampleWSDL`, then select **POBinding** in the Select **WSDL Definitions** pane.

   d. Click **Submit**.

6. Keep the default values for all other fields on the **General Configuration** page, then click **Next**.

7. Enter an endpoint URI in the **Endpoint URI** field on the **Transport Configuration** page.

8. Click **Add**, and then click **Next**.

9. Use the default values for all fields on the **SOAP Binding Configuration** page.

10. Click **Next**.

11. Review the configuration data that you have entered for this business service, and then click **Save**. The new business service `BusinesswithSampleWSDL` is included in the list of resources and is saved in the current session.

12. From the left navigation pane, click **Activate** under **Change Center**. The session ends and the configuration is deployed to runtime. You are now ready to use the examples—continue in Section 37.11.3.3, "Example 1: Selecting a Predefined Variable Structure."

37.11.3.3 Example 1: Selecting a Predefined Variable Structure

In this example, you select a predefined variable structure using the proxy service `ProxyWithSampleWSDL`, which has a service type **WSDL Web Service** that uses the binding **POBinding** from `SampleWSDL`.

The proxy service message flow needs to know the structure of the message in order to manipulate it. To achieve this, Oracle Service Bus automatically provides a predefined structure that maps the `body` variable to the SOAP body structure as defined by the WSDL of the proxy service for all the messages in the interface. This predefined structure mapping is labeled `body`.

> **Note:** This predefined structure is also supported for messaging services with a typed interface.

To select a predefined variable structure:

In the Variable Structures panel on the XQuery Expression Editor page, select **body** from the list of built-in structures.

The variable structure `body` is displayed in Figure 37–4.

Figure 37–4   Variable Structures—body
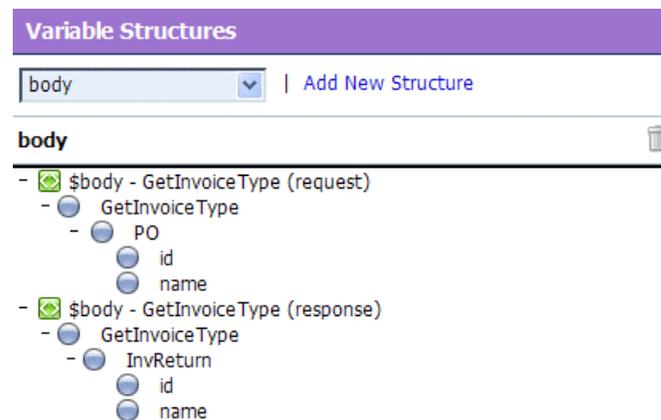
37.11.3.4 Example 2: Creating a Variable Structure That Maps a Variable to a Type

Suppose the proxy service `ProxyWithSampleWSDL` invokes a service callout to the business service `BusinessWithSampleWSDL`, which also has a service type WSDL Web Service that uses the binding **POBinding** from `SampleWSDL`. The operation `GetInvoiceType` is invoked.

In this example, the message flow needs to know the structure of the response parameter in order to manipulate it. To achieve this, you can create a new variable structure that maps the response parameter variable to the type `InvoiceType`.

To map a variable to a type:

1. In the Variable Structures panel, click **Add New Structure**. Additional fields are displayed in Figure 37–5.

Figure 37–5   Variable Structures—Add a New Structure

2. Select the **XML Type**.

3. In the **Structure Label** field, enter `InvoiceType` as the display name for the variable structure you want to create. This display name enables you to give a meaningful name to the structure so you can recognize it at design time but it has no impact at runtime.

4. In the **Structure Path** field, enter `$InvoiceType` as the path of the variable at runtime.

5. To select the type **InvoiceType**, do the following:

   a. Under the **Type** field, select the appropriate option, then select **WSDL Type** from the list.

   b. Click **Browse**. The **WSDL Browser** is displayed.

   c. In the **WSDL Browser**, select **SampleWSDL**, then select **InvoiceType** under **Types** in the **Select WSDL Definitions** pane.

   d. Click **Submit**. **InvoiceType** is displayed under your selection **WSDL Type**.

6. Click **Add**. The new variable structure **InvoiceType** is included under **XML Type** in the list of variable structures.

   The variable structure **InvoiceType** is displayed in Figure 37–6.

Figure 37–6   Variable Structures—InvoiceType

37.11.3.5 Example 3: Creating a Variable Structure that Maps a Variable to an Element

Suppose a temporary variable has the element **Invoice** described in the `SampleWSDL` WSDL. In this example, the `ProxyWithSampleWSDL` message flow needs to access this variable. To achieve this, you can create a new variable structure that maps the variable to the element **Invoice**.

To map a variable to an element:

1. In the Variable Structures panel, click **Add New Structure**.

2. Make sure you select the **XML Type**.

3. In the **Structure Label** field, enter `Invoice` as the meaningful display name for the variable structure you want to create.

4. In the **Structure Path** field, enter `$Invoice` as the path of the variable structure at runtime.

5. To select the element **Invoice**, do the following:

    a. For the **Type** field, make sure you select the appropriate option. Then select **WSDL Element**.

    b. Click **Browse**.

    c. In the **WSDL Browser**, select `SampleWSDL`, then select **Invoice** under **Elements** in the **Select WSDL Definitions** pane.

    d. Click **Submit**. **Invoice** is displayed under your selection **WSDL Element**.

6. Click **Add**. The new variable structure **Invoice** is included under **XML Type** in the list of variable structures.

    The variable structure **Invoice** is displayed in Figure 37–7.

Figure 37–7   Variable Structures—Invoice

37.11.3.6 Example 4: Creating a Variable Structure That Maps a Variable to a Child Element

The `ProxyWithSampleWSDL` proxy service routes to the document style `Any SOAP` business service that returns the Purchase Order in the `SOAP` body. In this example, the `ProxyWithSampleWSDL` proxy service message flow must then manipulate the response. To achieve this, you can create a new structure that maps the **body** variable to the **PO** element, and specify the **PO** element as a child element of the variable. You need to specify it as a child element because the **body** variable contains the SOAP Body element and the **PO** element is a child of the Body element.

To map a variable to a child element:

1. In the Variable Structures panel, click **Add New Structure**.

2. Make sure you select the **XML Type**.

3. In the **Structure Label** field, enter `body to PO` as the meaningful display name for the variable structure you want to create.

4. In the **Structure Path** field, enter `$body` as the path of the variable structure at runtime.

5. To select the **PO** element:

   a. Under the **Type** field, make sure you select the appropriate option, and then select **WSDL Element**.

   b. Click **Browse**.

   c. In the **WSDL Browser**, select `SampleWSDL`, then select **PO** under **Elements** in the **Select WSDL Definitions** pane.

   d. Click **Submit**.

6. Select the **Set as child** check box to set the **PO** element as a child of the **body to PO** variable structure.

7. Click **Add**. The new variable structure **body to PO** is included under **XML Type** in the list of variable structures.

   The variable structure **body to PO** is displayed in Figure 37–8.

Figure 37–8   Variable Structures—body to PO

37.11.3.7 Example 5: Creating a Variable Structure that Maps a Variable to a Business Service

The `ProxyWithSampleWSDL` proxy service routes the message to the `BusinessWithSampleWSDL` business service, which also has a service type WSDL Web Service that uses the binding **POBinding** from `SampleWSDL`. In this example, the message flow must then manipulate the response. To achieve this, you can define a

new structure that maps the **body** variable to the `BusinessWithSampleWSDL` business service. This results in a map of the **body** variable to the SOAP body for all the messages in the WSDL interface of the service.

> **Note:** This mapping is also supported for messaging services with a typed interface.

To map a variable to a business service

1. In the Variable Structures panel, click **Add New Structure**.

2. Select **Service Interface**.

3. In the **Structure Label** field, enter `BusinessService` as the meaningful display name for the variable structure.

4. In the **Structure Path** field, **$body** is already set as the default. This is the path of the variable structure at runtime.

5. To select the business service, do the following:

   a. Under the **Service** field, click **Browse**. The **Service Browser** is displayed.

   b. In the **Service Browser**, select the `BusinessWithSampleWSDL` business service, then click **Submit**. The business service is displayed under the **Service** field.

   c. In the **Operation** field, select **All**.

6. Click **Add**. The new variable structure `BusinessService` is included under **Service Interface** in the list of variable structures.

   The variable structure `BusinessService` is displayed in Figure 37–9.

Figure 37–9   Variable Structures—Business Service
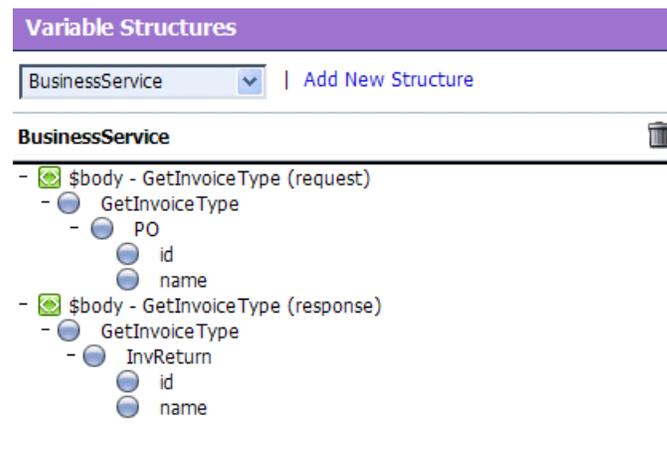
37.11.3.8 Example 6: Creating a Variable Structure That Maps a Child Element to Another Child Element

Modify the `SampleWSDL` so that the `ProxyWithSampleWSDL` proxy service receives a single attachment. The attachment is a Purchase Order. In this example, the proxy service message flow must then manipulate the Purchase Order. To achieve this, you can define a new structure that maps the **body** element in **$attachments** to the **PO**

element, which is specified as a child element. The **body** element is specified as a variable path of the form:

```
$attachments/ctx:attachment/ctx:body
```

You can select and copy the **body** element from the predefined **attachments** structure, paste this element as the variable path to be mapped in the new mapping definition.

To map a child element to another child element:

1. In the Variable Structures panel, select **attachments** from the list of built-in structures.

   The variable structure **attachments** is displayed in Figure 37–10.

Figure 37–10   Variable Structures—attachments
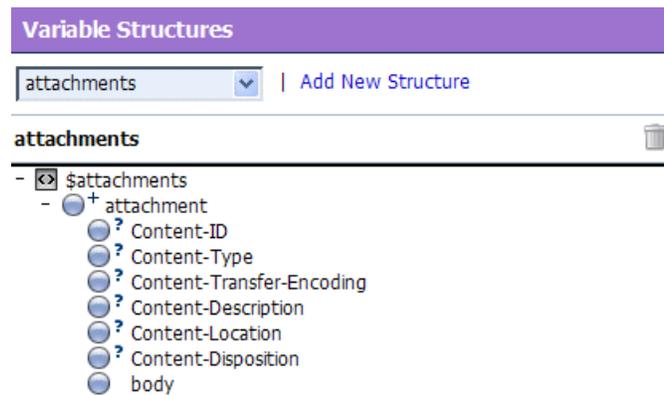
2. Select the `body` child element in the attachments structure. The variable path of the body element is displayed in the Property Inspector on the right side of the page:

   ```
   $attachments/ctx:attachment/ctx:body
   ```

3. Copy the variable path of the `body` element.

4. In the Variable Structures panel, click **Add New Structure**.

5. Select the **XML Type**.

6. In the **Structure Label** field, enter `PO` attachment as the meaningful display name for this variable structure.

7. In the **Structure Path** field, paste the variable path of the body element:

   ```
   $attachments/ctx:attachment/ctx:body
   ```

   This is the path of the variable structure at runtime.

8. To select the `PO` element:

   a. Under the **Type** field, make sure the appropriate option is selected, then select **WSDL Element**.

   b. Click **Browse**.

   c. In the **WSDL Browser**, select `SampleWSDL`, then select **PO** under **Elements** in the **Select WSDL Definitions** pane.

   d. Click **Submit**.

9. Select the **Set as child** check box to set the PO element as a child of the **body** element.

10. Click **Add**. The new variable structure **PO attachment** is included under XML Type in the list of variable structures.

11. If there are multiple attachments, add an index to the reference when you use fields from this structured variable in your XQueries. For example, if you drag the PO field to the XQuery field, but as PO will be the second attachment, change the inserted value from

```
$attachments/ctx:attachment/ctx:body/oracle:PO/oracle:id
```

to

```
$attachments/ctx:attachment[2]/ctx:body/oracle:PO/oracle:id
```

37.12 Quality of Service

The following sections discuss quality of service features in Oracle Service Bus messaging:

- Section 37.12.1, "Delivery Guarantees"
- Section 37.12.2, "Outbound Message Retries"

37.12.1 Delivery Guarantees

Oracle Service Bus supports reliable messaging. When messages are routed to another service from a route node, the default quality of service (QoS) is *exactly-once* if the proxy service is configured to be transactional; otherwise *best-effort* QoS is supported.

Quality of service is set in the `qualityOfService` element in the `$outbound` context variable.

The following delivery guarantee types are provided in Oracle Service Bus, shown in Table 37–10.

Table 37–10   Delivery Guarantee Types

Delivery Reliability	Description
Exactly-once	*Exactly-once* reliability means that messages are delivered from inbound to outbound exactly-once, assuming a terminating error does not occur before the outbound message send is initiated. *Exactly-once* means reliability is optimized.
	Exactly-once delivery reliability is a hint, not a directive. When *exactly-once* is specified, *exactly-once* reliability is provided if possible. If *exactly-once* is not possible, then *at-least-once* delivery semantics are attempted; if that is not possible, *best-effort* delivery is performed.
	Proxy services configured to be transactional have *exactly-once* quality of service.
	The default value of the `qualityOfService` element is also `exactly-once` for a route node action for the following inbound transports:
	■ email
	■ FTP
	■ SFTP
	■ File
	■ JMS (transactional)
	■ Tuxedo (transactional)
	■ MQ (with Backout Threshold set to zero)
	■ WS
	Note: Do not retry the outbound transport when the `QoS` is exactly-once
At-least-once	*At-least-once* semantics means the message is delivered to the outbound from the inbound at least once, assuming a terminating error does not occur before the outbound message send is initiated. Delivery is considered satisfied even if the target service responds with a transport-level error. However it is not satisfied in the case of a time-out, a failure to connect, or a broken communication link. If failover URLs are specified, *at-least-once* semantics is provided with respect to at least one of the URLs.
	At-least-once delivery semantics is attempted if *exactly-once* is not possible but the `qualityOfService` element is `exactly-once`.

Table 37–10   (Cont.) Delivery Guarantee Types

Delivery Reliability	Description
Best-effort	*Best-effort* means that there is no reliable messaging and there is no elimination of duplicate messages—however, performance is optimized. It is performed if the `qualityOfService` element is `best-effort`. *Best-effort* delivery is also performed if *exactly-once* and *at-least-once* delivery semantics are not possible but the `qualityOfService` element is `exactly-once`.
	The default value of the `qualityOfService` element for a route node is `best-effort` for the following inbound transports:
	■ HTTP
	■ JMS (non-transactional)
	■ Tuxedo (non-transactional)
	■ MQ (with Backout Threshold set to greater than zero)
	The default value of the `qualityOfService` element is always `best-effort` for the following:
	■ Service callout action – always `best-effort`, but can be changed if required.
	■ Publish action – defaults to `best-effort`, modifiable
	For more information on QoS behavior in publish actions, see Section 37.4.1, "Transformations and Publish Actions."
	Note: When the value of the `qualityOfService` element is `best-effort` for a publish action, all errors are ignored. However, when the value of the `qualityOfService` element is `best-effort` for a route node action or a Service Callout action, any error will raise an exception.

For more detailed information about quality of service for other transports, see "Transports" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

37.12.1.1 Overriding the Default Element Attribute

To override the default *exactly-once* quality of service attribute, you must set the `qualityOfService` in the outbound message context variable (`$outbound`). For more information, see Section 39.10, "Message Context Schema."

You can override the default `qualityOfService` element attribute for the following message flow actions:

- Publish
- Dynamic Publish
- Publish Table
- Service Callout
- Routing
- Dynamic Routing
- Routing Table

To override the `qualityOfService` element attribute, add a Routing Options action to any of the above actions, select the QoS option, and choose the override value.

For more information on message context variables, see Section 39.10, "Message Context Schema."

37.12.1.2 Delivery Guarantee Rules

The delivery guarantee supported when a proxy service publishes a message or routes a request to a business service depends on the following conditions:

- The value of the `qualityOfService` element.

- The inbound transport (and connection factory, if applicable).

- The outbound transport (and connection factory, if applicable).

However, if the inbound proxy service is a Local Transport and is invoked by another proxy service, the inbound transport of the invoking proxy service is responsible for the delivery guarantee. That is because a proxy service that invokes another proxy service is optimized into a direct invocation if the transport of the invoked proxy service is a Local Transport. For more information on transport protocols, see Chapter 20, "Proxy Services: Creating and Managing" and Chapter 19, "Business Services: Creating and Managing."

Note: No delivery guarantee is provided for responses from a proxy service.

The following rules govern delivery guarantees, shown in Table 37–11.

Table 37–11   Delivery Guarantee Rules

Delivery Guarantee Provided	Rule
Exactly-once	The proxy service inbound transport is transactional and the value of the `qualityOfService` element is `exactly-once` to an outbound transport.
At-least-once	The proxy service inbound transport is file, FTP, or email and the value of the `qualityOfService` element is `exactly-once`.
At-least-once	The proxy service inbound transport is transactional and the value of the `qualityOfService` element, where applicable, is `exactly-once` to an outbound transport that is not transactional.
No delivery guarantee	All other cases, including all response processing cases.

Note: To support *at-least-once* and *exactly-once* delivery guarantees with JMS, you must exploit JMS transactions and configure a retry count and retry interval on the JMS queue to ensure that the message is redelivered in the event of a server crash or a failure that is not handled in an error handler with a *Reply* or *Resume* action. File, FTP, and email transports also internally use a JMS/XA queue. The default retry count for a proxy service with a JMS/XA transport is 1. For a list of the default JMS queues created by Oracle Service Bus, see the *Oracle Fusion Middleware Deployment Guide for Oracle Service Bus*.

The following are additional delivery guarantee rules:

- If the transport of the inbound proxy service propagates or starts a transaction, the request processing is performed in a transaction.

  - When the `qualityOfService` element is set to `exactly-once`, any route node actions executed in the request flow to a transactional destination are

performed in the same transaction. Publish and Service Callout actions in a transaction context are *best-effort* by default and therefore execute outside of the transaction context. Setting those actions to *exactly-once* causes them to execute in the transaction context.

- When the `qualityOfService` element is set to `best-effort` for any action in a route node, service callout or publish actions are executed outside of the request flow transaction. Specifically, for JMS, Tuxedo, Transactional Tuxedo, or EJB transport, the request flow transaction is suspended and the Transactional Tuxedo work is done without a transaction or in a separate transaction that is immediately committed.

- If an error occurs during request processing, but is caught by a user error handler that manages the error (by using the resume or reply action), the message is considered successfully processed and the transaction commits. A transaction is aborted if the system error handler receives the error—that is, if the error is not handled before reaching the system level. The transaction is also aborted if a server failure occurs during request pipeline processing.

■ If a response is received by a proxy service that uses a JMS/XA transport to business service (and the proxy inbound is not Transactional Tuxedo), the response processing is performed in a single transaction.

- When the `qualityOfService` element is set to `exactly-once`, all route, service callout, and publish actions are performed in the same transaction.

- When the `qualityOfService` element is set to `best-effort`, all publish actions and service callout actions are executed outside of the response flow transaction. Specifically, for JMS, EJB, or transactional Tuxedo types of transports, the response flow transaction is suspended and the service is invoked without a transaction or in a separate transaction that is immediately committed.

- Proxy service responses executed in the response flow to a JMS/XA destination are always performed in the same transaction, regardless of the `qualityOfService` element setting.

■ If the proxy service inbound transport is transactional Tuxedo, or if you set the "Same Transaction for Response" option on a proxy service, both the request processing and response processing are done in this transaction.

Note: You will encounter a runtime error when the inbound transport is transactional Tuxedo and the outbound is an asynchronous transport, for example, JMS/XA.

37.12.1.3 Threading Model

The Oracle Service Bus threading model works as follows:

■ The request and response flows in a proxy service execute in different threads.

■ Service callouts are always blocking. An HTTP route or publish action is non-blocking (for request/response or one-way invocation), if the value of the `qualityOfService` element is `best-effort`.

■ JMS route actions or publish actions are always non-blocking, but the response is lost if the server restarts after the request is sent because Oracle Service Bus has no persistent message processing state.

> **Note:** In a request or response flow publish action, responses are always discarded because publish actions are inherently a one-way message send.

37.12.1.4 Splitting Proxy Services

You may want to split a proxy service in the following situations:

- When HTTP is the inbound and outbound transport for a proxy service, you may want to incorporate enhanced reliability into the middle of the message flow. To enable enhanced reliability in this way, split the proxy service into a front-end HTTP proxy service and a back-end JMS (one-way or request/response) proxy service with an HTTP outbound transport. In the event of a failure, the first proxy service must quickly place the message in the queue for the second proxy service, in order to avoid loss of messages.

- To disable the direct invocation optimization for a non-JMS transport when a proxy service, say `loanGateway1` invokes another proxy service, say `loanGateway2`. Route to the proxy service `loanGateway2` from the proxy service `loanGateway1` where the proxy service `loanGateway2` uses JMS transport.

- To have an HTTP proxy service publish to a JMS queue but have the publish action rollback if there is a exception later on in the request processing, split the proxy service into a front-end HTTP proxy service and a back-end JMS proxy service. The publish action specifies a `qualityOfService` element of `exactly-once` and uses an XA connection factory.

37.12.2 Outbound Message Retries

In addition to configuring inbound retries for messages using JMS, you can configure outbound retries and load balancing. Load balancing, failover, and retries work in conjunction to provide performance and high availability. For each message, the list of URLs you provide as failover URLs is automatically ordered based on the load balancing algorithm into a failover sequence. If the retry count is N, the entire sequence is retried N times before stopping. The system waits for the specified retry interval before commencing subsequent loops through the sequence. After completing the retry attempts, if there is still an error, the error handler pipeline for the route node is invoked. For more information on the error handler pipeline, see Section 24.3, "Adding Pipeline Error Handlers.".

> **Note:** For HTTP transports, any HTTP status other than 200 or 202 is considered an error by Oracle Service Bus and must be retried. Because of this algorithm, it is possible that Oracle Service Bus retries errors like authentication failure that may never be rectified for that URL within the time period of interest. On the other hand, if Oracle Service Bus also fails over to a different URL for subsequent attempts to send a given message, the new URL may not give the error.
>
> For `quality of service=exactly-once`, failover or retries will not be executed.

37.13 Using Work Managers with Oracle Service Bus

Oracle WebLogic Server helps you optimize the performance of your applications and Web services environment as well as maintain service-level agreements with a feature

called Work Managers. You create Work Manager resources and configure them by defining work execution rules. Oracle WebLogic Server uses the rules in a Work Manager to help prioritize work and allocate threads in whatever application or component the Work Manager is placed.

For more information about Work Managers, see "Using Work Managers to Optimize Scheduled Work" in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

37.13.1 Configuring Business and Proxy Services to use Work Managers

In Oracle Service Bus, several transports for proxy and business services provide a configuration option called **Dispatch Policy** that lets you associate a Work Manager with a service to prioritize service work. By configuring the Dispatch Policy of an Oracle Service Bus proxy service, the startup and execution of its request pipeline are governed by the rules of the Work Manager. For example, given a proxy service using a Dispatch Policy with a Max Constraint of five, the proxy service will have no more than five request pipeline tasks executing simultaneously.

While the request pipeline is governed by the proxy service's Dispatch Policy, the response pipeline is governed by the business service or Split-Join Dispatch Policy. The RouteTo action specifies a business service or Split-Join to route the message to, and the response is subject to any Dispatch Policy on that business service or Split-Join.

When a RouteTo action specifies a local proxy service, the Dispatch Policy of the original proxy applies to work done in the local proxy's request pipeline. When the local proxy or chain of local proxies reaches a RouteTo action that invokes a business service or Split-Join, the business service, Split-Join, and all the following response pipelines are governed by the Dispatch Policy of that business service or Split-Join.

If an error occurs in the request, the error response is handled in the same thread as the request pipeline.

37.13.2 Quality of Service

The quality of service (QoS) specified in the outbound metadata can also impact the way requests are threaded and thus impact what you see when monitoring Work Managers. The QoS on the RouteTo action defaults to the inbound QoS. For example, some inbound transactional transports set QoS to "exactly-once," such as JMS/XA, SB, Tuxedo, and WS (WS-RM). Other inbound transports, such as HTTP, set QoS to "best-effort" by default. The QoS on the RouteTo action is inherited from the inbound unless overridden by user settings in the RouteTo action.

When "best-effort" is used, the Route node invokes the business service asynchronously. In the case of "best-effort," the work thread returns to the pool and does not wait for the response; therefore, there is not a thread counting against a Work Manager constraint even though there is pending work due back asynchronously. But if "exactly-once" is selected, the request thread sends the request, blocks waiting for the response, and counts against the Work Manager constraints. In the case of "exactly-once," the waiting thread applies to the Work Manager assigned to the proxy service. Once a positive response arrives, a new thread processes the response pipeline using the Dispatch Policy assigned to the business service or Split-Join.

While using "exactly-once" is more expensive from performance, memory, and threading standpoints, "exactly-once" is necessary to maintain integrity on transactional inbound and outbound resources.

For more information on QoS, see Section 37.12, "Quality of Service."

37.13.3 Best Practices

When applying Work Managers to Oracle Service Bus components, follow these recommendations for best results. Ignoring the first two items in the list below could result in a deadlock at runtime.

- If you apply a Work Manager to a proxy service that uses a synchronous transport, apply a different Work Manager to the business service that is invoked by that proxy service's message flow.

- In the above case, the Work Managers for the proxy service and the business service must have different constraints, even if the values should be the same.

- Apply a new Work Manager to Service Bus components instead of using the default Work Manager. You need to create the Work Manager using the WebLogic Server Administration Console before you can apply it to a Service Bus component.

- To avoid associating a Work Manager with multiple Service Bus components, give the Work Manager and Max Thread Constraint the same name as the Service Bus component and add a suffix to identify them, such as "WorkManager" and "MaxThread" respectively.

37.14 Content Types, JMS Type, and Encoding

To support interoperability with heterogeneous endpoints, Oracle Service Bus lets you control the content type, the JMS type, and the encoding used.

Oracle Service Bus does not make assumptions about what the external client or service needs, but uses the information configured for this purpose in the service definition. Oracle Service Bus derives the content type for outbound messages from the service type and interface. Content type is a part of the email and HTTP protocols.

If the service type is:

- XML or SOAP with or without a WSDL, the content type is text/XML.

- Messaging and the interface is MFL or binary, the content type is binary/octet-stream.

- Messaging and the interface is text, the content type is text/plain.

- Messaging and the interface is XML, the content type is text/XML.

- Messaging and the interface is Java, the content type is a Java Object.

Additionally, there is a JMS type, which can be byte or text for non-Java-type messages. You configure the JMS type to use when you define the service in Oracle Service Bus Administration Console or in the Oracle Service Bus plug-ins for Eclipse.

You can override the content type in the outbound context variable (`$outbound`) for proxy services invoking a service, and in the inbound context variable (`$inbound`) for a proxy service response. For more information on `$outbound` and `$inbound` context variables, see Section 39.4, "Inbound and Outbound Variables."

Encoding is also explicitly configured in the service definition for all outbound messages. For more information on service definitions, see Chapter 20, "Proxy Services: Creating and Managing" and Chapter 19, "Business Services: Creating and Managing."

37.15 Throttling Pattern

In Oracle Service Bus, you can restrict the message flow to a business service. This technique of restricting a message flow to a business service is known as throttling. For information, see Chapter 50, "Throttling."

37.16 WS-I Compliance

Oracle Service Bus provides Web Service Interoperability (WS-I) compliance for SOAP 1.1 services in the runtime environment. The WS-I basic profile has the following goals:

- Disambiguate the WSDL and SOAP specifications wherever ambiguity exists.

- Define constraints that can be applied when receiving messages or importing WSDLs so that interoperability is enhanced. When messages are sent, construct the message so that the constraints are satisfied.

The WS-I basic profile is available at the following URL:
http://www.ws-i.org/Profiles/BasicProfile-1.1.html

When you configure a proxy service or business service based on a WSDL, you can use the Oracle Service Bus Administration Console or the Oracle Service Bus plug-ins for Eclipse to specify whether you want Oracle Service Bus to enforce WS-I compliance for the service. For information on how to do this, see Section 20.2.17, "Operation Selection Configuration Page."

When you configure WS-I compliance for a proxy service, checks are performed on inbound request messages received by that proxy service. When you configure WS-I compliance for an invoked service, checks are performed when any proxy receives a response message from that invoked service. Oracle recommends that you create an error handler for these errors, since by default, the proxy service SOAP client receives a system error handler-defined fault. For more information on creating fault handlers, see:

- Section 24, "Proxy Services: Error Handlers."

- "Adding and Configuring Error Handlers in Message Flows" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

For messages sent from a proxy service, whether as outbound request or inbound response, WS-I compliance checks are not explicitly performed. That is because the pipeline designer is responsible for generating most of the message content. However, the parts of the message generated by Oracle Service Bus should satisfy all of the supported WS-I compliance checks. This includes the following content:

- Service invocation request message.

- System-generated error messages returned by a proxy service.

- HTTP status codes generated by a proxy service.

The Enforce WS-I Compliance check box is displayed as shown in Figure 37–11.

Figure 37–11   Enforce WS-I Compliance Check Box

37.16.1 WS-I Compliance Checks

> **Note:**   WS-I compliance checks require that the system knows what
> operation is being invoked on a service. For request messages received
> by a proxy service, that means that the context variable $operation
> should not be null. That depends upon the operation selection
> algorithm being configured properly. For response messages received
> from invoked services, the operation should be specified in the action
> configurations for route, publish, and service callout.

When you configure WS-I compliance checking for a proxy service or a business
service, Oracle Service Bus carries out the following checks, shown in Table 37–12.

Table 37–12   Oracle Service Bus WS-I Compliance Checks

Check	WS-I Basic Profile Details	Oracle Service Bus Description
3.1.1 SOAP Envelope Structure	**R9980** An Envelope must conform to the structure specified in SOAP 1.1, Section 4, "SOAP Envelope" (subject to amendment).	This check applies to request and response messages. If a response message is checked and the message does not possess an outer `Envelope` tag, a `soap:client` error is generated. If the message is an `Envelope` tag but possesses a different namespace, it is handled by the 3.1.2 SOAP Envelope Namespace.
3.1.2 SOAP Envelope Namespace	**R1015** A Receiver must generate an error if they encounter an envelope whose document element is not `soap:Envelope`.	This check applies to request and response messages and is related to the 3.1.1 SOAP Envelope Structure. If a request message has a local name of `Envelope`, but the namespace is not SOAP 1.1, a `soap:VersionMismatch` error is generated.
3.1.3 SOAP Body Namespace Qualification	**R1014** The child elements of the `soap:body` element in an Envelope must be namespace qualified.	This check applies to request and response messages. All request error messages generate a `soap:Client` error.
3.1.4 Disallowed Constructs	**R1008** An Envelope must not contain a Document Type Declaration.	This check applies to request and response messages. All request error messages generate a `soap:Client` error.

Table 37–12 (Cont.) Oracle Service Bus WS-I Compliance Checks

Check	WS-I Basic Profile Details	Oracle Service Bus Description
3.1.5 SOAP Trailers	**R1011** An Envelope must not have any child elements of `soap:Envelope` following the `soap:body` element.	This check applies to request and response messages. All request error messages generate a `soap:Client` error.
3.1.9 SOAP attributes on SOAP 1.1 elements	**R1032** The `soap:Envelope`, `soap:header`, and `soap:body` elements in an Envelope must not have attributes in the namespace `http://schemas.xmlsoap.org/soap/envelope/`	This check applies to request and response messages. Any request error messages generate a `soap:client` error.
3.3.2 SOAP Fault Structure	**R1000** When an Envelope is a fault, the `soap:Fault` element must not have element children other than `faultcode`, `faultstring`, `faultactor`, and `detail`.	This check only applies to response messages.
3.3.3 SOAP Fault Namespace Qualification	**R1001** When an Envelope is a Fault, the element children of the `soap:Fault` element must be unqualified.	This check only applies to response messages.
3.4.6 HTTP Client Error Status Codes	**R1113** An instance should use a "`400 Bad Request`" HTTP status code if a HTTP request message is malformed. **R1114** An instance should use a "`405 Method not Allowed`" HTTP status code if a HTTP request message is malformed. **R1125** An instance must use a 4xx HTTP status code for a response that indicates a problem with the format of a request.	Only applies to responses for a proxy service where you cannot influence the status code returned due to errors in the request.
3.4.7 HTTP Server Error Status Codes	**R1126** An instance must return a "`500 Internal Server Error`" HTTP status code if the response envelope is a fault.	This check applies differently to request and response messages. For request messages, any faults generated have a `500 Internal Server Error` HTTP status code. For response messages, an error is generated if fault responses are received that do not have a `500 Internal Server Error` HTTP status code.
4.7.19 Response Wrappers	**R2729** An envelope described with an rpc-literal binding that is a response must have a wrapper element whose name is the corresponding `wsdl:operation` name suffixed with the string `Response`.	This check only applies to response messages. Oracle Service Bus never generates a non-fault response from a proxy service.

Table 37–12 (Cont.) Oracle Service Bus WS-I Compliance Checks

Check	WS-I Basic Profile Details	Oracle Service Bus Description
4.7.20 Part Accessors	**R2735** An envelope described with an rpc-literal binding must place the part accessor elements for parameters and return value in no namespace. **R2755** The part accessor elements in a message described with an rpc-literal binding must have a local name of the same value as the name attribute of the corresponding `wsdl:part` element.	This check applies to request and response messages. Any request error messages generate a `soap:client` error.
4.7.22 Required Headers	**R2738** An envelope must include all `soapbind:headers` specified on a `wsdl:input` or `wsdl:output` of a `wsdl:operation` of a `wsdl:binding` that describes it.	This check applies to request and response messages. Any request error messages generate a `soap:client` error.
4.7.25 Describing SOAPAction	**R2744** A HTTP request message must contain a SOAPAction a HTTP header field with a quoted value equal to the value of the soapAction attribute of `soap:operation`, if present in the corresponding WSDL description. **R2745** A HTTP request message must contain a `SOAP` action a HTTP header field with a quoted empty string value, if in the corresponding WSDL description, the SOAPAction of `soapbind:operation` is either not present, or present with an empty string as its value.	This check applies to request messages and a `soap:client` error is returned.

37.17 Converting Between SOAP 1.1 and SOAP 1.2

Oracle Service Bus supports SOAP 1.1 and SOAP 1.2. A SOAP 1.1 proxy service can invoke a SOAP 1.2 business service or vice versa. However, due to differences between SOAP 1.1 and 1.2, Oracle Service Bus cannot automatically convert between the two in every situation. Use the following guidance to ensure proper conversion between SOAP 1.1 and 1.2:

- The SOAP namespace is automatically changed by Oracle Service Bus before invoking the business service. If a fault comes back from the business service it is automatically changed to the SOAP version of the proxy service. It is, however, up to the pipeline actions to map the SOAP header-related XML attributes (like `MustUnderstand`) between the two versions. It is also up to the pipeline actions to change the SOAP encoded namespace for encoded envelopes.

- Automatic conversion from SOAP 1.1 to SOAP 1.2 will work correctly only if the payload uses doc/ or rpc/literal encoding.

- In SOAP 1.1, the encodingStyle attribute is allowed on any element in the envelope. In SOAP 1.2, the encodingStyle attribute is allowed only on child elements of the Body. Automatic conversion from SOAP 1.1 to SOAP 1.2 may result in an invalid envelope if the encodingStyle attribute in SOAP 1.1 is present outside of child elements of Body, Header, and Fault detail. You must perform SOAP conversion in the proxy service pipeline to ensure a valid envelope.

- If the SOAP 1.1 and SOAP 1.2 services use different encoding styles, such as rpc/encoded to doc/literal, you must perform SOAP conversion in the proxy service pipeline.

38

Improving Service Performance with Split-Join

This chapter provides an overview of Split-Joins and illustrates static and dynamic Split-Join scenarios.

Oracle Service Bus's advanced mediation feature, called Split-Join, helps you improve service performance by concurrently processing individual messages in a request. This topic, which describes Split-Join, includes the following sections:

- Section 38.1, "Introduction to Split-Join"
- Section 38.2, "Developing Split-Joins"

38.1 Introduction to Split-Join

Oracle Service Bus's Split-Join feature lets you split a service payload, such as an order, into individual messages for concurrent processing. Concurrent processing, as opposed to sequential processing, greatly improves service performance. Split-Join achieves this task by splitting an input message payload into sub messages (split), routing them concurrently to their destinations, and aggregating the responses into one overall return message (join). This process of payload splitting and response aggregation is called a Split-Join pattern.

Split-Joins are particularly useful for optimizing overall response times in scenarios where payloads delivered by faster systems are being directed to responding services on slower systems. Without Split-Join, individual messages in a payload are normally resolved in sequential order by the recipient, which can take a long time if the responding system is slow. (The overall response time is the sum of the individual response times for each message.) With Split-Join, multiple messages are processed simultaneously, which reduces burden on the responding system and greatly enhances response times. (The overall response time is roughly that of the longest individual message's response time plus some minor system overhead.)

There are two patterns supported by the Split-Join feature:

- Section 38.1.1, "Static Split-Join"
- Section 38.1.2, "Dynamic Split-Join"

38.1.1 Static Split-Join

The static Split-Join branches from the main execution thread of an Oracle Service Bus message flow by splitting a payload into a fixed number of new branches according to the configuration of the Split-Join. At design time you determine the number and variety of services to be invoked.

38.1.1.1 Static Split-Join – Sample Scenario

A telco company might want to employ static Split-Join when processing a customer's order for a communications services package. In this case, the customer might sign up for DSL and voice services all at once. Rather than executing each request in the payload separately in order, the telco can execute the messages in parallel using a callout from the Oracle Service Bus message flow to a Split-Join employing the static Split-Join pattern.

Static Split-Join is the ideal pattern in this case because the developer knows there will always be exactly two incoming service requests for this particular service package: DSL and voice. Splitting the requests into parallel branches allows them to be processed concurrently, which improves the overall response time for processing the payload. After all messages have been processed, the generated responses are aggregated back into one reply in the execution thread.

Figure 38–1 illustrates a static Split-Join that splits two known service requests, DSL activation and phone activation, processes each request in parallel, and joins the responses into a single reply.

Figure 38–1 Static Split-Join – Known number of service requests
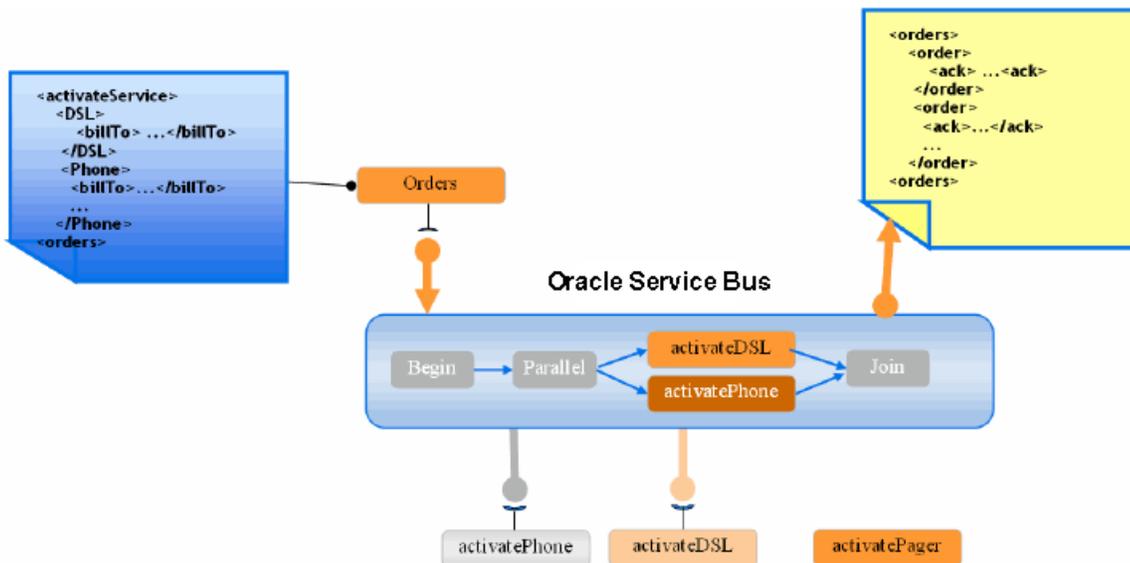
38.1.2 Dynamic Split-Join

The dynamic Split-Join branches from the main execution thread of an Oracle Service Bus message flow by dynamically creating new branches according to the contents of the incoming payload. The dynamic Split-Join uses conditional logic to determine the number of branches to create. All requests are handled simultaneously, and the responses are aggregated into a single reply.

38.1.2.1 Dynamic Split-Join – Sample Scenario

A company might want to use dynamic Split-Join when placing automated stationery orders for its employees. If the orders are automatically placed every week based on employee submissions, there is no way of knowing how many individual orders are placed in any one weekly order. Rather than placing each order separately, the company could use dynamic Split-Join to place the orders concurrently using a callout

from the Oracle Service Bus message flow to a Split-Join employing the dynamic Split-Join pattern.

Dynamic Split-Join is the ideal pattern in this case, because the developer has no way of knowing how many orders will be submitted each week. The dynamic Split-Join loops through all the orders and places them in parallel. The developer can also limit the number of orders processed. After all of the orders have been processed, the generated order responses are aggregated back into one reply in the execution thread.

Figure 38–2 illustrates a dynamic Split-Join that splits 15 orders, processes them concurrently, and joins the responses into a single reply.

Figure 38–2   Dynamic Split-Join – Unknown number of service requests
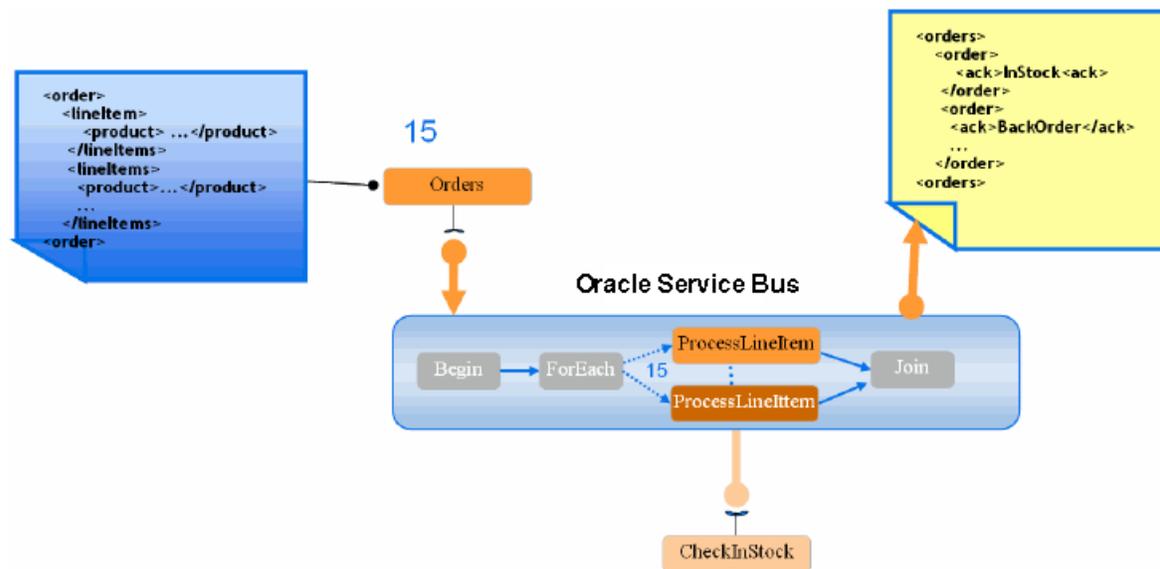
38.1.3 Split-Join Framework

A Split-Join, which takes the form of a .flow file in Eclipse, is based on a WSDL operation. The Split-Join is wrapped in a WSDL-based business service that communicates across a FLOW transport, which is a dedicated transport for Split-Joins. The business service is invoked from a proxy service. Figure 38–3 illustrates the Split-Join framework.

Figure 38–3   Split-Join framework
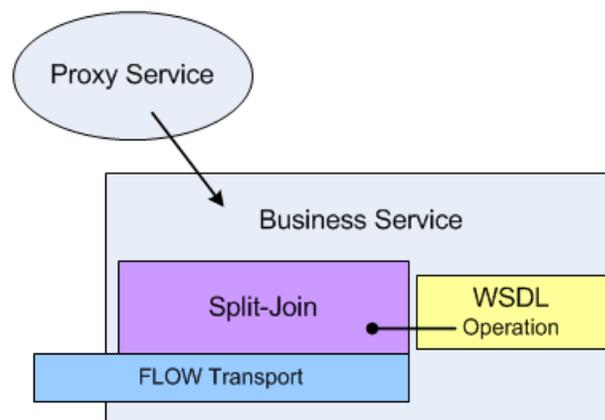
38.2 Developing Split-Joins

You create and configure Split-Joins in Eclipse, then import them into the Oracle Service Bus Administration Console for use in runtime configuration. For information on developing Split-Joins, see "Working with Split-Join" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

38.2.1 Split-Join Resource Type and Environment Variable

If you reference Split-Joins in any scripts or custom code, use the following values:

- **typeId** – FLOW

- **Work manager environment value type** – Split-Join Work Manager (Dispatch Policy)

39

Message Context

This chapter describes the Oracle Service Bus (OSB) message context model and the predefined context variables that are used in message flows.

This chapter includes the following sections:

- Section 39.1, "The Message Context Model"
- Section 39.2, "Predefined Context Variables"
- Section 39.3, "Message-Related Variables"
- Section 39.4, "Inbound and Outbound Variables"
- Section 39.5, "Operation Variable"
- Section 39.6, "Fault Variable"
- Section 39.7, "Initializing Context Variables"
- Section 39.8, "Performing Operations on Context Variables"
- Section 39.9, "Constructing Messages to Dispatch"
- Section 39.10, "Message Context Schema"

39.1 The Message Context Model

Oracle Service Bus message context is a set of properties that hold message content as well as information about messages as they are routed through Oracle Service Bus. These properties are referred to as context variables—for example, service endpoints are represented by predefined context variables. Oracle Service Bus also supports user-defined context variables.

The message context is defined by an XML schema. You typically use XQuery expressions to manipulate the context variables in the message flow that defines a proxy service.

39.2 Predefined Context Variables

Table 39–1 describes the predefined context variables. The predefined context variables can be grouped into the following types: message-related variables, inbound and outbound variables, the operation variable, and the fault variable.

For information about the element types in the message context variables, see Section 39.10, "Message Context Schema."

Table 39–1 Predefined Context Variables in Oracle Service Bus

Context Variable[1]	Description	See Also...
header	For SOAP messages, `header` contains the SOAP header. (If the proxy service is SOAP 1.2, `header` contains a SOAP 1.2 Header element.)	Section 39.3, "Message-Related Variables"
	For message types other than SOAP, `header` contains an empty SOAP header element.	
body	For the following cases:	Section 39.3, "Message-Related Variables"
	■ SOAP messages—contains the `<SOAP:Body>` part extracted from the SOAP envelope. (If the proxy service is SOAP 1.2, the `body` variable contains a SOAP 1.2 Body element.)	
	■ Non-SOAP, non-binary messages—contains the entire message content wrapped in a `<SOAP:Body>` element.	
	■ Binary messages—contains a `<SOAP:Body>` wrapped reference to an in-memory copy of the binary message.	
	■ Java objects—contains a `<SOAP:Body>` wrapped reference to an in-memory copy of the Java object.	
attachments	Contains the MIME attachments for a given message.	Section 39.3, "Message-Related Variables"
inbound	Contains:	Section 39.4, "Inbound and Outbound Variables"
	■ Information about the proxy service that received a message	
	■ The inbound transport headers	
outbound	Contains:	Section 39.4, "Inbound and Outbound Variables"
	■ Information about the target service to which a message is to be sent	
	■ The outbound transport headers	
operation	Identifies the operation that is being invoked on a proxy service.	Section 39.5, "Operation Variable"
fault	Contains information about errors that have occurred during the processing of a message.	Section 39.6, "Fault Variable"

[1] The Message Context Schema specifies the element types for the message context variables.

39.3 Message-Related Variables

Together, the message-related variables `header`, `body` and `attachments` represent the canonical format of a message as it flows through Oracle Service Bus. These variables are initialized using the message content received by a proxy service and are used to construct the outgoing messages that are routed or published to other services.

If you want to modify a message as part of processing it, you must modify these variables.

A message payload (that is, a message content exclusive of headers or attachments) is contained in the `body` variable. The decision about which variable's content to include

in an outgoing message is made at the point at which a message is dispatched (published or routed) from Oracle Service Bus.

That determination is dependent upon whether the target endpoint is expecting a SOAP or a non-SOAP message:

- When a SOAP message is expected, the `header` and `body` variables are combined in a SOAP envelope to create the message.

- When a non-SOAP message is expected, the contents of the Body element in the `body` variable constitutes the entire message.

- In either case, if the service expects attachments, a MIME package is created from the resulting message and the `attachments` variable.

39.3.1 Header Variable

The `header` variable contains SOAP headers associated with a message. The `header` variable points to a `<SOAP:Header>` element with headers as sub-elements. (If the proxy service is SOAP 1.2, the `header` variable contains a SOAP 1.2 Header element.) In the case of non-SOAP messages or SOAP messages with no headers, the `<SOAP:Header>` element is empty, with no sub-elements.

39.3.2 Body Variable

The `body` variable represents the core message payload and always points to a `<SOAP:Body>` element. (If the proxy service is SOAP 1.2, `body` contains a SOAP 1.2 Body element.) The core payload for both SOAP and non-SOAP messages is available in the same variable and with the same packaging—that is, wrapped in a `<SOAP:Body>` element:

- In the case of SOAP messages, the SOAP body is extracted from the envelope and assigned to the `body` variable.

- In the case of non-SOAP, non-binary, messages, the full message contents are placed within a newly created `<SOAP:Body>` element.

- In the case of binary messages, rather than inserting the message content into the `body` variable, a `<binary-content/>` reference element is created and inserted into the `<SOAP:Body>` element. To learn how binary content is handled, see Section 39.3.4, "Binary Content in the body and attachments Variables."

- In the case of Java objects, a `<java-content/>` reference element is created and inserted into the `<SOAP:Body>` element. To learn how Java content is handled, see Section 39.3.5, "Java Content in the body Variable."

39.3.3 Attachments Variable

The `attachments` variable holds the attachments associated with a message. The `attachments` variable is defined by an XML schema. It consists of a single root node: `<ctx:attachments>`, with a `<ctx:attachment>` sub-element for each attachment. The sub-elements contain information about the attachment (derived from MIME headers) as well as the attachment content. As with most of the other message-related variables, `attachments` is always set, but if there are no attachments, the `attachments` variable consists of an empty `<ctx:attachments>` element.

Each attachment element includes a set of sub-elements, as described in Table 39–2.

Table 39–2   Sub-Elements of the Attachments Variable

Elements of the Attachments Variable	Description[1]
Content-ID	A globally-unique reference that identifies the attachment.The type is `string`.
Content Type	Specifies the media type and sub-type of the attachment. The type is `string`.
Content-Transfer-Encoding	Specifies how the attachment is encoded. The type is `string`.
Content-Description	A textual description of the content. The type is `string`.
Content-Location	A locally-unique URI-based reference that identifies the attachment. The type is `string`.
Content-Disposition	Specifies how the attachment should be handled by the recipient. The type is `string`.
body	Holds the attachment data. The type is `anyType`.

[1] The Message Context Schema specifies the element types for the message context variables.

With the exception of the untyped `body` element, all other elements contain string values that are interpreted in the same way as they are interpreted in MIME—for example, valid values for the `Content-Type` element include `text/xml` and `text/xml; charset=utf-8`.

The parsing of attachments is not recursive. If an attachment has a `Content-Type` of `multipart/...`, the `body` element holds the original unpacked MIME content as a stream of bytes and does not contain attachment sub-elements. Because the MIME stream may contain binary data, it is represented by a `<binary-content>` reference element.

To learn how binary content is handled, see Section 39.3.4, "Binary Content in the body and attachments Variables."

Messages whose `Content-Type` is `multipart/form-data` are constructed at runtime as follows:

- Inbound: All parts of a received inbound `multipart/form-data` type message are assigned to the `$attachments` variable. The `$body` variable is left empty.

- Outbound: The content of an outbound `multipart/form-data` type message is built from the content of the `$attachments` variable. Nothing from `$header` or `$body` is included.

> **Note:**   If the inbound message is of a different `multipart` type than `multipart/form-data` (for example, `multipart/related`) and the outbound message is `multipart/form-data`, you must explicitly preserve the headers and content of the inbound root part, because they will not otherwise be passed through.

Attachments are supported on inbound requests and on outbound responses (that is, in messages received by a proxy service) only when the transport is HTTP, HTTPS or email. Attachments are supported for all transport types for outbound requests and inbound responses (that is for messages sent by a proxy service).

Oracle Service Bus does not support sending attachments to EJB-, Tuxedo-, and DSP-based services.

39.3.4 Binary Content in the body and attachments Variables

In the case of both the `body` and `attachments` variables, `text-`, `XML-` and `MFL`-based content is placed directly inside of an XML element. For binary data, which can contain byte values that are illegal in XML, Oracle Service Bus does not place the binary content in the XML element. Consequently, the binary content cannot be manipulated, but it is handled efficiently.

When binary content is received, the Oracle Service Bus runtime stores it in an in-memory hash table and a reference to that content is inserted into the XML (`body` or `attachments`) element. This reference is represented by the following XML snippet:

```
<binary-content ref="..."/>
```

where the ref attribute contains a URI or URN that uniquely identifies the binary content. This XML can be manipulated in an Oracle Service Bus pipeline, branch, or route node in the same way any other content can be manipulated, but only the reference and not the underlying binary content is affected.

For example:

- Binary content in the `body` variable can be copied to an attachment by copying the reference XML to the `body` sub-element of an attachment element.

- Binary content in two different attachments can be swapped by swapping the snippets of reference XML or by swapping the values of the ref attributes.

When messages are dispatched from Oracle Service Bus, the URI in the reference XML is used to restore the relevant binary content in the outgoing message. For information about how outbound messages are constructed, see Section 39.9, "Constructing Messages to Dispatch."

Clients and certain transports, notably email, file and FTP can use this same reference XML to implement pass-by-reference. In this case, the transport or client creates the reference XML rather than the proxy service runtime. Also, the value of the URI in the `ref` attribute is specified by the user that creates the reference XML. For these cases in which the reference XML is not created by the proxy service runtime—specifically, when the URI is not recognized as one referring to internally managed binary content—Oracle Service Bus does not de-reference the URI, and the content is not substituted into an outgoing message.

39.3.4.1 Sending SOAP with Attachments to Business Processes

When you send a SOAP with Attachments (SwA) document from Oracle Service Bus to a BPEL business process and use the XPath function `ora:getAttachmentContent('inputVariable','bin','/bin')` to retrieve the multipart attachments in BPEL, make sure to do the following:

- Set the `Content-ID` attachment variable for the multipart attachments.

- Refer to the attachments in the primary soap envelope using an `href` attribute.

Note that in some cases, the `cid:` information might be absent from the `href` attribute.

Below is an example:

```
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
        start="<rootpart@example.com>"
Content-Description: This is the optional message description.
--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@example.com>
```

```
<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body xmlns:types="http://example.com/mimetypes">
    <m:SendClaim xmlns:m="http://example.com/mimewsdl">
      <ClaimDetail>
           <Name>...</Name>
           <!-- additional claim details -->
      </ClaimDetail>
      <ClaimPhoto href="cid:4d7a5fa2-14af-451c-961b5c3abf786796@example.com"/>
    </m:SendClaim>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <4d7a5fa2-14af-451c-961b5c3abf786796@example.com>

...MIME attachment of binary photograph...
--MIME_boundary—
```

39.3.5 Java Content in the body Variable

The Oracle Service Bus pipeline supports Java objects as inputs and outputs to Java callout actions. A POJO returned by a Java callout is cached in the pipeline, and its key is returned wrapped in an XML message of the form `<java-content ref="cid:kkkkeeeeyyyy"/>`, where *cid:kkkkeeeeyyyy* is a key automatically generated by the producing action and used to index the object in the pipeline's cache. Any subsequent action then passes that XML unmodified as an argument.

The content of a POJO variable is not directly accessible by pipeline actions at configuration time. Rather, the content can be handled in the following ways:

- The content's metadata (that is, its key) can be handled as any other XML, for example in an XQuery such as `$pojo/java-content/@ref`. This may be useful for logging or debugging, but the content of the object cannot be directly accessed.

- The content can be assigned to a new variable that automatically becomes typed (in the pipeline) as a POJO. The object itself is not touched. The `<java-content...../>` XML snippet is copied from the source variable to the target variable.

- The content can be passed to another appropriate action (like Java callout) as a variable (for example, `$pojo`). The object itself is not touched. The argument is automatically de-referenced to the actual object.

The Java object is removed from the pipeline's cache when you delete all variables holding the object's key (in `<java-content />`) or when you delete all XPaths pointing to the `<java-content />` snippet.

39.3.6 Streaming body Content

For processing message content, you can specify that the Oracle Service Bus pipeline streams the content rather than loading it into memory. When you enable content streaming for a proxy service, you specify whether to buffer the streamed content to memory or a disk file as an intermediate step during the processing of the message. The creation of these temporary files might affect performance. For information about protecting temporary files, see "Protection of Temporary Files With Streaming body

Content" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

When you enable the streaming option, content streaming applies *only* to the `body` variable.

In general, use content streaming:

- When processing large content messages. See the guidelines in Section 39.3.6.1, "Best Practices for Using Content Streaming."

- In use cases where Oracle Service Bus accesses the payload a small number of times.

- For content-based routing without transformations; content streaming results in better performance due to the benefits from partial parsing.

39.3.6.1 Best Practices for Using Content Streaming

Use the following guidelines and recommendations:

- When you enable streaming for large message processing, you cannot use the insert, replace, rename, for each, validate, and delete actions with respect to the `body` message context variable, because these actions require the input variable to be fully materialized in memory and full materialization is incompatible with the content streaming option.

- You can use the results of an XQuery or XSL transformation from a very large `$body` with these pipeline actions:

  - Assign, insert, and replace actions—to update the value of another context variable (not `$body`). However, you must ensure that the result of the expression is small enough to be fully materialized and stored in the message context.

  - Java callouts—to pass input arguments. All input to Java callouts is fully materialized, therefore, the results of expressions used as input must be small enough to be fully materialized.

  - MFL transformations—to transform very large payloads without first materializing the input as an XML Bean. When using a very large `$body` as an input to an MFL transformation, declare a messaging service, binary message type proxy service. If you declare a messaging service, text message type proxy service, `$body` will get fully materialized to obtain an input stream for the transformation.

  - Alert, log, and report actions—to report the result of an XQuery or XSL transformation on a very large `$body`.

  - Service callouts

- For XSL transformations, all input is fully materialized in order to perform the transformation, therefore, you must ensure that the input is small enough so that it can be fully materialized and processed by the XSLT processor.

- With very large MFL input, you should use a MFL service instead of a MFL stage action to perform a MFL-to-XML transformation.

- Do not use the Test Console to test proxy services with very large content messages because the content will be fully materialized, potentially causing an OOM exception, and displayed, causing a slowdown in the Test Console window.

- When writing XQueries, use proper indexing to achieve partial parsing.

For example, instead of writing `$body/*:DateTimeStruct`, which would consume the entire input stream, write:

```
($body/*:DateTimeStruct)[1] or $body[1]/*:DateTimeStruct[1]
```

By using indexing, only content up to and including the first `DateTimeStruct` element will be parsed.

■ Because each variable that is accessed by two or more consumers (expressions) is materialized, when writing XQueries, avoid statements such as:

```
let $labdata1 := $body/*
return <HEADER>{ $labdata1/HEADER/@*, $labdata1/HEADER/node() }</HEADER>
```

In this case, `$labdata1` is bound to the whole document without the root element so the XQuery engine runs out of memory when trying to materialize it.

One way of changing this query to avoid excessive materialization would be to move the `/HEADER` path expression inside the `let` clause.

```
let $labdata1 := $body/*/HEADER
...
```

In this case, the XQuery engine will only materialize the `HEADER` element or elements.

Another way to avoid materialization would be to use the `fn-bea:rename()` function in which you can rename elements in a streaming mode.

```
fn-bea:rename($oldelements as element()*, $newname as element()) as element()*
```

For example:

```
fn-bea:rename($body/*/HEADER, <HEADER_NEW/>)
```

■ At runtime, processing large messages is subjected to the limitations and restrictions of the underlying transport; for example, the message size handling limitations of the transport. Be aware of the JVM and RMI settings that limit the capacity of the transport to accept large messages.

39.3.7 Streaming Attachments

For processing message attachments, you can specify that Oracle Service Bus page MIME attachments to disk and then stream the contents rather than buffering the attachments in memory and parsing them into XML. This is particularly advantageous when working with large attachments.

Using this approach, Oracle Service Bus buffers only the headers and exposes the rest of the message payload as a stream from which smaller portions can be read at a time. Streamed transfers can improve the scalability of a service by eliminating the need for large memory buffers.

Oracle Service Bus supports streaming MIME attachments for proxy and business services using the following transports:

■ HTTP/S

■ Local (when chained through an HTTP proxy with streaming attachments enabled, inheriting the paged attachments setting)

Note: Oracle Service Bus does not support streaming attachments for Email, SB, or WS transports.

When enabled for HTTP proxy services, the setting applies to the handling of inbound request messages. For HTTP business services, the setting applies in the handling of outbound response messages. The following actions that allow outbound messages to be dispatched support the streaming of attachments:

- Route, Dynamic Route, and Route Table

- Publish, Dynamic Publish, and Publish Table

When streaming is enabled, all attachments, including binary, text, or XML, are processed as opaque data. This means that when streaming attachments are enabled, you cannot run XQueries/XPath expressions based on the XML content of the attachment.

39.3.7.1 Inbound Message Handling

When Oracle Service Bus receives an inbound message and is configured to stream attachments, the contents of each MIME attachment is saved to a separate file on disk.

The value of the `$attachments` variable is then set such that standard MIME headers for each attachment, including Content-ID, Content-Type, Content-Transfer-Encoding, Content-Description, Content-Location, and Content-Disposition, when present, are added under the attachment element.

The `body` child element then has a single `binary-content` child element that refers to the corresponding source in the source repository.

For example, an $attachments variable might appear as follows:

```
<con:attachments xmlns:con="http://www.oracle.com/wli/sb/context">
    <con:attachment>
        <con:Content-Type>image/jpeg</con:Content-Type>
        <con:Content-ID>
            &lt;1.urn:uuid:BFB7D745CBAF21BA5B12023554608963@apache.org>
        </con:Content-ID>
        <con:Content-Transfer-Encoding>
            binary
        </con:Content-Transfer-Encoding>
        <con:body>
            <con:binary-content ref="cid:23976580:1183dd6aab9:-7fe0"/>
        </con:body>
    </con:attachment>
</con:attachments>
```

This is done regardless of the attachment content type. This means that `text/xml` attachments, for example, are treated identically to `image/jpeg` attachments, namely as opaque binary data with a binary content element.

This style of message handling differs from when attachments are not streamed, in which case certain types, such as `text/xml` or `text/plain` are recognized and used to initialize the body XML element (to contain the XML contents) or the text contents of the attachment respectively.

39.3.7.2 Outbound Message Handling

When Oracle Service Bus processes an outbound message and is configured to stream attachments, the contents of each MIME attachment is saved to a separate file on disk. Oracle Service Bus then initializes the `$attachments` message context variable in a manner similar to inbound requests (see Section 39.3.7.1, "Inbound Message Handling").

39.3.8 Understanding XOP/MTOM Support

Oracle Service Bus enables you to configure proxy services to decode and parse inbound messages in XOP/MTOM format and to send responses using the XOP/MTOM format, when appropriate. Service Bus also enables you to configure business services to encode outbound messages in XOP/MTOM format.

Oracle Service Bus supports proxy services with the following binding types to accept and decode incoming XOP/MTOM payloads:

- Any XML

- Messaging (XML)

- Any SOAP

- WSDL-based

Proxy services of any other service binding type that receive XOP/MTOM payloads treat them the same as any other MIME multipart request without MTOM-specific handling.

In addition, the following Oracle Service Bus transports support XOP/MTOM:

- HTTP/S

- Local

- SB (when applicable, such as with chained Oracle Service Bus domains)

Oracle Service Bus supports all existing actions that allow outbound messages to be dispatched, including the following:

- Route, Dynamic Route and Route Table

- Publish, Dynamic Publish and Publish Table

- Service Callout

Oracle Service Bus supports Oracle WSM message protection and non-message protection policies with XOP/MTOM-enabled SOAP services. However, Oracle WSM message protection policies that enable signing, encryption, or both for SOAP with Attachments (SwA) are not compatible with MTOM encoded messages, so these policies are not supported with XOP/MTOM.

39.3.8.1 Configuring Proxy Services

You can enable proxy services to decode and parse inbound messages in XOP/MTOM format and to send responses using the XOP/MTOM format, when appropriate. When XOP/MTOM Support is enabled, you can further select how to handle binary data in the $body message context variables from among the following options:

- **Include Binary Data by Reference**: (Default) See Section 39.3.8.1.1, "Understanding the Binary by Reference Option."

- **Include Binary Data by Value**: See Section 39.3.8.1.2, "Understanding the Binary by Value Option."

If XOP/MTOM Support is enabled for a proxy service, it is not required that every inbound message be in the MTOM format. Instead, this setting specifies that when an MTOM-formatted message arrives, the proxy service should handle it accordingly. Note also that when a proxy service that is not enabled for XOP/MTOM support receives an MTOM-formatted message, the service rejects the message and issues a runtime error.

The following sections describe proxy service behavior when using the binary data options. For information about how this behavior changes when a proxy service has XOP/MTOM enabled and uses Oracle WSM inbound request enforcement policies, see "Oracle WSM Policies and MTOM-Formatted Messages" in Chapter 50, "Securing Oracle Service Bus with Oracle Web Services Manager," of *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

39.3.8.1.1 Understanding the Binary by Reference Option Use the **Include Binary Data by Reference** option when you need direct access to binary data, for example to pass data to a Java callout or Message Format Language (MFL) transform.

When the **Binary by Reference** option is selected, Oracle Service Bus parses the root of the inbound message checking for the presence of `xop:Include` tags. These tags, when found, are converted to `ctx:binary-content` elements with a reference pointing to the corresponding source in binary repository. The resulting document is represented by the `$body` message context variable. The `$attachments` message context variable, in contrast, does not contain any information (is null).

This means that when pipeline actions access the contents of the `$body` message context variable, the actions do not encounter `xop:Include` elements, but instead work with `ctx:binary-content` elements.

When the proxy service needs to send a response back to the client, the binding layer creates an XOP/MTOM package response by replacing `ctx:binary-content` references with `xop:Include` tags in the root of the message and adding a separate MIME part for each corresponding binary content reference.

39.3.8.1.2 Understanding the Binary by Value Option Use **Include Binary Data by Value** in the following cases:

- To bridge between MTOM and non-MTOM services. For example, consider an MTOM-enabled proxy service that receives a request that is then routed to a non-MTOM-enabled service. You could use this option to comply with existing standards for sending binary data in XML in Base64-encoded form.

- To validate the contents of the message against an XML schema that requires a base64binary element to be used in place of binary data.

When the **Binary by Value** option is selected, Oracle Service Bus parses the root of the inbound MIME message checking for the presence of `xop:Include` tags. When found, the body of the corresponding MIME parts (the binary data) are Base64-encoded and the resulting text replaces the `xop:Include` tags in the `$body` message context variable. The $attachments message context variable, in contrast, does not contain any information (is null).

39.3.8.2 Configuring Business Services

You can enable business services to encode outbound messages in XOP/MTOM format. When XOP/MTOM Support is enabled, you can further select how to handle binary data in the $header and $body message context variables from among the following options:

- **Include Binary Data by Reference**: (Default) In an outbound response message, replace `xop:Include` elements with `ctx:binary-content` elements when setting up the $body message context variable.

- **Include Binary Data by Value**: In an outbound response message, replace `xop:Include` elements with Base64-encoded text versions of corresponding binary data when setting up the $body message context variable.

If XOP/MTOM Support is enabled for a business service, it is not required that every outbound message be in the MTOM format. Instead, this setting specifies that the business service is capable of handling an MTOM payload.

The following sections describe business service behavior when using the binary data options. For information about how this behavior changes when a business service has XOP/MTOM enabled and uses Oracle WSM policies, see "Oracle WSM Policies and MTOM-Formatted Messages" in Chapter 50, "Securing Oracle Service Bus with Oracle Web Services Manager," of *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

39.3.8.2.1 Handling Outbound Messages When XOP/MTOM support is enabled for a business service, Oracle Service Bus examines the contents of $body message context variable searching for `ctx:binary-content` elements. If any are present, Oracle Service Bus creates an XOP/MTOM MIME package replacing `ctx:binary-content` with `xop:Include` elements and with the corresponding MIME part in the payload.

Oracle Service Bus always uses XOP/MTOM when it is enabled and the body has binary content, regardless of the size of the content (for example, even when it is smaller than 1KB).

Note also that since Oracle Service Bus does not support the combination of MTOM and SwA, the system throws a runtime exception when Oracle Service Bus needs to dispatch an outbound request to a business service and the following conditions are met:

- The business service is XOP/MTOM enabled
- The $attachments message context variable is not null

39.3.8.3 XOP/MTOM Attachments Streaming

With XOP/MTOM enabled on a service, Oracle Service Bus automatically modifies elements in the $header and $body of a message, as described in the previous subsections. Because of the need to modify XOP/MTOM message content, Oracle Service Bus does not support streaming the message $header and $body.

However, there may be situations in which you need to stream the binary attachments of XOP/MTOM messages directly to disk rather than to the Oracle Service Bus heap (the default), such as when attachments are large and cause out-of-memory errors.

> **Note:** XOP/MTOM attachments are not sent in the $attachments variable of the message. They are included or referenced in the message $body as previously described.

To stream XOP/MTOM message attachments directly to disk, use the following settings on your service configuration:

- Use the XOP/MTOM Include Binary Data by Reference option. (The Include Binary Data by Value option puts the binary data directly in the $body, which cannot be streamed for XOP/MTOM support.)

- With Include Binary Data by Reference selected, you can also select the Page Attachments to Disk option (HTTP transport only). With Page Attachments to Disk selected, Oracle Service Bus generates the XOP/MTOM message to reference the binary attachment on disk rather than in memory.

> **Note:** Referencing attachments in memory is recommended for optimum performance, so use the Page Attachments to Disk option only when you have a specific need for doing so, such as when sending large attachments whose size could cause out-of-memory errors.

39.4 Inbound and Outbound Variables

The `$inbound` and `$outbound` context variables contain information about the inbound and outbound endpoints. The `$inbound` variable contains information about the proxy service that received the request message; the `$outbound` variable contains information about the target business service to which a message is sent.

The `$outbound` variable is set in the route action in route nodes and publish actions. You can modify `$outbound` by configuring request and response actions in route nodes and by configuring request actions in publish actions.

> **Caution:** Some modifications that you can make for the `inbound` and `outbound` context variables are not honored at runtime. That is, the values of certain headers and metadata can be overwritten or ignored by the Oracle Service Bus runtime. The same limitations are true when you set the transport headers and metadata using the transport headers and service callout actions, and when you use the Test Console to test your proxy or business services.
>
> For information about the headers and metadata for which there are limitations, see Section 33.4, "Understanding How the Runtime Uses the Transport Settings in the Test Console." Note also that any modifications you make to `$outbound` in the message flow *outside* of the request or response actions in route nodes and publish actions are ignored. In other words, those modifications are overwritten when `$outbound` is initialized in the route nodes and publish actions.

You cannot modify the `outbound` variable in service callout actions.

The `inbound` and `outbound` variables have the following characteristics:

- Have the same XML schema—the `inbound` and `outbound` context variables are instances of the `endpoint` element as described in Section 39.10, "Message Context Schema."

- Contain a single `name` attribute that identifies the name of the endpoint as it is registered in the service directory. The `name` attribute should be considered read-only for both `inbound` and `outbound`.

  > **Caution:** The read-only rule is not enforced. Changing read-only elements can result in unpredictable behavior.

- Contain the `service`, `transport` and `security` sub-elements described in the following section.

Attachments are supported on inbound requests and outbound responses (that is, in messages received by a proxy service) only when the transport is HTTP, HTTPS or email.

Attachments are supported for all transport types for outbound requests and inbound responses (that is for messages sent by a proxy service).

Oracle Service Bus does not support sending attachments to EJB-, Tuxedo-, and DSP-based services.

39.4.1 Sub-Elements of the inbound and outbound Variables

This section describes the sub-elements of the inbound and outbound context variables, including information about whether a given sub-element is initialized at runtime. To learn about how context variables are initialized, see Section 39.7, "Initializing Context Variables." The sub-elements include:

- Section 39.4.1.1, "service"
- Section 39.4.1.2, "transport"
- Section 39.4.1.3, "security"

39.4.1.1 service

The service element is read-only for both inbound and outbound. Sub-elements include providerName and operation.

Table 39–3   Sub-Elements of the service Element

Sub-Elements[1]	Description...
providerName	Specifies the name of the service key provider.
	Initialized based on the configuration of publish and routing actions.
operation (outbound only)	Used in the outbound variable, specifies the name of the operation to be invoked on the target business service.
	Initialized based on the inbound and outbound.
	Note: This element is used for the outbound variable only. In the case of inbound messages, the name of the operation to be invoked on the proxy service is specified by the operation variable.

[1]   The Message Context Schema specifies the element types for the message context variables.

39.4.1.2 transport

The transport element is read-only on inbound, except for the response element, which you can modify to set the response transport headers. The sub-elements of the transport element are described in Table 39–4.

Table 39–4   Sub-Elements of the Transport Element

Sub-Elements[1]	Description...
uri	Identifies the URI of the endpoint:
	■ When used in the inbound variable, this is the URI by which the message arrived.
	■ When used in the outbound variable, this is the URI to use when sending the message—it overrides any URI value registered in the service directory.
	Initialization
	The URI element is initialized as follows:
	■ Always initialized on the inbound variable
	■ Never initialized on the outbound variable. You can set the URI on outbound when you want to override the set of URIs in the service configuration. URI failover is not supported if this element is set.
request This element is read-only[2] in the inbound variable. You can modify it for the outbound variable.	Specifies transport-specific metadata about the request (including transport headers). The value for this element is defined by the transport protocol (specifically, the RequestMetaData XML defined by the transport SDK).Therefore, the structure of this element depends on the transport being used.
	To learn about the transport-specific types for this element, see the appropriate transport schema, which are available in the following directory in your Oracle Service Bus installation:
	OSB_ORACLE_HOME/lib/transports/
	Initialization
	The URI element is initialized as follows:
	■ Initialized on the inbound variable using information from the request message received by Oracle Service Bus.
	■ On the outbound variable, the request element is created with the proper typing. The typing is transport-dependent. The request element is typically initialized as an empty element, with the exception of certain important transport headers—for example, content-type and SOAPAction.
	You can set a filename for an outbound message using the File transport protocol by configuring $outbound in a route node request action, as follows:
	■ If the fileName only is specified, a file of that name is stored at the location specified by the endpoint URI of the target business service.
	■ If isFilePath is set to true, the value of fileName is used as a relative path appended to the endpoint URI of the target business service. For example, if the endpoint URI is file:////apollo/ob/data, and the fileName header is set to ./myPath/myFile.xml, and isFilePath is set to true, the message will be stored at /apollo/ob/data/myPath/myFile.xml.
	If a file already exists with that name, a new name is generated, following the format *path/filename_random-number*.xml, where *random-number* is an integer in the range of 0 to 999999.

Table 39–4   (Cont.) Sub-Elements of the Transport Element

Sub-Elements[1]	Description...
response This element is read-only in the `outbound` variable. You can modify it for the `inbound` variable.	Specifies transport-specific metadata about the response (including transport headers). The value for this element is defined by the transport protocol (specifically, the `ResponseMetaData` XML defined by the transport SDK).Therefore, the structure of this element depends on the transport being used. To learn about the transport-specific types for this element, see the appropriate transport schema, which are available in the following directory in your Oracle Service Bus installation: *OSB_ORACLE_HOME*/lib/transports/ **Initialization** The `URI` element is initialized as follows: ■ Initialized on the `outbound` variable using information from the response message received by Oracle Service Bus. ■ On the `inbound` variable, the `response` element is created with the proper typing. The typing is transport-dependent. The `response` element is typically initialized as an empty element, with the exception of certain important transport headers—for example, `content-type` and `SOAPAction`. For a description of the standard HTTP headers, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html For a description of the standard JMS headers, see "Value-Added Public JMS API Extensions" in *Oracle Fusion Middleware Programming JMS for Oracle WebLogic Server*. **Note**: The following MQ headers do not have equivalents in Oracle JMS: `ApplOriginData`, `ApplIdentityData`, `Accounting Token`
mode	Specifies whether the communication style is `request` (one-way) or `request-response` (two-way). **Initialization** Initialized on the `inbound` and `outbound` variables using information from the service and its operations (if applicable). For example, if a request-only operation is being invoked, the `mode` element is set to `request`, rather than to `request-response`.

Table 39–4   (Cont.) Sub-Elements of the Transport Element

Sub-Elements[1]	Description...
qualityOfService This element is read only for `inbound`. You can modify it for the `outbound` case— in the outbound request actions of a publish or routing action.	Specifies the quality of service expected when sending or receiving a message. Valid values include `best-effort` and `exactly-once`: ■ `best-effort` means that each dispatch defines its own transactional context (if the transport is transactional). Best effort means that there is no reliable messaging and no elimination of duplicate messages—however, performance is optimized. For the scenario in which a message is dispatched as a result of a publish action, any dispatch errors are suppressed. For the scenario in which a message is dispatched from a routing node, dispatch errors are not suppressed. ■ `exactly-once` means that the dispatch is included as part of the inbound transactional context (if one exists and if the outbound transport is transactional) and errors cause processing to abort and trigger the relevant error handler (in the case of both the route and publish scenarios). *Exactly once* reliability means that messages are delivered from inbound to outbound exactly once, assuming a terminating error does not occur before the outbound message send is initiated. **Initialization** The `qualityOfService` element is initialized on the inbound and outbound variables as follows: ■ In the inbound case, the quality of service (QoS) is dictated by the transport. For example, for the JMS/XA transport, the QoS is *exactly once*; for the HTTP transport, the QoS is *best effort*. ■ In the outbound case, the QoS is set differently for publishing and for routing, as follows: **Routing**—When messages are routed to another service from a route node, the QoS is always initialized using the value from the inbound context variable. In other words, the outbound QoS is set to *exactly once* if (and only if) the inbound QoS is *exactly once*. Otherwise, the outbound QoS is set to *best effort*. **Publishing**—When a message is published to another service as the result of a publish action, the quality of service (QoS) is always initialized to *best effort* regardless of the inbound setting.
retryCount (`outbound` only)	Specifies the number of retries to attempt when sending a message from Oracle Service Bus. If `retryCount` is set, the setting overrides any retry count value configured in the target service configuration.

[1]   The Message Context Schema specifies the element types for the message context variables.

[2]   The read-only rule is not enforced. Changing read-only elements can result in unpredictable behavior.

39.4.1.3 security

The sub elements of the `security` element are described in Table 39–5.

Table 39–5   Sub-Elements of the Security Element

Sub-Elements[1]	Description...
transportClient (`inbound` only, read only[2])	Specifies authenticated transport-level user information. The user information includes a username and any optional principals. The principals can themselves include zero or more groups, one for each group the subject belongs to. **Note**: If the subject is anonymous, then the username is "anonymous" and there are no groups. Initialized by Oracle Service Bus. The inbound `transportClient` element is read-only.
messageLevelClient (`inbound` only, read only[2])	Specifies authenticated message-level user information. The user information includes a username and any optional principals. The principals can themselves include zero or more groups, one for each group the subject belongs to. **Note**: If the subject is anonymous, then the username is "anonymous" and there are no groups. Initialized by Oracle Service Bus. The inbound `messageLevelClient` element is read-only.
doOutboundWss `outbound` only)	Oracle Service Bus sets the value of this element during routing or publishing. Some infrequently used design patterns set the value to `false` to preempt a proxy service from automatically generating the outbound WS-Security SOAP envelope. For more information, see "Disabling Outbound WS-Security" under "Configuring Message-Level Security for Web Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. **Note**: When one proxy service invokes another proxy service (such as a local proxy) that contains Oracle Web Services Manager service policies, outbound WS-Security processing does not occur. OSB handles that behavior automatically and does not use the doOutboundWss property. For more information, see "Using Oracle WSM Security with Local Proxies" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

[1] The Message Context Schema specifies the element types for the message context variables.
[2] The read-only rule is not enforced. Changing read-only elements can result in unpredictable behavior.

39.4.2 Related Topics

Chapter 21, "Proxy Services: Actions"

Section 22.8, "Adding Route Nodes to Message Flows"

For a description of the standard HTTP headers, see
http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

For a description of the standard JMS headers, see "Understanding WebLogic JMS" in *Oracle Fusion Middleware Programming JMS for Oracle WebLogic Server*.

39.5 Operation Variable

The `operation` variable is a read-only variable. It contains a string that identifies the operation to be invoked on a proxy service. If no operations are defined for a proxy service, the `operation` variable is not set and returns the equivalent of null.

Oracle Service Bus provides the `operation` variable as a stand-alone variable, rather than as a sub-element of the `inbound` variable to optimize performance—the

computation of the operation may be deferred until the `operation` variable is explicitly accessed rather than anytime the `inbound` variable is accessed.

39.6 Fault Variable

The `fault` variable is used to hold information about any error that has occurred during message processing. When an error occurs, this variable is populated with information before the appropriate error handler is invoked.

> **Note:** This variable is defined only in error handler pipelines and is not set in request and response pipelines, or in route or branch nodes.

The `fault` variable includes the `errorCode`, `reason`, `details`, and `location` sub-elements described in Table 39–6.

Table 39–6   Sub-Elements of the Fault Variable

Elements of the Fault Variables	Description[1]...
errorCode	Specifies the error code as a string value
reason	Contains a text description of the error
details	Contains user-defined XML content related to the error
location	Identifies the node, pipeline and stage in which the error occurred. Also identifies if the error occurred in an error handler. The sub-elements include: ■ node—the name of the pipeline, branch, or route node where an error occurred; a string. ■ pipeline—the name of the Pipeline where an error occurred (if applicable); a string. ■ stage—the name of the stage where an error occurred (if applicable); a string. ■ error-handler—indicates if an error occurred from inside an error handler; a boolean.

[1]   The Message Context Schema specifies the element types for the message context variables.

The contents of the `fault` variable are modeled after SOAP faults to facilitate fault generation when replying from a SOAP-based proxy service. The values for error codes generated by Oracle Service Bus correspond to system error codes and are prefixed with a BEA string.

The error codes associated with the errors surface inside the element of the `fault` context variable. You can access the value using the following XQuery statement:

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

Oracle Service Bus defines three generic error codes for the three classes of possible errors. The format of the generic codes is `BEA-xxx000`, where `xxx` represents a generic category as follows:

■ 380 Transport

■ 382 Proxy

■ 386 Security

- 394 UDDI

This yields the generic codes as follows:

- BEA–380000—BEA–380999

    Indicates a transport error (for example, failure to dispatch a message).

- BEA–382000—BEA–382499

    Indicates a proxy service runtime error (for example, a stage exception).

- BEA–382500—BEA–382999

    Indicates an error in a proxy service action.

- BEA–386000—BEA–386999

    Indicates a WS-Security error (for example, authorization failure).

- BEA–394500—BEA–394999

    Indicates an error in the UDDI sub system.

Oracle Service Bus defines unique codes for specific errors. For example:

BEA-382030—Indicates a message parse error (for example, a SOAP proxy service received a non-SOAP message).

BEA-382500—Reserved for the case in which a service callout action receives a SOAP Fault response.

For information about these and other specific error codes, see Appendix A, "Error Codes." See also Section 37.7, "Handling Errors in Message Flows."

39.7 Initializing Context Variables

The message context and its variables are initialized in the binding layer when a message is received and before message processing begins. Table 39–7 summarizes how context variables are initialized.

Table 39–7   Initializing Context Variables

Context Variable	How Initialized
fault	Initialized to null because no routing or errors have yet occurred.
outbound	Initialized in the route action in route nodes and in publish actions. You can modify $outbound through the request actions in routing nodes and publish actions (also in the response actions in routing nodes). For more information, see Section 39.4, "Inbound and Outbound Variables."
	For information about the initialization of sub-elements of outbound, see Section 39.4.1, "Sub-Elements of the inbound and outbound Variables."
inbound	Initialized with service, transport and security information that is obtained from Service Bus metadata about the registered proxy service and transport-level metadata (transport headers, authenticated user information, and so on) about the specific incoming request.
	For information about the initialization of sub-elements of inbound, see Section 39.4.1, "Sub-Elements of the inbound and outbound Variables."

Table 39–7   (Cont.) Initializing Context Variables

Context Variable	How Initialized
header body attachments operation	Initialized using the content of the inbound message. How the initialization is performed depends on the type of proxy service, as described in the subsequent topics in this section: ■ Section 39.7.1, "Initializing the attachments Context Variable" ■ Section 39.7.2, "Initializing the header and body Context Variables" The header, body, and attachments variables are re initialized after routing using the content of the response that is received. If no routing is performed or if the communication mode is request-only, then these variables are not re initialized. That is, they are not cleared of any content.

39.7.1 Initializing the attachments Context Variable

The attachments context variable is initialized with any MIME attachments that accompany the message, but does not include the part representing the main message (whether it is SOAP, XML, MFL, and so on). Each <attachment> element is initialized using the MIME headers that accompany each part in the MIME package.

The contents of the <body> element in the <attachment> can be one of the following depending on the attachment's Content-Type:

■ XML

■ text

■ A snippet of reference XML that refers to the attachment content (see Section 39.3.4, "Binary Content in the body and attachments Variables.")

39.7.2 Initializing the header and body Context Variables

This section describes how the initialization of header and body context variables is performed depending on the type of proxy service: Section 39.9.1, "SOAP Services," Section 39.9.2, "XML Services (Non SOAP)," Section 39.9.3, "Messaging Services."

39.7.2.1 SOAP Services

Messages to SOAP-based services are SOAP messages containing XML that is contained in a <soap:Envelope> element. In the case that messages include attachments, the content of the inbound message is a MIME package that includes the SOAP envelope as one of the parts—typically the first part or one identified by the top-level Content-Type header. The context variables are initialized as follows:

■ header—initialized with the <soap:Header> element from the SOAP message

■ body—initialized with the <soap:Body> element from the SOAP message

39.7.2.2 XML Services (Non SOAP)

The messages to XML-based services are XML, but can be of any type allowed by the proxy service configuration. In the case that messages include attachments, the content of the inbound messages is a MIME package that includes the primary XML payload as one of the parts—typically the first part or one identified by the top-level Content-Type header.

The context variables are initialized as follows:

■ header—initialized with an empty <soap:Header/> element.

■ body—initialized with a `<soap:Body>` element that wraps the entire XML payload.

39.7.2.3 Messaging Services

Messaging services are those that can receive messages of one data type and respond with messages of a different data type. The supported data types include XML, MFL, text, untyped binary. The context variables are initialized as follows:

■ header—initialized with an empty `<soap:Header/>` element.

■ body—initialized with a `<soap:Body>` element that wraps the entire payload.

    – In the case of XML, MFL, and text content, it is placed directly within the `<soap:Body>` element.

    – In the case of binary content, a piece of reference XML is created and inserted inside the `<soap:Body>` element (see Section 39.3.4, "Binary Content in the body and attachments Variables"). The binary content cannot be accessed or modified, but the reference XML can be examined, modified, and replaced with inline content.

39.8 Performing Operations on Context Variables

You interact with and manipulate the message context through actions in the pipelines, branch, or route nodes that define a proxy service. Most actions expose the XQuery language to do so. Each context variable is represented as an XQuery variable of the same name. For example, the header variable is accessible in XQuery as `$header`, the body variable is accessible as `$body`, and so on. The examples in this section show the use of XQuery to examine and manipulate context variables.

39.8.1 $body

The `$body` variable includes the `<soap-env:Body>...</soap-env:Body>` element. (If the proxy service is SOAP 1.2, the body variable contains a SOAP 1.2 Body element.)

For example, if you assign data to the body context variable using the Assign action, you must wrap it with the `<soap-env:Body>` element. In other words, you build the SOAP package by including the `<soap-env:Body>` element in the context variable.

There is an exception to this behavior in Oracle Service Bus—for the case in which you build the Request Document Variable for the service callout action. Service callout actions work with the core payload (RPC parameters, documents, and so on) and Oracle Service Bus builds the SOAP package around the core payload. In other words, when you configure the Request Document Variable for a service callout action, you do not wrap the input document with `<soap-env:Body>...</soap-env:Body>`.

For information about configuring the service callout action, see Section 21.6, "Adding Service Callout Actions."

39.8.2 $header

The `$header` variable includes the `<soap-env:Header>...</soap-env:Header>` element. (If the proxy service is SOAP 1.2, the header variable contains a SOAP 1.2 Header element.)

For example if you assign data to the header context variable using the Assign action, you must wrap it with the `<soap-env:Header>` element. In other words, you build the SOAP package by including the `<soap-env:Header>` element in the context variable. This is true for all manipulations of `$header`, including the case in which you can set

one or more SOAP headers for a service callout request. For information about configuring SOAP headers for a service callout action, see Section 21.6, "Adding Service Callout Actions."

Extract the WS-Addressing Header—From: `$header/wsa:From`

Extract the Payload From a Non-SOAP Message: `$body/*`

Extract the user-header From an Outbound Response Message:
`$outbound/ctx:transport/ctx:response/tp:user-header[@name='myheader']/@value`

When creating a `body` input variable that is used for the request parameter in a service callout to a SOAP Service, you would define that variable's contents using `body/*` (to remove the wrapper `soap-env:Body`), not `$body` (which results in keeping the `soap-env:Body` wrapper).

Assign Variable Contents for Request Parameter in a Service Callout: `$body/*`

39.8.3 Related Topics

For more information about handling context variables using the XQuery and XPath editors, see:

- Section 37.11, "Working with Variable Structures"
- Chapter 23, "Proxy Services: XQuery and XPath Editors"

39.9 Constructing Messages to Dispatch

When Oracle Service Bus publishes or routes a message, the content of the message is constructed using the values of variables in the message context. For example, transport headers and other transport-specific metadata are taken from `$outbound/transport/request`. As is the case with initialization of the context, the message content for outbound messages is handled differently depending upon the type of the target service. How the outbound message content is created depends on the type of the target service, as described in the following topics:

- Section 39.9.1, "SOAP Services"
- Section 39.9.2, "XML Services (Non SOAP)"
- Section 39.9.3, "Messaging Services"

39.9.1 SOAP Services

An outgoing SOAP message is constructed by wrapping the contents of the `header` and `body` variables inside a `<soap:Envelope>` element. If the invoked service is a SOAP 1.2 service, the envelope created is a SOAP 1.2 envelope. If the invoked service is a SOAP 1.1 service, the envelope created is a SOAP 1.1 envelope. If the `body` variable contains a piece of reference XML, it is sent as is—in other words, the referenced content is not substituted into the message.

If attachments are defined in the `attachments` variable, a MIME package is created from the main message and the attachment data. The handling of the content for each attachment part is similar to how content is handled for messaging services.

39.9.2 XML Services (Non SOAP)

The messages to XML-based services from Oracle Service Bus is constructed from the contents of the `body` variable:

- If the `body` variable is empty, then a zero-size message is sent.

- If the `body` variable contains multiple XML snippets, then only the first snippet is used in the outbound message. For example, if `<soap:Body>` contains `<abc/><xyz/>`, only `<abc/>` is sent.

- If the content of the `body` variable is text and not XML, an error is thrown.

- If the `body` variable contains a piece of reference XML, it is sent as is—in other words, the referenced content is not substituted into the message.

- If attachments are defined in the `attachments` variable, a MIME package is created from the XML message and the attachment data. In the case of a null XML message, the corresponding MIME body part is empty. The handling of the content for each attachment part is similar to how content is handled for messaging services.

Regardless of any data it contains, the `header` variable does not contribute any content to the outbound message.

For examples of how messages are constructed for service callout actions, see Chapter 21, "Proxy Services: Actions."

39.9.3 Messaging Services

The messages to messaging services from Oracle Service Bus are constructed from the contents of the `body` variable.

- If the `body` variable is empty, then a zero-size message is sent, regardless of the outgoing message type.

- If the outgoing message type is XML, then the message is constructed in the same way as it is for Section 39.9.2, "XML Services (Non SOAP)."

- If the outgoing message type is MFL, then the behavior is similar to that for XML message types except that the extracted XML is converted to MFL. (An error occurs if the XML > MFL conversion cannot be performed.)

- If the target service requires text messages, the contents of the `body` variable are interpreted as text and sent. In this way, it is possible for Oracle Service Bus to handle incoming XML messages that must be delivered to a target service as text. In other words, you do not need to configure the message flow to handle such messages.

- For target services that expect binary messages, the `body` variable must contain a piece of reference XML—the reference URI references the binary data stored in the Oracle Service Bus in-memory hash table. The referenced content is sent to the target service.

  For cases in which a client, a transport, or the designer of a proxy service specifies the reference URI, the referenced data is not stored in the Oracle Service Bus and thus cannot be de referenced to populate the outbound message. Consequently, the reference XML is sent in the message.

  If the `body` variable contains a piece of reference XML, and the target service requires a message type other than binary, the reference XML inside the `body` variable is treated as content. In other words, it is sent as XML, converted to text, or converted to MFL. This is true regardless of the URI in the reference XML.

Regardless of any data it contains, the header variable does not contribute any content to the outbound message.

For examples of how messages are constructed for service callout actions, see Chapter 21, "Proxy Services: Actions."

39.9.3.1 About Sending Binary Content in Email Messages

For binary messages, Oracle Service Bus does not insert the message content into the body variable. Instead, a `<binary-content/>` reference element is created and inserted into the `<SOAP:Body>` element (see Section 39.3, "Message-Related Variables"). However, the email standard does not support sending binary content type as the main part of a message. If you want to send binary messages through email to a messaging service that accepts text or XML documents and optional attachments, you can do so as follows:

1. Transfer the binary-content reference XML from $body to $attachments.

2. Replace the content of $body with text or XML wrapped in a `<SOAP:Body>` element.

For the case in which the outgoing message type is MFL, the contents of $body is converted from XML to text or binary based on the MFL transformation:

- If the target service expects to receive text message, you can set the content-type (the default is binary for MFL message type) as text/plain in $outbound

- If the target service expects to receive binary messages, it is not possible to send MFL content through the email transport.

To learn more about how binary content is handled, see Section 39.3.4, "Binary Content in the body and attachments Variables."

39.9.4 Related Topics

Section 39.10, "Message Context Schema"

Section 21.6, "Adding Service Callout Actions"

Section 21.7, "Adding Transport Header Actions"

Section 22.8, "Adding Route Nodes to Message Flows"

39.10 Message Context Schema

The message context schema (MessageContext.xsd) that specifies the types for the message context variables is shown in Section 39–1, "Message Context.xsd."

When working with the message context variables, you need to reference MessageContext.xsd which is available in a JAR file, *OSB_ORACLE_HOME*/lib/sb-schemas.jar, and the transport-specific schemas, which are available at *OSB_ORACLE_HOME*/lib/transports/.

Example 39–1   Message Context.xsd

```
<schema targetNamespace="http://www.bea.com/wli/sb/context"
      xmlns:mc="http://www.bea.com/wli/sb/context"
      xmlns="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified"
      attributeFormDefault="unqualified">
   <!--=============================================================== -->

   <!-- The context variable 'fault' is an instance of this element -->
```

```
    <element name="fault" type="mc:FaultType"/>

    <!-- The context variables 'inbound' and 'outbound' are instances of this element -->
    <element name="endpoint" type="mc:EndpointType"/>

    <!-- The three sub-elements within the 'inbound' and 'outbound' variables -->
    <element name="service" type="mc:ServiceType"/>
    <element name="transport" type="mc:TransportType"/>
    <element name="security" type="mc:SecurityType"/>

    <!-- The context variable 'attachments' is an instance of this element -->
    <element name="attachments" type="mc:AttachmentsType"/>

    <!-- Each attachment in the 'attachments' variable is represented by an instance of this
element -->
    <element name="attachment" type="mc:AttachmentType"/>

    <!-- Element used to represent binary payloads and pass-by reference content -->
    <element name="binary-content" type="mc:BinaryContentType"/>

    <!-- ================================================================ -->

    <!-- The schema type for  -->
    <complexType name="AttachmentsType">
        <sequence>
            <!-- the 'attachments' variable is just a series of attachment elements -->
            <element ref="mc:attachment" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="AttachmentType">
        <all>
            <!-- Set of MIME headers associated with attachment -->
            <element name="Content-ID" type="string" minOccurs="0"/>
            <element name="Content-Type" type="string" minOccurs="0"/>
            <element name="Content-Transfer-Encoding" type="string" minOccurs="0"/>
            <element name="Content-Description" type="string" minOccurs="0"/>
            <element name="Content-Location" type="string" minOccurs="0"/>
            <element name="Content-Disposition" type="string" minOccurs="0"/>

            <!-- Contains the attachment content itself, either in-lined or as <binary-content/>
-->
            <element name="body" type="anyType"/>
        </all>
    </complexType>

    <complexType name="BinaryContentType">
        <!-- URI reference to the binary or pass-by-reference payload -->
        <attribute name="ref" type="anyURI" use="required"/>
    </complexType>

    <!-- ================================================================ -->

    <complexType name="EndpointType">
        <all>
            <!-- Sub-elements holding service, transport, and security details for the endpoint -->
            <element ref="mc:service" minOccurs="0" />
            <element ref="mc:transport" minOccurs="0" />
            <element ref="mc:security" minOccurs="0" />
        </all>
```

```
        <!-- Fully-qualified name of the service represented by this endpoint -->
        <attribute name="name" type="string" use="required"/>
    </complexType>


    <!-- ================================================================ -->


    <complexType name="ServiceType">
        <all>
            <!-- name of service provider -->
            <element name="providerName" type="string" minOccurs="0"/>

            <!-- the service operation being invoked -->
            <element name="operation" type="string" minOccurs="0"/>
        </all>
    </complexType>


    <!-- ================================================================ -->


    <complexType name="TransportType">
        <all>
            <!-- URI of endpoint -->
            <element name="uri" type="anyURI" minOccurs="0" />

            <!-- Transport-specific metadata for request and response (includes transport headers)
-->
            <element name="request" type="anyType" minOccurs="0"/>
            <element name="response" type="anyType" minOccurs="0" />

            <!-- Indicates one-way (request only) or bi-directional (request/response)
communication -->
            <element name="mode" type="mc:ModeType" minOccurs="0" />

            <!-- Specifies the quality of service -->
            <element name="qualityOfService" type="mc:QoSType" minOccurs="0" />

            <!-- Retry values (outbound only) -->
            <element name="retryInterval" type="integer" minOccurs="0" />
            <element name="retryCount" type="integer" minOccurs="0" />
        </all>
    </complexType>

    <simpleType name="ModeType">
        <restriction base="string">
            <enumeration value="request"/>
            <enumeration value="request-response"/>
        </restriction>
    </simpleType>

    <simpleType name="QoSType">
        <restriction base="string">
            <enumeration value="best-effort"/>
            <enumeration value="exactly-once"/>
        </restriction>
    </simpleType>


    <!-- ================================================================ -->


    <complexType name="SecurityType">
        <all>
```

```
            <!-- Transport-level client information (inbound only) -->
            <element name="transportClient" type="mc:SubjectType" minOccurs="0"/>

            <!-- Message-level client information (inbound only) -->
            <element name="messageLevelClient" type="mc:SubjectType" minOccurs="0"/>

            <!-- Boolean flag used to disable outbound WSS processing (outbound only) -->
            <element name="doOutboundWss" type="boolean" minOccurs="0"/>
        </all>
    </complexType>

<complexType name="SubjectType">
  <sequence>
    <!-- User name associated with this tranport- or message-level subject -->
    <element name="username" type="string"/>
    <element name="principals" minOccurs="0">
     <complexType>
      <sequence>
      <!-- There is an element for each group this subject belongs to, as
        determined by the authentication providers -->
      <element name="group" type="string"
                  minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
     </complexType>
<!-- ================================================================= -->

    <complexType name="FaultType">
        <all>
            <!-- A short string identifying the error (e.g. BEA38229) -->
            <element name="errorCode" type="string"/>

            <!-- Descriptive text explaining the reason for the error -->
            <element name="reason" type="string" minOccurs="0" />

            <!-- Any additional details about the error  -->
            <element name="details" type="anyType" minOccurs="0" />

            <!-- Information about where the error occured in the proxy -->
            <element name="location" type="mc:LocationType" minOccurs="0" />
        </all>
    </complexType>

    <complexType name="LocationType">
        <all>
            <!-- Name of the Pipeline/Branch/Route node where error occured -->
            <element name="node" type="string" minOccurs="0" />

            <!-- Name of the Pipeline where error occured (if applicable) -->
            <element name="pipeline" type="string" minOccurs="0" />

            <!-- Name of the Stage where error occured (if applicable) -->
            <element name="stage" type="string" minOccurs="0" />

            <!-- Indicates if error occured from inside an error handler -->
            <element name="error-handler" type="boolean" minOccurs="0" />
        </all>
    </complexType>
    <!-- Encapsulates any stack-traces that may be added to a fault <details> -->
    <element name="stack-trace" type="string"/>
</schema>
```

39.10.1 Related Topics

40

Using the Test Console

This chapter provides guidelines and information on testing services using the Oracle Service Bus Test Console, including a section that describes how to undeploy the Test Console in production environments.

The Oracle Service Bus Test Console is a browser-based test environment you use validate and test the design of your system. It is an extension of the Oracle Service Bus Administration Console. (The Test Console is also available in Eclipse.) You configure the object of your test (proxy service, business service, XQuery, XSLT, or MFL resource), execute the test, and view the results in the Test Console. In some cases, you can trace through the code and examine the state of the message at specific trace points. Design time testing helps isolate design problems before you deploy a configuration to a production environment.

The Test Console can test specific parts of your system in isolation and it can test your system as a unit. You can use the Test Console in clustered environments. However, Oracle does not recommend deploying the Test Console in production environments.

You can access the Test Console from:

- The Project Explorer
- The Resource Browser
- The XQuery Editor
- Eclipse, using the Run As and Debug As options

For detailed procedural information, see Chapter 33, "Test Console" and "Using the Oracle Service Bus Debugger" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

This chapter contains the following topics:

- Section 40.1, "Prerequisites"
- Section 40.2, "Testing Proxy Services"
- Section 40.3, "Testing Business Services"
- Section 40.4, "Recommended Approaches to Testing Proxy and Business Services"
- Section 40.5, "Tracing Proxy Services Using the Test Console"
- Section 40.6, "Testing Resources"
- Section 40.7, "Performing XQuery Testing"
- Section 40.8, "Testing Services With Web Service Security"
- Section 40.9, "Test Console Transport Settings"

- Section 40.10, "About Security and Transports"

- Section 40.11, "Undeploying the Test Console"

40.1 Prerequisites

To use the Test Console:

- You must have Oracle Service Bus running and you must have activated the session that contains the resource you want to test.

- You must disable the pop-up blockers in your browser for the XQuery testing to work. If you have toolbars in the Internet Explorer browser, this may mean disabling pop-up blockers from under the **Options** menu as well as for all toolbars that are configured to block them. XQuery testing is done only in the design time environment (in an active session).

- If you receive an error saying the Test Console service is not running, try setting the Admin server listen address to a specific valid address, such as localhost. In the Oracle WebLogic Server Console, go to **Environment** > **Servers** > *admin_server_name* > **Configuration** > **General** to set the Listen Address. Also, in a cluster, make sure all managed nodes are running.

- If you want the Test Console to generate and send SAML tokens to a proxy service, you must configure the proxy service to require SAML tokens and to be a relying party. For more information on creating a SAML relying party, see "Create a SAML 1.1 Relying Party" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

Note: When creating a SAML relying party:

- Only WSS/Sender-Vouches and WSS/Holder-of-Key SAML profiles are applicable to a proxy service.

- When you are configuring the relying party, for the target URL value provide the *URI* of the proxy service. You can view the URI of the proxy service by clicking on the proxy service name in the Oracle Service Bus Administration Console Project Explorer module. The URI displays in the Endpoint URI row of the Transport Configuration table.

40.2 Testing Proxy Services

You can test the following types of proxy services:

- WSDL Web service

- Messaging Service

- Any Soap Service

- Any XML Service

40.2.1 Direct Calls

> **Caution:** Testing proxy services with the direct call option enabled bypasses some important security steps, including access control. Oracle recommends that you not use the test framework in production systems. For information on untargeting the Test Console, see Section 40.11, "Undeploying the Test Console."

A direct call is used to test a proxy service that is collocated in the Oracle Service Bus domain. Using the direct call option, messages are sent directly to the proxy service, bypassing the transport layer. When you employ the direct call option, tracing is turned on by default, allowing you to diagnose and troubleshoot a message flow in the Test Console. By default, testing of proxy services is done using the direct call option.

When you use the direct call option to test a proxy service, the configuration data you input to the Test Console must be that which is expected by the proxy service from the client that invokes it. In other words, the Test Console plays the role of the client invoking the proxy service. Also, when you do direct call testing, you bypass the monitoring framework for the message.

Figure 40–1 illustrates a direct call. The message bypasses the transport layer; it is delivered directly to the proxy service (P1).

Figure 40–1   Direct Call to Test a Proxy Service
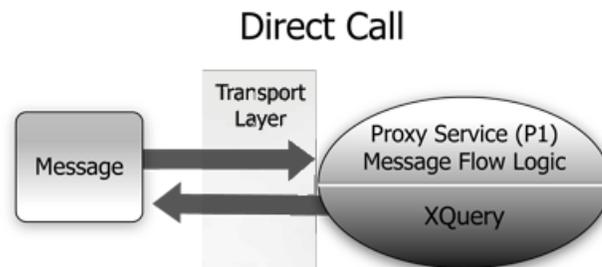
A direct call strategy is best suited for testing the internal message flow logic of proxy services. Your test data should simulate the expected message state at the time it is dispatched. Use this test approach in conjunction with setting custom (inbound) transport headers in the Test Console Transport panel to accurately simulate the service call.

40.2.2 Indirect Calls

When you test a proxy service with an *indirect* call (that is, when the direct call option is not selected), the message is sent to the proxy service through the transport layer. The transport layer performs manipulation of message headers or metadata as part of the test. The effect is to invoke a proxy service to proxy service runtime path.

Figure 40–2 illustrates an indirect call. The message is first processed through the transport layer and is subsequently delivered to the proxy service (P1).

Figure 40–2   Indirect Call to Test a Proxy Service
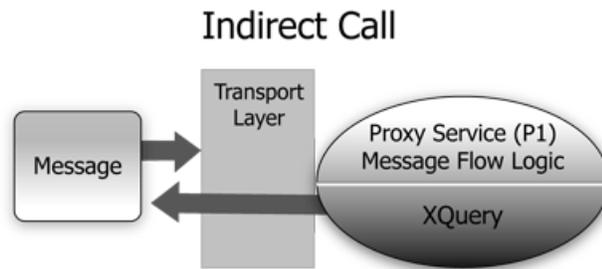
Oracle recommends this testing strategy when testing a proxy service to proxy service interface when both services run in the same JVM. Use this test approach in conjunction with setting custom (outbound) transport headers in the Test Console Transport panel to accurately simulate the service call. For more information on transport settings, see Section 40.9, "Test Console Transport Settings."

Using an indirect call, the configuration data you input to the test is the data being sent from a proxy service, for example, from a route node or a service callout action of another proxy service. In the indirect call scenario, the Test Console plays the role of the proxy service that routes to, or makes a callout to, another service.

> **Note:**   Using an indirect call to a request/response MQ proxy service will not work.
>
> In addition, the Test Console does not display the response from an indirect call to an MQ or JMS request/response proxy service using a correlation based on a messageID. When you test an MQ or JMS request/response proxy service with an indirect call, the response is retained in the response queue, and not displayed in the Test Console.
>
> For more information, see "MQ Transport" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

40.2.3 HTTP Requests

When you test proxy services, the Test Console never sends a HTTP request over the network, therefore, transport-level access control is not applied. Transport-level access control is achieved through the Web application layer—therefore, even in the case that an indirect call is made through the Oracle Service Bus Administration Console transport layer, an HTTP request is not sent over the network and transport-level access control is not applied. For information about message processing in the transport layer, see "Architecture Overview" in the *Oracle Fusion Middleware Concepts and Architecture for Oracle Service Bus*.

For information about transport settings, see Section 33.4, "Understanding How the Runtime Uses the Transport Settings in the Test Console.".

40.3 Testing Business Services

You can test the following types of business services:

- WSDL Web service
- Transport Typed Service
- Messaging Service

- Any Soap Service

- Any XML Service

When testing business services, the messages are always routed through the transport layer. The direct call option is not available. The configuration data that you provide to the Test Console to test the service is that which represents the state of the message that is expected to be sent to that business service—for example, from a route node or a service callout action of a proxy service. The Test Console functions in the role of the caller proxy service when you use it to test a business service.

> **Tip:** Ensure that the user name and password that you specify in the Test Console exists in the local Oracle Service Bus domain even if the business service being tested is in a remote domain. The test service performs a local authentication before invoking any proxy or business service.

40.4 Recommended Approaches to Testing Proxy and Business Services

In the scenario depicted in Figure 40–3, a client invokes the proxy service (P1). The message flow invokes business service B1, then proxy service P2, then proxy service P3 before returning a message to the client. Interfaces are identified by number.

Figure 40–3   Test Scenario Example
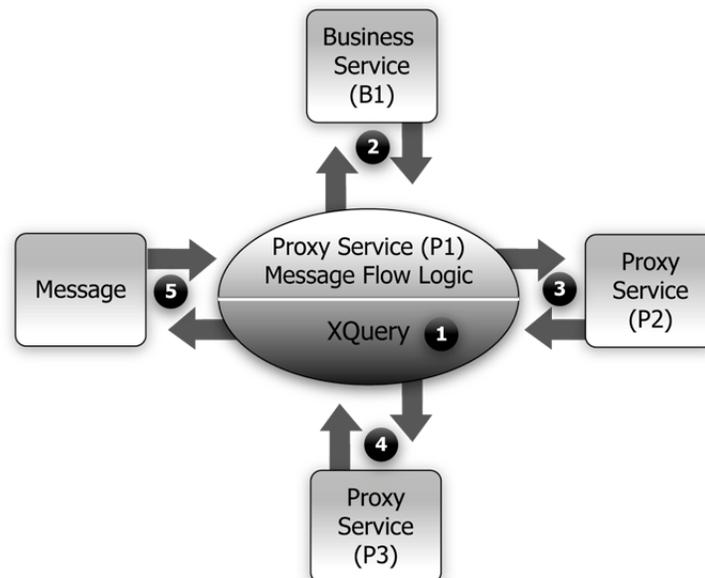
There are many valid test strategies for this scenario. Oracle recommends the following:

- Complete the testing of interfaces other than the client interface to a given proxy service before you test the client call. In the sample scenario illustrated in Figure 40–3, this means that you complete the testing of interfaces 1 through 4 first, then test interface 5. In this way, the message flow logic for the proxy service (P1) can be iteratively changed and tested (through interface 5) knowing that the other interfaces to the proxy service function correctly.

- Validate and test all the XQuery expressions in a message flow prior to a system test. In Figure 40–3, interface 1 refers to XQuery expression tests.

- Test proxy service to business service (interface 2 in Figure 40–3) using a *indirect call*. In other words, route the messages through the transport layer.

- Test proxy service to proxy service (interfaces 3 and 4 in Figure 40–3) using an *indirect call*. In other words, disable the direct call option, which means that during testing, the messages are routed through the transport layer.

- Make your final *system* test simulate the client invoking the proxy service P1. This test is represented by interface 5 in Figure 40–3. Test interface 5 with a direct call. In this way, during the testing, the messages bypass the transport layer. By default, tracing is enabled with a direct call.

- Save the message state after executing successful interface tests to facilitate future troubleshooting efforts on the system. Testing interface 5 is in fact a test of the complete system. Knowing that all other interfaces in the system work correctly helps narrow the troubleshooting effort when system errors arise.

40.5 Tracing Proxy Services Using the Test Console

Tracing the message through a proxy service involves examining the message context and outbound communications at various points in the message flow. The points at which the messages are examined are predefined by Oracle Service Bus. Oracle Service Bus defines tracing for stages, error handlers, and route nodes.

For each stage, the trace includes the changes that occur to the message context and all the services invoked during the stage execution. The following information is provided by the trace:

- (New variables)—The names of all new variables and their values. To view values, click the + sign.

- (Deleted variables)—The names of all deleted variables.

- (Changed variables)—The names of all variables for which the value changed. To view the new value, click the + sign.

- Publish—Every publish call is listed. For each publish call, the trace includes the name of the service invoked, and the value of the `outbound`, `header`, `body`, and `attachment` variables.

- Service callout—Every service callout is listed. For each service callout, the trace includes the name of the service that is invoked, the value of the `outbound` variable, the value of the `header`, `body`, and `attachment` variables for both the request and response messages.

The trace contains similar information for route nodes as for stages. In the case of route nodes, the trace contains the following categories of information:

- The trace for service invocations on the request path

- The trace for the routed service

- The trace for the service invocations on the response path

- Changes made to the message context between the entry point of the route node (on the request path) and the exit point (on the response path)

40.5.1 Example: Testing and Tracing a Proxy Service

The following example scenario uses one of the proxy services in the Oracle Service Bus Examples domain as a basis of instruction, the `loanGateway3` proxy service associated with the *Validating a Loan Application* example.

To test this proxy service in the Oracle Service Bus Examples domain using the Test Console, complete the following procedure:

1. Start the Oracle Service Bus Examples domain and load the samples data.

2. Log in to the Oracle Service Bus Administration Console, then select **Resource Browser** and locate the `LoanGateway3` proxy service.

3. Click the *Launch Test Console* icon for the `LoanGateway3` proxy service. The `Proxy Service Testing - LoanGateway3` page appears. The **Direct Call** and the **Include Tracing** options are selected.

4. Edit the test XML provided to send the following test message, illustrated in Example 40–1.

Example 40–1 Test Message for LoanGateway3

```
<loanRequest xmlns:java="java:normal.client">
    <java:Name>Name_4</java:Name>
    <java:SSN>SSN_11</java:SSN>
    <java:Rate>4.9</java:Rate>
    <java:Amount>2500</java:Amount>
    <java:NumOfYear>20.5</java:NumOfYear>
    <java:Notes>Name_4</java:Notes>
</loanRequest>
```

5. Click **Execute**.

Scroll to the bottom of the results page to view the tracing results in the **Invocation Trace** panel, shown in Figure 40–4.

Figure 40–4   Invocation Trace for LoanGateway3 Proxy Service

The trace indicates the following:

- Initial Message Context—Shows the variables initialized by the proxy service when it is invoked. To see the value of any variable, click the + sign associated with the variable name.

- Changed Variables—`$header`, `$body`, and `$inbound` changed as a result of the processing of the message through the validate loan application stage. These changes are seen at the end of the message flow.

- The contents of the `fault` context variable (`$fault`) is shown as a result of the stage error handler handling the validation error. The non-integer value (**20.5**) you entered for the `<java:NumOfYear>` element in caused the validation error in this case.

You can test the proxy service using different input parameters or by changing the message flow in the Oracle Service Bus Administration Console. Then run the test again and view the results.

40.6 Testing Resources

You can test the following resources:

- Section 40.6.1, "MFL"

- Section 40.6.2, "XSLT"

- Section 40.6.3, "XQuery"

40.6.1 MFL

A Message Format Language (MFL) document is a specialized XML document used to describe the layout of binary data. MFL resources support the following transformations:

- XML to binary—There is one required input (XML) and one output (binary).

- binary to XML—There is one required input (binary) and one output (XML).

Each transformation accepts only one input and provides a single output.

40.6.1.1 Example

The following example illustrates testing an MFL transformation. The Test Console generates a sample XML document from the MFL file. Execute the test using the XML input. A successful test results in the transformation of the message content of the input XML document to binary format.

Example 40–2 shows an example MFL file.

Example 40–2   Contents of an MFL File

```
<?xml version='1.0' encoding='windows-1252'?>
<!DOCTYPE MessageFormat SYSTEM 'mfl.dtd'>
  <MessageFormat name='StockPrices' version='2.01'>
   <StructFormat name='PriceQuote' repeat='*'>
    <FieldFormat name='StockSymbol' type='String' delim=':' codepage='windows-1252'/>
    <FieldFormat name='StockPrice' type='String' delim='|'codepage='windows-1252'/>
   </StructFormat>
  </MessageFormat>
```

The XML document generated by the Test Console to test the MFL file in the Example 40–2 is shown in Example 40–3.

Example 40–3   Test Console XML Input

```
<StockPrices>
    <PriceQuote>
        <StockSymbol>StockSymbol_31</StockSymbol>
        <StockPrice>StockPrice_17</StockPrice>
    </PriceQuote>
</StockPrices>
```

In the Test Console, click **Execute** to run the test. Example 40–4 shows the resulting test data, the stock symbol and stock price in binary format.

Example 40–4   MFL Test Console Results

```
00000000:53 74 6F 63 6B 53 79 6D 62 6F 6C 5F 33 31 3A 53 StockSymbol_31:S
00000010:74 6F 63 6B 50 72 69 63 65 5F 31 37 7C .. .. .. StockPrice_17|...
```

40.6.2 XSLT

Extensible Stylesheet Language Transformation (XSLT) describes XML-to-XML mappings in Oracle Service Bus. You can use XSL transformations when you edit XQuery expressions in the message flow of proxy services.

To test an XSLT resource, you must supply an input XML document. The Test Console returns the output XML document. You can create parameters in your document to assist with a transformation. XSLT parameters accept either primitive values or XML document values. You cannot identify the types of parameters from the XSL transformation. In the Input and Parameters panel of the XSLT Resource Testing page in the Test Console, you must provide the values to bind to the XSLT parameters defined in your document.

For more information, see Section 33.2.2, "Testing XSLT Transformations."

40.6.3 XQuery

XQuery uses the structure of XML to express queries across different kinds of data, whether physically stored in XML or viewed as XML.

An XQuery transformation can take multiple inputs and returns one output. The inputs expected by an XQuery transformation are variable values to bind to each of the XQuery external variables defined. The value of an XQuery input variable can be a primitive value (String, integer, date), an XML document, or a sequence of the previous types. The output value can be a primitive value (String, integer, date), an XML document, or a sequence of the previous types.

XQuery is a typed language—every external variable is given a type. The types can be categorized into the following groups:

- Simple/primitive type—String, int, float, and so on.

- XML nodes

- Untyped

In the Test Console, a single-line edit box is displayed if the expected type is a simple type. A multiple-line edit box is displayed if the expected data is XML. A combination input is used when the variable is not typed. The Test Console provides the following field in which you can declare the variable type: `[] as XML`. Input in the Test Console is rendered based on the type to make it easier to understand the type of data you must enter.

Figure 40–5 shows an XQuery with three variables: int, XML, and undefined type.

Figure 40–5 Input to the XQuery Test

default/newXquery2

Last Modified By	weblogic	**Description**
Last Modified On	06/05/06 11:57 AM	- no description -
References	0	
Referenced By	1	

XQuery
```
<xs:element name="routing">
        <xs:annotation>
                <xs:documentation>This is a simple routing table</xs:documentation>
        </xs:annotation>
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="local" maxOccurs="unbounded">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="logical"/>
                                                <xs:element name="physical"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="remote" minOccurs="0" maxOccurs="unbounded">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="logical"/>
                                                <xs:element name="physical"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

In the Test Console, all three variables are listed in the Variables panel. By default, XML is selected for the untyped variable as it is the most typical case. You must configure these variables in the Variables panel. See Section 33.2.3, "Testing XQuery Transformations."

You can also test an XQuery expression from the XQuery Editor.

40.7 Performing XQuery Testing

You must disable the pop-up blockers in your browser for the XQuery testing to work. If you have toolbars in the Internet Explorer browser, you may need to disable pop-up blockers from under the browser **Options** menu as well as for all toolbars that are configured to block them.

When performing XQuery testing in the Test Console, use the **Back** button to execute a new test. However, if you want to execute a new test after making changes to the XQuery, you must close and re-open the Test Console for the changes to take effect. For detailed information, see Section 33.2.3, "Testing XQuery Transformations."

40.8 Testing Services With Web Service Security

The Test Console supports testing proxy services and business services protected with Web Service Security (WSS). A SOAP service is protected with WSS if it has WS-Policies with WS-Security assertions assigned to it. Specifically, a service operation is protected with WS-Security if its effective request or response WS-Policy includes WS-Security assertions. WS-Policies are assigned to a service by WS-PolicyAttachment. See "Attaching WS-Policy Statements to WSDL Documents" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. An operation might have both a request policy and a response policy.

When an operation has a request or response WS-Policy, the message exchange between the Test Console and the service is protected by the mechanisms of WS-Security. According to the operation's policy, the test service digitally signs or encrypts the message (more precisely, parts of the message) and includes any applicable security tokens. You specify the input to the digital signature and encryption operations is the clear-text SOAP envelope specified as described in Section 33.1.2, "Configuring Proxy Services Test Data" and Section 33.1.6, "Configuring Business Services Test Data."

Similarly, the service processes the response according to the operation's response policy. The response may be encrypted or digitally signed. The test service then processes this response and decrypts the message or verifies the digital signature.

The Test Console (Security panel) displays fields used for testing services with WS-Security: **Service Provider**, **Username**, and **Password**.

If you specify a service key provider in the Test Console, all client-side PKI key-pair credentials required by WS-Security are retrieved from the service key provider. You use the user name and password fields when an operation's request policy specifies an Identity assertion and user name Token is one of the supported token types.

The **Service Provider**, **Username**, and **Password** fields are displayed whenever the operation has a request or response policy. Whether the values are required depends on the actual request and response policies.

Table 40–1 and Table 40–2 describe security scenarios.

Table 40–1   Digital Signature and Encryption Scenarios

Scenario	Is Service Key Provider Required?
The request policy has a Confidentiality assertion.	**No**. The test service encrypts the request with the service's public key. When testing a proxy service, the test service automatically retrieves the public key from the encryption certificate assigned to the service key provider of the proxy service. When testing a business service, the encryption certificate is embedded in the WSDL of the business service. The test service automatically retrieves this WSDL from the WSDL repository and extracts the encryption certificate from the WSDL.
The response policy has a Confidentiality assertion.	**Yes**. In this scenario, the operation policy requires the client to send its certificate to the service. The service will use the public key from this certificate to encrypt the response to the client. A service key provider *must* be specified and *must* have an associated encryption credential. If both request and response encryption are supported, different credentials must be used.

Table 40–1   (Cont.) Digital Signature and Encryption Scenarios

Scenario	Is Service Key Provider Required?
The request policy has an Integrity assertion.	**Yes**. The client must sign the request. A service key provider *must* be specified and *must* have an associated digital signature credential. Furthermore, if this is a SAML holder-of-key integrity assertion, a user name and password is needed in addition to the service key provider.
The response policy has an Integrity assertion.	**No**. In this case, the policy specifies that the service must sign the response. The service signs the response with its private key. The Test Console simply verifies this signature. When testing a proxy service, this is the private key associated to the service key provider's digital signature credential for the proxy service. When testing a business service, the service signing key-pair is configured in a product-specific way on the system hosting the service. In the case that the current security realm is configured to do a Certificate Lookup and Validation, then the certificate that maps to the service key provider must be registered and valid in the certificate lookup and validation framework. For more information on Certificate Lookup and Validation, see "Configuring the Certificate Lookup and Validation Framework" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

Table 40–2   Identity Policy Scenarios (Assuming that the Policy has an Identity Assertion)

Supported Token Types[1]	Description	Comments
UNT	The service only accepts WSS user name tokens	You must specify a user name and password in the Security panel.
X.509	The service only accepts WSS X.509 tokens	You must specify a service key provider in the Security panel and the service key provider must have an associated WSS X.509 credential.
SAML	The service only accepts WSS SAML tokens	You must specify a user name and password in the Security panel *or* a user name and password in the Transport panel. If both are specified, the one from the Security panel is used as the identity in the SAML token.
UNT, X.509	The service accepts UNT or X.509 tokens	You must specify a user name and password in the Security panel *or* a service key provider in the Security panel with an associated WSS X.509 credential. If both are specified, only a UNT token is generated.
UNT, SAML	The service accepts UNT or SAML tokens	You must specify a user name and password in the Security panel *or* a user name and password in the Transport panel. If both are specified, only a UNT token is sent.

Table 40–2   (Cont.) Identity Policy Scenarios (Assuming that the Policy has an Identity Assertion)

Supported Token Types[1]	Description	Comments
X.509, SAML	The service accepts X.509 or SAML tokens	You must specify one of the following: ■ user name and password in the Security panel ■ user name and password in the Transport panel ■ service key provider with an associated WSS X.509 credential
UNT, X.509, SAML	The service accepts UNT, X.509 or SAML tokens	You must specify one of the following: ■ user name and password in the Security panel ■ user name and password in the Transport panel ■ service key provider with an associated WSS X.509 credential

[1] From the Identity Assertion inside the request policy.

40.8.1 Limitations for Services and Policies

The following limitations exist for testing proxy services with SAML policies and business services with SAML holder-of-key policies:

■ Testing proxy services with inbound SAML policies is not supported.

■ Testing business services with a SAML holder-of-key policy is a special case. The SAML holder-of-key scenario can be configured in two ways:

  – as an integrity policy (this is the recommended approach)

  – as an identity policy

In both cases, you must specify a user name and password—the SAML assertion will be on behalf of this user. If SAML holder-of-key is configured as an integrity policy, you must also specify a service key provider. The service key provider must have a digital signature credential assigned to it. This case is special because this is the only case where a user name and password must be specified even if there is not an identity policy.

> **Note:**   After executing a test in the Test Console, the envelope generated with WSS is not always a valid envelope—the results page in the Test Console includes white spaces for improved readability. That is, the secured SOAP message is displayed with extra white spaces. Because white spaces can affect the semantics of the document, this SOAP message cannot always be used as the literal data. For example, digital signatures are white-space sensitive and can become invalid.

40.9 Test Console Transport Settings

You use the Transport panel in the Test Console to specify the metadata and transport headers for messages in your test system.

Figure 40–6 shows the Transport panel for a WSDL-based proxy service.

Figure 40–6   Transport Panel in the Test Console

By setting the metadata and the transport headers in the message flow of a proxy service, you influence the actions of the outbound transport. You can test the metadata, the message, and the headers so that you can view the pipeline output. The fields that are displayed in the Transport panel when testing a proxy service represent those headers and metadata that are available in the pipeline. The Test Console cannot filter the fields it displays depending on the proxy service. The same set of transport parameters are displayed for every HTTP-based request.

The **Username** and **Password** fields are used to implement basic authentication for the user that is running the proxy service. The **Username** and **Password** fields are not specifically transport related.

Metadata fields are located below the **Username** and **Password** fields and above the transport header fields. The fields displayed are based on the transport type of the service. Certain fields are pre-populated depending on the operation selection algorithm you selected for the service when you defined it.

For example, in the Transport panel displayed in Figure 40–6, the SOAPAction header field is populated with "http://example.orgprocessLoanApp". This value comes from the service definition (the selection algorithm selected for this proxy service was

SOAPAction Header). For more information about the selection algorithms, see Chapter 37, "Modeling Message Flow in Oracle Service Bus."

Specify the values in the Transport panel fields according to whether the message will be processed through the transport layer (for example, if you are testing the service using a direct call), or not (an indirect call).

When testing a proxy service with a direct call, the test data must represent the message as if it had been processed through the transport layer. That is, the test data should represent the message in the state expected at the point it leaves the transport layer and enters the service. When testing a proxy or business service using an indirect call, the test data represents the data that is sent from a route node or a service callout. The test message is processed through the transport layer.

When testing services that use the MQ transport, message text might appear garbled depending on the character set being used. The test console writes messages in the UTF-8 character format. If the MQ connection's coded character set identifier (CCSID) is configured to a value other than 1208 (UTF-8), the text appears garbled in the console. To avoid this, override the CCSID by defining a **User Header** named **characterSet** on the Transport panel of the Test Console. Set the value of the header to **1208**.

For information about specific headers and metadata and how they are handled by the test framework, see Section 33.4, "Understanding How the Runtime Uses the Transport Settings in the Test Console.".

40.10 About Security and Transports

When using the Test Console to test HTTP business services with BASIC authentication, the Test Console authenticates the user name and password from the service account of the business service. Similarly, when testing JMS, email, or FTP business services that require authentication, the Test Console authenticates the service account associated with the business service.

When you test proxy services, the Test Console never sends a HTTP request over the network. Therefore, transport-level access control is not applied.

40.11 Undeploying the Test Console

Oracle recommends that you not use the test framework in production systems. For example, testing proxy services with the direct call option bypasses some important security steps, including access control.

When you create an Oracle Service Bus domain, the Configuration Wizard, by default, includes the "ALSB Test Framework" (Test Console) as a target on the Admin Server and any Managed Servers. The following section describe different options for undeploying the Test Console:

■ Section 40.11.1, "Untargeting the Test Console Before Domain Creation"

■ Section 40.11.2, "Untargeting the Test Console when the Server is Running"

■ Section 40.11.3, "Untargeting the Test Console when the Server is Not Running"

40.11.1 Untargeting the Test Console Before Domain Creation

To untarget the Test Console in the Oracle Fusion Middleware Configuration Wizard before a domain is created:

1. When creating an Oracle Service Bus domain with the Configuration Wizard, select optional configuration for Deployments and Services.

2. In the related wizard pages that follow, for each server, deselect the **ALSB Test Framework** application.

When the wizard creates the domain, the Test Console (*OSB_ORACLE_HOME*\lib\sbTestFwk.ear) is not deployed.

40.11.2 Untargeting the Test Console when the Server is Running

To undeploy the Test Console when an Oracle Service Bus domain is running, do the following:

1. Start the Oracle WebLogic Server Administration Console and log in.

2. In the left navigation area, under Domain Structure, click **Deployments**. The Summary of Deployments page is displayed.

3. In the Deployments table, click the "ALSB Test Framework." The Settings page for the ALSB Test Framework is displayed.

4. Click the **Targets** tab.

5. Select the **Component** check box to select all the test framework resources, and click **Change Targets**.

6. On the Target Deployments page, deselect the AdminServer and all Managed Servers.

7. Click **Yes**. A message is displayed indicating that the settings have been successfully updated.

40.11.3 Untargeting the Test Console when the Server is Not Running

If an Oracle Service Bus domain is not running, you can use the WebLogic Scripting Tool (WLST) to untarget the Test Console from the Oracle Service Bus domain. For more information about WLST, see *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

To untarget the Test Console ("ALSB Test Framework"), complete the following steps:

1. If you have not already set up your environment to use WLST, see "Main Steps for Using WLST" in "Using the WebLogic Scripting Tool" in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

2. Invoke WLST Offline.

   ```
   C:>java com.bea.plateng.domain.script.jython.WLST_offline
   ```

3. To read the domain that was created using the Configuration Wizard execute:

   ```
   wls:/offline>readDomain("C:/oracle/user_projects/domains/base_domain")
   ```

4. To untarget the ALSB Test Framework application, execute:

   ```
   wls:/offline/base_domain>unassign("AppDeployment", "ALSB Test Framework",
   "Target", "AdminServer", "ManagedServer_1", "ManagedServer_2")
   ```

   Include the names of all managed servers in the command.

5. To update the domain execute:

   ```
   wls:/offline/base_domain>updateDomain()
   ```

6. To close the domain execute:

```
wls:/offline/base_domain>closeDomain()
```

7. Exit from the WLST command prompt execute:

```
wls:/offline>exit()
```

41

UDDI

This chapter describes how to use Oracle Service Bus with Universal Description, Discovery, and Integration (UDDI) registries such as Oracle Service Registry.

This chapter contains the following sections:

41.1 UDDI, UDDI Registries, and Web Services

UDDI stands for Universal Description, Discovery, and Integration. The UDDI Project is an industry initiative which aims to enable businesses to quickly, easily, and dynamically find and carry out transactions with one another.

A populated UDDI registry contains cataloged information about businesses; the services that they offer; and communication standards and interfaces they use to conduct transactions. A UDDI registry provides a standards-based foundation infrastructure for locating services, invoking services, and managing metadata about services (security, transport, or quality of service). The UDDI registry can store and provide these metadata using arbitrary categorizations. These categorizations are called taxonomies.

UDDI registries are used in an enterprise to share Web services. Using UDDI registries helps companies organize and catalog Web services for sharing and reuse in an enterprise or with trusted external partners. The UDDI version 3.0 specification is available at:
http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3

UDDI registries are based on this specification, which provides details on how to publish and locate information about Web services using UDDI. The specification does not define runtime aspects of the services (it is only a directory of the services). UDDI provides a framework in which to classify your business, its services, and the technical details about the services you want to expose.

Publishing a service to a registry requires knowledge of the service type and the data structure representing that service in the registry. A registry entry has certain properties associated with it and these property types are defined when the registry is created. You can publish your service to a registry and make it available for other organizations to discover and use. Proxy services developed in Oracle Service Bus can be published to a UDDI registry. Oracle Service Bus can interact with any version 3.0-compliant UDDI registry. Oracle provides the Oracle Service Registry.

Figure 41–1 illustrates the integration of Oracle Service Bus with a UDDI registry.

Figure 41–1   Oracle Service Bus integration with UDDI
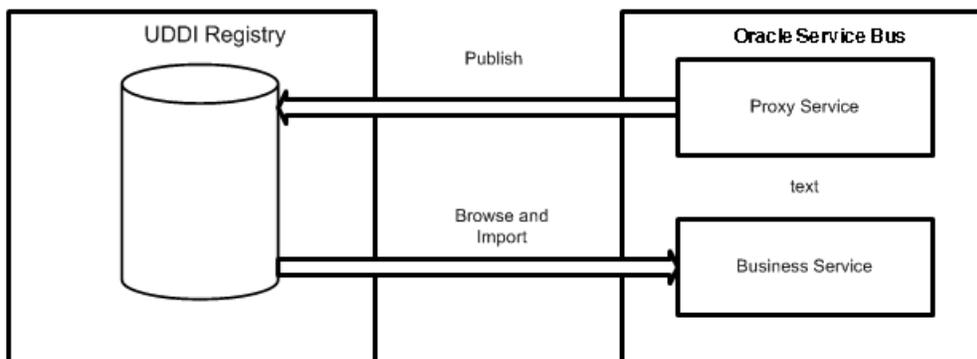
The Oracle Service Bus Web-based interface to Oracle Service Registry makes the registry accessible and easy to use. In working with UDDI, Oracle Service Bus promotes the reuse of standards-based Web services. In this way, Oracle Service Bus registry entries can be searched for, discovered, and used by multiple domains. Web services and UDDI are built on a set of standards, so reuse promotes the use of acceptable, tested Web services and application development standards across the enterprise. The Web services and interfaces can be catalogued by type, function, or classification so that they can be discovered and managed more easily.

41.1.1 Basic Concepts of the UDDI Specification

UDDI is based upon several established industry standards, including HTTP, XML, XML Schema Definition (XSD), SOAP, and WSDL. The UDDI specification describes a registry of Web services and its programmatic interfaces. UDDI itself is a set of Web services. The UDDI specification defines services that support the description and discovery of:

- Businesses, organizations, and other Web services providers

- The Web services they make available

- The technical interfaces that can be used to access and manage those services

41.1.2 Benefits of Using a UDDI Registry with Oracle Service Bus

A UDDI registry stores data and metadata about business services. A UDDI registry offers a standards-based mechanism to classify, catalog, and manage Web services so that they can be discovered and consumed by other applications. UDDI offers several benefits to IT managers and developers at both design time and runtime, including the following:

- UDDI improves infrastructure management by publishing information about services to the registry and categorizing the services for discovery. This ability of UDDI to categorize a growing portfolio of services makes it easier to manage them and helps you to understand relationships among components, supports versioning, and manages dependencies.

- You can import UDDI services from a registry to configure the parameters required to invoke the Web service and the necessary transport and security protocols.

- UDDI promotes the use of standards-based Web services and business services development in business applications and provides a link to a library of resources for Web services developers. This decreases development time and improves productivity. It also increases the prospect of interoperability between business applications by sharing standards-based resources.

- UDDI provides a user-friendly interface for searching and discovering Web services.

41.1.3 Introduction to UDDI Entities

UDDI uses a specific data model to represent entities that define organizations and services. Figure 41–2 shows the relationships between different UDDI entities.

Figure 41–2   UDDI Entities Representing Organizations and Services
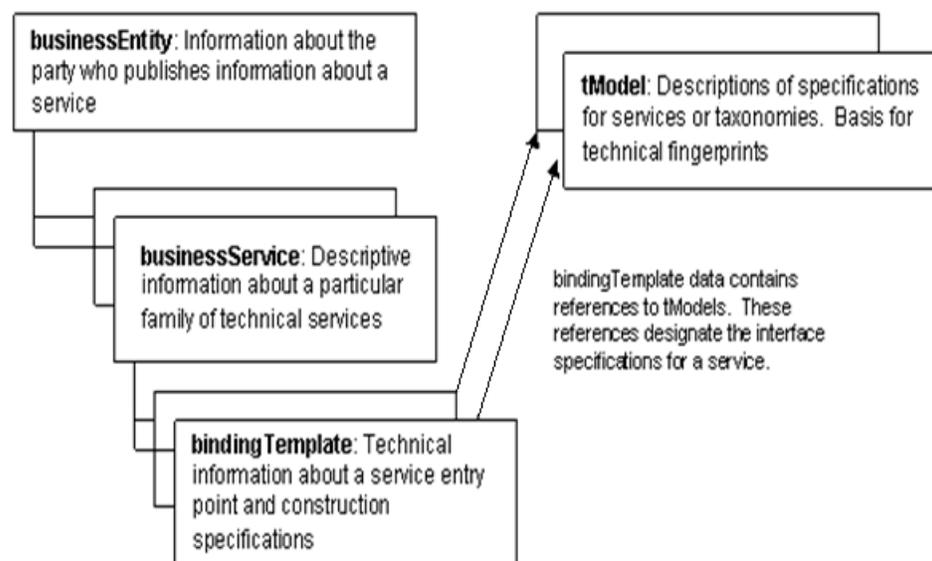
Table 41–1 provides a high-level overview of UDDI entities.

Table 41–1   High-Level Description of UDDI Entities

UDDI Entity	Description
Business Entity	An organization or group of people who own and provide the services. A business entity can be described by a set of names, descriptions, contact details for the service provider, a set of categories that represent the business entity features, unique identifiers, and discovery URLs.
Business Service	Represents functionality or resources provided by a business entity. A business service is described by a name, a description, and a set of categories that represent the function of the service. A business service in a UDDI registry does not necessarily represent a Web service. The UDDI registry can register arbitrary services, for example EJB, CORBA, and such.
Binding Template	Represents the technical details of how to invoke a business service. A business service can contain one or more binding templates. Binding templates are described by access points representing service endpoints (the endpoint URI and protocol specification), tModel instance information, and categories to reference specific features of the binding template.
tModel	Represents a technical specification; typically a specifications pointer, or metadata about a specification document, describing how services must be represented in the UDDI registry. The description of a service includes a name, a description, an overview document (a reference to a document specifying the purpose of the `tModel`), a category, and an identifier (to uniquely identify the `tModel`).

For more information on the UDDI data model and entities used in UDDI, see "Publishing and Finding Web Services Using UDDI" in *Oracle Fusion Middleware Programming Advanced Features of JAX-RPC Web Services for Oracle WebLogic Server*.

41.2 Sample Business Scenarios for Oracle Service Bus and UDDI

The following are two sample business scenarios that highlight the benefit of using UDDI.

41.2.1 Basic Proxy Service Communication with a UDDI Registry

This scenario shows how you can use Oracle Service Bus to import services from a registry and then publish Oracle Service Bus proxy services back to the registry. See Figure 41–3.

Figure 41–3   Proxy Service Communication with a UDDI Registry
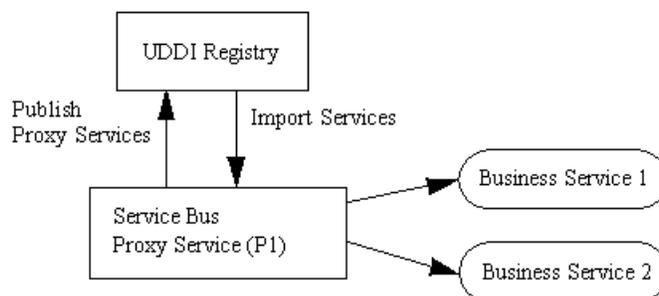
Oracle Service Bus imports business services from a UDDI registry. Proxy services are configured to communicate with the business services in the proxy service message flow. The proxy services can then be published back to the registry and made available for use by other domains.

41.2.2 Cross-Domain Deployment in Oracle Service Bus

This scenario shows cross-domain deployment using Oracle Service Bus. In this scenario, an Oracle Service Bus application in one domain requires access to an Oracle Service Bus service in another domain at runtime. See Figure 41–4.

Figure 41–4   Sample Business Case of Cross-Domain Deployment
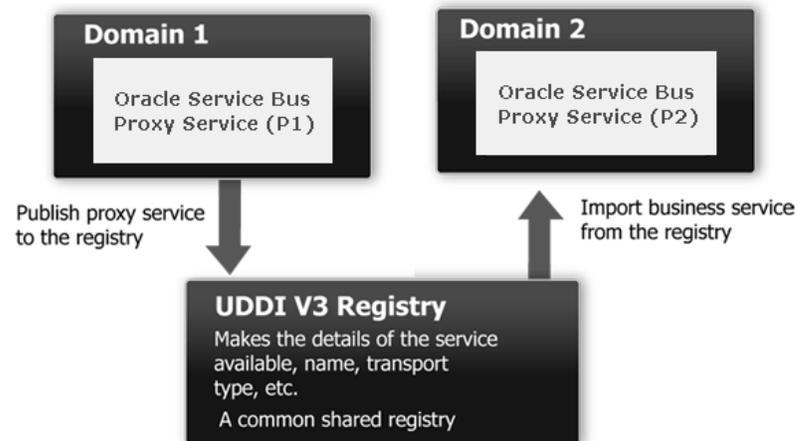
An instance of Oracle Service Bus is deployed in each of two domains. The Oracle Service Bus proxy service (P1) is configured in domain (D1). The Oracle Service Bus proxy service (P2) in domain (D2) requires access to proxy service (P1). As the domains cannot communicate directly with each other, P2 in D2 cannot use P1 in D1. The Oracle Service Bus import and export feature does not support runtime discovery of services in different domains, but publishing the service to a UDDI registry allows the discovery and use of a service in any domain. Once P1 is made available in the UDDI registry it can be invoked at runtime (for example, get a stock quote) and imported as a business service in another Oracle Service Bus proxy service.

When importing and exporting from different domains you should have network connectivity. A proxy service might reference schemas located in the repository of a different domain, in which case you need HTTP access to the domain to import it using the URL. In the absence of connectivity an error message is returned.

41.3 Using Oracle Service Bus and UDDI

Oracle Service Bus works with any UDDI registry that is compliant with the version 3.0 implementation of UDDI. Oracle Service Registry is a V3.0-compliant UDDI registry and is certified to work with Oracle Service Bus.

Using the Oracle Service Bus Administration Console or Eclipse, you can:

- Configure Oracle Service Bus to work with one or more V3.0-compliant UDDI registries.

- Configure a registry to allow users to publish services and import services.

- Publish information about any proxy service to a registry, including the following service types: WSDL, messaging, any SOAP, and any XML.

- Search for specific services in a registry or list all services available. You can search on business entity, service name pattern, or both.

- Import business services from a registry.

For detailed procedural information, see:

- Section 30.2, "Configuring UDDI Registries"

- Section 30.4, "Importing Business Services From a UDDI Registry"

- Section 30.7, "Publishing Proxy Services to a UDDI Registry"

41.3.1 A UDDI Workflow

The typical workflow for using a UDDI registry with Oracle Service Bus is as follows:

- Install Oracle Enterprise Repository

- Install Oracle Service Registry or any V3.0-compliant UDDI registry.

> **Note:** Oracle Enterprise Repository and Oracle Service Registry are not provided with Oracle Service Bus. In order to use those products, you have to buy a separate licence from Oracle.
>
> For more information on Oracle Enterprise Repository, see the Oracle Fusion Middleware documentation for Oracle Enterprise Repository.

- In the Oracle Service Bus IDE, generate business services from service assets in Oracle Enterprise Repository, as described in Section 19.1.2, "Generating a Business Service from Oracle Enterprise Repository."

- If you do not create a new UDDI Registry resource as part of generating a business service from Oracle Enterprise Repository, configure the service registry in the Oracle Service Bus Administration Console or in Eclipse. See:

  - Section 30.2, "Configuring UDDI Registries"

  - "Working with UDDI Registries" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*

- Set a default registry in the Oracle Service Bus Administration Console. See Section 30.3, "Setting Up a Default UDDI Configuration."

41.4 Configuring a Registry

You can configure a UDDI registry, make it available in Oracle Service Bus, and then publish Oracle Service Bus proxy services to it or import business services from the registry to be used in a proxy service. You must be in an active session in the Oracle Service Bus Administration Console to configure the registry. For detailed information, see:

- Section 30.2, "Configuring UDDI Registries"

- "Working with UDDI Registries" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

When publishing services to Oracle Service Registry, you need a valid user name and password for authentication to gain access to the registry. The user name and

password combination is implemented as a service account resource in Oracle Service Bus. You must define service accounts before configuring proxy services. See Section 17.1, "Specifying Service Accounts."

You can set up registries with multiple user names and passwords allowing different users to have different permissions based on the associated service accounts. In Oracle Service Registry, administrators manage user privileges and create views into the registry, specific to the needs of different users. In Oracle Service Bus, user permissions govern access to the registries, their content, and available functionality.

41.5 Publishing a Proxy Service to a UDDI Registry

You can use the Oracle Service Bus Administration Console or IDE to publish proxy services to the Oracle Service Registry. To do this you must have a user account set up in Oracle Service Registry. You can publish any proxy service to a UDDI registry. The permitted service types and transports are listed in Table 41–2.

Table 41–2   Service Types and Transports for Proxy Services

Service Type	Transports
WSDL	HTTP, JMS, Local, SB, WS
Transport Typed	JEJB
Any SOAP	HTTP, JMS, Local, SB
Any XML	Email, File, FTP, HTTP, JMS, Local, MQ, SB, SFTP, Tuxedo
Messaging	Email, File, FTP, HTTP, JMS, Local, MQ, SFTP, Tuxedo
	Note: Messaging services can have different content for requests and responses, or can have no response at all (one-way messages). Email, File, SFTP, and FTP should be one-way.

You can select the Business Entity under which a service is to be published. Business Entity Administration (including creation, removal, update, and deletion of entities) is done using the management console provided by the registry vendor (for example, the Business Service Console of Oracle Service Registry). The first time you publish to a registry you must load the `tModels` to that registry. You do this when you configure the publishing details in the Oracle Service Bus Administration Console or Oracle Service Bus plug-ins for Eclipse. For more information on how to publish to a UDDI registry, see Section 30.7, "Publishing Proxy Services to a UDDI Registry."

> **Note:**   An error can occur when you attempt to import a service from a UDDI registry if that service was originally published to the registry from an Oracle Service Bus cluster in which *any* of the clustered servers uses the localhost address. Specifically, when the service being imported references a resource (WSDL or XSD) which references other resources (WSDL or XSD).
>
> Ensure that before you publish services to a UDDI registry from a clustered domain, none of the servers in the cluster use localhost in the server addresses. Instead, use either the computer name or the IP address.

41.5.1 Publishing Local Proxy Services to UDDI

You can publish local proxy services to UDDI so you can associate them with Oracle Service Bus generic proxy services. For example, you might have an any SOAP or any XML generic proxy service that dynamically routes to multiple local transport proxy services with concrete WSDLs. Alternatively, you might have a generic proxy service in Enterprise Service Bus (ESB) 1 that dynamically routes to a generic proxy service in ESB 2 where the business service is attached. From the UDDI registry, you can get the WSDL of the local proxy service and the URL of the any SOAP or any XML generic proxy service. Combining the WSDL and URL creates an effective WSDL for sending messages to the local proxy service through the generic proxy service.

41.6 Using Auto-Publish

When you create a proxy service you can configure it to be published automatically to a default UDDI registry. You must first set up a default registry. See Section 30.3, "Setting Up a Default UDDI Configuration."

To enable the auto-publish feature for individual proxy services, you select the **Publish To Registry** check box on the Create a Proxy Service-General Configuration page. When you enable the **Publish To Registry** option, the proxy service is published to the default registry upon session activation. If the UDDI registry is unavailable, the publish action is retried. Any further changes to the proxy service resets the retry attempts. When a proxy service is republished to a UDDI registry, all taxonomies and categorizations, which are defined in UDDI for the proxy service, are preserved.

When you change the default registry, all the proxy services that have auto-publish enabled will be published to the new default registry. Synchronization then takes place with the current default registry. When a proxy service is not synchronized, the Oracle Service Bus Administration Console displays an unsynchronized icon.

> **Note:** When you have a default registry and you import a `sbconfig.jar`, which has a default registry set with the same logical name during the import, it is possible that the default registry will have an incorrect value for the business entity. You might now see errors on the Auto Publish Status page, if there are any auto-published proxy services. You can correct this situation by selecting the default registry again.

41.7 Importing a Service from a Registry

You can import services from a UDDI registry as Oracle Service Bus business services. When importing a WSDL-based service, if multiple UDDI binding templates are encountered, Oracle Service Bus creates a different business service for each binding template.

To establish access to UDDI registries in Oracle Service Bus you must have Oracle Service Bus IntegrationAdmin or IntegrationDeployer privileges. See "Role-Based Access in the Oracle Service Bus Administration Console" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*. The registry entries are located on the System Administration, Import from UDDI page in the Oracle Service Bus Administration Console. When importing, you select from the list of available registries. To discover a service in a registry you must query a specific registry. Entries in registries are unique. The query is performed when you specify what registry you want to use for importing a service.

As a governance best-practice alternative to direct import from a UDDI registry, you can also generate business services from service assets stored in Oracle Enterprise Repository, which are in turn associated with services in Oracle Service Registry.

You can import the following business services types from a UDDI registry into Oracle Service Bus:

- WSDL over HTTP binding. When multiple UDDI binding templates are present, a business service is created for each binding template.

- SOAP or XML binding over HTTP.

- Services that are categorized as Oracle Service Bus services. These are Oracle Service Bus proxy services that are published to a UDDI registry. This feature is primarily used in multi-domain Oracle Service Bus deployments where proxy services from one domain need to discover and route to proxy services in another domain.

For information on importing services from a UDDI registry, see:

- Section 30.4, "Importing Business Services From a UDDI Registry"

- Section 19.1.2, "Generating a Business Service from Oracle Enterprise Repository"

When a service is updated, you must re-import the service from the registry to get the most recent version, unless you have selected the **Enable Auto Import** option to auto-synchronize imported services with the UDDI registry. Any service that is imported with this option selected will be kept in synchrony with the UDDI registry. See Section 41.9, "Auto-Synchronization of Services With UDDI." If there is any failure during auto-synchronization, it will be reported on the Auto-Import Status page where you can update it manually.

Services have documents associated with them and these documents can include a number of other documents (schemas, policies, and so on). On import, the UDDI registry points to the document location based on the inquiry URL of the service. When a document that includes or references other resources is located, all of the referenced information and each included item is added as a separate resource in Oracle Service Bus.

Business Entity and pattern are the criteria used to search for a service in a registry. For example, you can enter `foo%`, when searching for a service. Services published by Oracle Service Bus have specific `tmodel` keys identifying the services that you use when searching for the service in the registry.

The Business Entity is the highest level of organization in the registry, though you can use other search criteria, such as business, application type, and so on. If you require authentication, then you need a user name and password which you must get from your system administrator.

41.7.1 Related References

See the following related content:

- Technical Notes can be found at
  http://www.oasis-open.org/committees/uddi-spec/doc/tns.htm. The note on *Using WSDL in a UDDI Registry* is important.

- UDDI product and development tool information is available at the OASIS UDDI Solutions page at http://uddi.org/solutions.html.

- The UDDI specifications
  http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

The specification defines the following:

- SOAP APIs that applications use to query and to publish information to a UDDI registry

- XML schema of the registry data model and the SOAP message formats

- WSDL definitions of the SOAP APIs

- UDDI registry definitions (`tModels`) of various identifier and category systems that may be used to identify and categorize UDDI registrations

41.8 Using Auto-Import

You can use the auto-import feature to synchronize the business services, which are imported from the Oracle Service Registry, with the corresponding services in the registry. See Section 30.5, "Using Auto-Import Status."

> **Note:** Auto-import is available only in the Oracle Service Bus Administration Console, not in the Oracle Service Bus plug-ins for Eclipse.

You can use the Auto Import Status page to do the following:

- Section 41.8.1, "Synchronize"
- Section 41.8.2, "Detach"

41.8.1 Synchronize

You can synchronize the services you have imported from the registry. If the services in the registry change, you can synchronize services in the Oracle Service Bus Administration Console with those in the registry. The following use case illustrates the process of synchronization. If the business service is not detached from the registry, Oracle Service Bus automatically subscribes to any changes to the service in the registry. If the service changes, an unsynchronized icon appears in the Resource Browser and Project Explorer indicating that the service must be synchronized. In addition, the Auto Import Status page shows this service and provides options for synchronizing the service or detaching it from the registry. Under certain circumstances, synchronizing the service might result in semantic validation errors that show up on the View Conflicts page. You will have to fix these errors before activating the session.

When a service is synchronized, the service is updated only with fields that are obtained from UDDI. Other fields in the service definition will preserve their values if modified since last import.

Consider a scenario where you publish services from Domain1 to a registry (see Figure 41–5). You then import these services from the registry into a domain, Domain2. Then you make changes to the services in Domain1 and update them in the registry. You can update the services in Domain2 by synchronizing them with the registry using the auto-import feature.

Figure 41–5   Sample Business Case of Cross-Domain Deployment
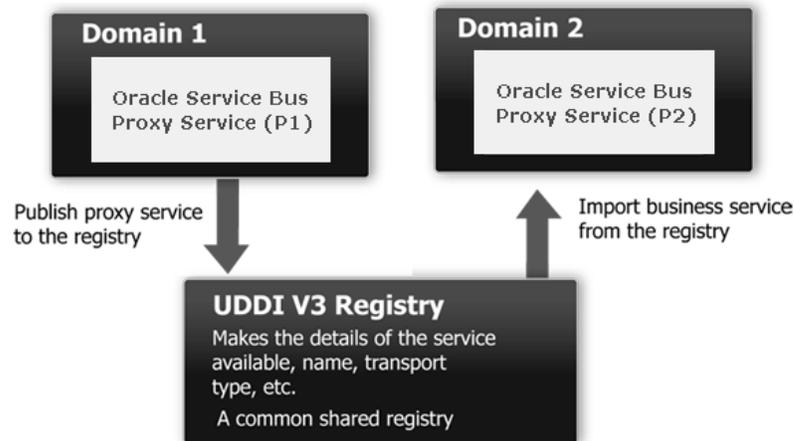
41.8.2 Detach

Sometimes you do not want the service in the Oracle Service Bus Administration Console to be synchronized with the corresponding service in the registry. You can avoid synchronization by detaching the service from the registry. See Section 30.6, "Detaching Services."

41.9 Auto-Synchronization of Services With UDDI

You can keep the service definitions in Oracle Service Bus automatically synchronized (both ways) with those in UDDI.

Services can be automatically published to a UDDI registry after they are created or changed within Oracle Service Bus and business service definitions can be imported from UDDI and automatically updated when the original service is changed in UDDI. Alternatively, you can configure the Oracle Service Bus Administration Console or the Oracle Service Bus plug-ins for Eclipse to prompt you for approval for synchronization when a service changes in the UDDI registry.

When configuring a registry, select the **Enable Auto Import** option to auto-synchronize imported services with the UDDI registry. Any service that is imported with this option enabled will be kept in synchrony with the UDDI registry automatically. If there is any failure during auto-synchronization, it is reported on the Auto-Import Status page where you can update it manually. See Section 30.2, "Configuring UDDI Registries."

41.10 Mapping Oracle Service Bus Proxy Services to UDDI Entities

Oracle Service Bus proxy service attributes must be mapped to the data model supported by the UDDI registry to allow a proxy service to be published as a UDDI business entity. Table 41–3 shows the service types, message types, and transports relevant to the UDDI registry mapping for an Oracle Service Bus proxy service.

Table 41–3   Proxy Service Attributes and Service Types

Service Type	Message Content Type	Transports
WSDL	SOAP or XML (with attachment)	HTTP, JMS, Local, SB, WS

Table 41–3   (Cont.) Proxy Service Attributes and Service Types

Service Type	Message Content Type	Transports
Transport Typed	SOAP or XML	JEJB
Any SOAP	Untyped SOAP (with attachment)	HTTP, JMS, Local, SB
Any XML	Untyped XML (with attachment)	Email, File, FTP, HTTP, JMS, Local, MQ, SB, SFTP, Tuxedo
Messaging	Binary, Text, MFL, XML (schema)	Email, File, FTP, HTTP, JMS, Local, MQ, SFTP, Tuxedo

> **Note:** Optional parts are listed in parentheses. Messaging services can have different content for requests and responses, or can have no response at all (one-way messages). Email, File, SFTP, and FTP should be one-way.

Proxy services have attributes in common and also attributes that are specifically defined by the transport protocols used by the service and the type of service. Each proxy service can deliver messages of a certain type.

The primary relevant entities in UDDI are:

- businessService: represents the service as a whole and contains high-level general information about the service.

- bindingTemplate: contains information for accessing the service.

- tModels: supplies the individual attributes for categorizing and defining the service.

Figure 41–6 shows how WSDL-based services are mapped to UDDI business entities.

Figure 41–6   WSDL Service to UDDI Mapping
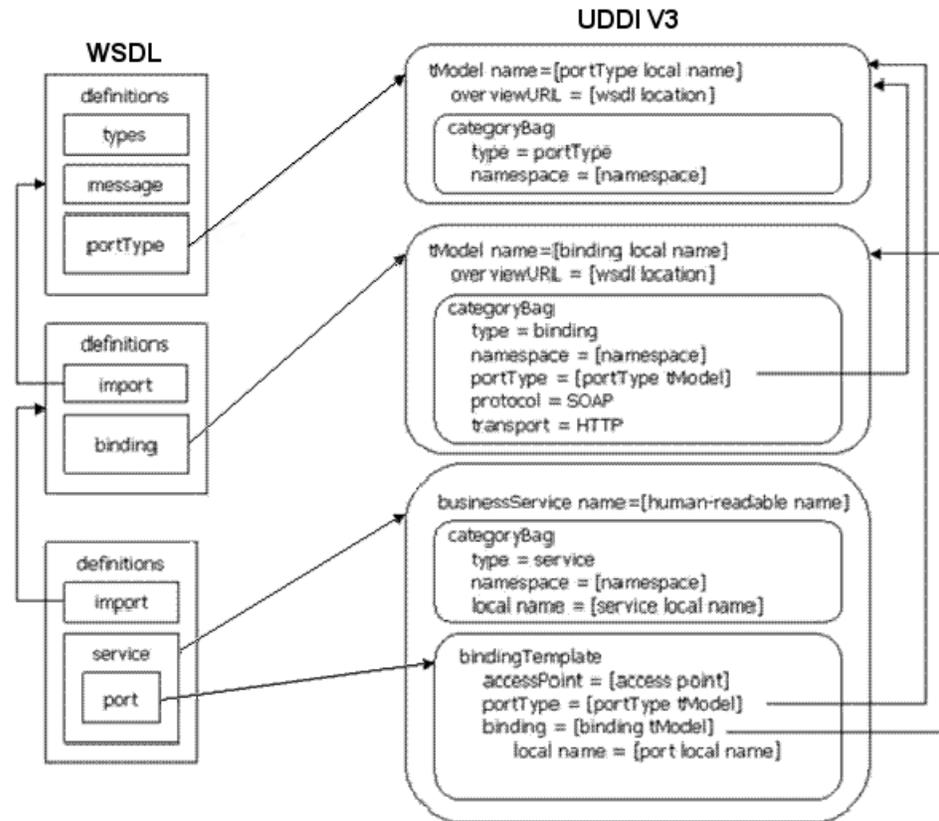
The technical note on *Using WSDL in a UDDI registry, version 2.0.2*, at http://www.oasis-open.org/committees/uddi-spec/doc/tns.htm, is used as the basis for publishing WSDL-based proxy services to the UDDI registry. This document is also used as a reference point for publishing non-WSDL based services. The document and the base UDDI specification describe the canonical technical models (tModels) used to describe UDDI entities. To publish Oracle Service Bus proxy services as entities in the UDDI registry, you must provide additional canonical tModels to support some of the constructs specific to Oracle Service Bus. Not all attributes of an Oracle Service Bus proxy service are useful when searching for a service, for example, service type and transport details. These attributes do not categorize the service. tmodels are configuration details of the service once it has been discovered. These configuration details are mapped to the business service binding template tmodelinstanceDetails section. Other attributes specifically identify a service and can be used as the search criteria for the service. These attributes are mapped using keyed references to tModels with values in the categoryBag of the binding template.

An example of how Oracle Service Bus maps to UDDI is shown in Figure 41–7.

Figure 41–7   Oracle Service Bus to UDDI Mapping
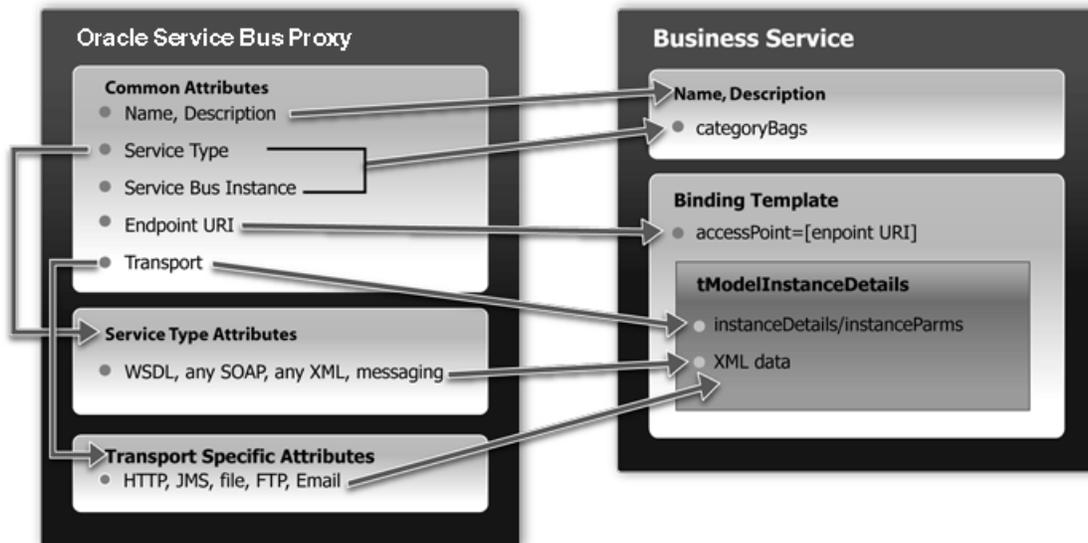
41.10.1 UDDI Mapping Details for an Oracle Service Bus Proxy Service

Oracle Service Bus high-level proxy service information maps to the business service as follows:

- Name and Description map to `businessService elements`.

- There is a special `keyedReferenceGroup` for Oracle Service Bus properties. An example of a key is `uddi:bea.com:attributes:oracleservicebus`.

- Oracle Service Bus type (WSDL, SOAP, XML, and Mixed) and instance are mapped to `keyedReferences` in the service category. An example of a key is `uddi:bea.com:servicetype`.

- An Oracle Service Bus instance maps to a `keyedReference` in the Oracle Service Bus `keyedReferenceGroup` (Name = "OracleServiceBus", Values = URL of the Oracle Service Bus instance).

  This instance serves two purposes:

  - To indicate that this service is in fact hosted by an Oracle Service Bus server.

  - To contain the URL of the Oracle Service Bus instance.

Example 41–1 shows a mapping of high-level proxy service information to a business service.

Example 41–1   Sample Proxy Service to Business Service Mapping

```
<keyedReferenceGroup tModelKey="uddi:bea.com:servicebus:properties">
  <keyedReference  tModelKey="uddi:bea.com:servicebus:servicetype"
    keyName="Service Type"
    keyValue="SOAP"/>
  <keyedReference  tModelKey="uddi:bea.com:servicebus:instance"
    keyName="Service Bus Instance"
    keyValue="http://FOO02.amer.bea.com:7001"/>
</keyedReferenceGroup>
```

> **Note:** The key for the businessService created when a proxy service is published is a publisher assigned key name. It is derived from the Oracle Service Bus domain name, the path of the proxy service, and the proxy service name. It takes the following form:
>
> ```
> uddi:bea.com:servicebus:<domainname>:<path>:<servicename>
> ```
>
> For example, AnonESBan, which is a domain in Oracle Service Bus, contains a project named Proxy, which contains a folder named Accounting, which in turn contains a proxy service called PayoutProxy. When PayoutProxy is published to UDDI, its businessService is created with the following key:
>
> ```
> uddi:bea.com:servicebus:AnonESB:Proxies:Accounting:PayoutProxy
> ```

Oracle Service Bus detailed proxy service information maps into the binding template as follows:

- The Endpoint URI maps to the access point.
- The Marker tModel for each transport maps to `tModelInstanceDetails`.
  - Transport `tModels` for HTTP, JMS, File, FTP, Email. New `tModels` are packaged with Oracle Service Bus to support JMS and File transports.
  - Detailed Oracle Service Bus configuration information maps to instanceParms.
- The Marker tModel for each service type maps to the tModelInstanceDetails. This includes the following:
  - Protocol `tModels` for WSDL, any SOAP, any XML, and Messaging. New `tModels` are packaged with Oracle Service Bus to support anySOAP, anyXML, and Messaging.
  - WSDL maps via WSDL to UDDI technology note.
  - Messaging has detailed configuration information that maps to `InstanceParms`.

Example 41–2 shows a detailed information mapping to the binding template.

Example 41–2 Sample Detailed Mapping to the Binding Template

```
<bindingTemplate bindingKey="uddi:" serviceKey="uddi:">
  <accessPoint useType="endPoint">file:///c:/temp/in3</accessPoint>
  <tModelInstanceDetails>
    <tModelInstanceInfo tModelKey="uddi:uddi.org:transport:file">
      <InstanceDetails>
      <InstanceParms><ALSBInstanceParms xmlns="http://www.bea.com/wli/sb/uddi">
        <property name="fileMask" value="*.*"/>
        <property name="sortByArrival" value="false"/> </ALSBInstanceParms>
      </InstanceParms>
      </InstanceDetails>
    </tModelInstanceInfo>
    <tModelInstanceInfo tModelKey="uddi:bea.com:servicebus:protocol:
        messagingservice">
      <InstanceDetails>
      <InstanceParms><ALSBInstanceParms xmlns="http://www.bea.com/wli/sb/uddi">
        <property name="requestType" value="XML"/>
        <property name="RequestSchema" value="http://domain.com:7001
          /sbresource?SCHEMA%2FDJS%2FOAGProcessPO"/>
        <property name="RequestSchemaElement"
```

```
            value="PROCESS_PO"/>
        <property name="responseType" value="None"/></ALSBInstanceParms>
      </InstanceParms>
      </InstanceDetails>
    </tModelInstanceInfo>
  </tModelInstanceDetails>
</bindingTemplate>
```

41.10.2 Transport Attributes

Each of the transport types in the `uddi:uddi.org:transport: *` group has a different set of detailed metadata. See Table 41–3. This metadata provides the configuration details of the transport for the proxy service. It is neither useful for characterizing the service nor useful in querying the service. However, after the service has been discovered, this data is needed to access the service. The metadata is represented by an XML string and is located in the `instanceParms` field in `tModelInstanceInfo`.

If you are mapping a proxy service that uses the HTTP transport, and as part of the HTTP configuration you need to describe some configuration details, including the required client authorization and the request and response character encoding. Example 41–3 provides an example of what must appear in the bindingTemplate `tModelInstanceDetails`.

Example 41–3   Example of tModelInstanceDetails

```
<tModelInstanceDetails>
  <tModelInstanceInfo tModelKey="uddi:uddi.org:transport:http">
    <instanceDetails>
      <instanceParms>
        <ALSBInstanceParms xmlns="http://www.bea.com/wli/sb/uddi">
          <property name="client-auth" value="basic"/>
          <property name="request-encoding" value="iso-8859-1"/>
          <property name="response-encoding" value="utf-8"/>
          <property name="Scheme" value="http"/>
        </ALSBInstanceParms>
      </instanceParms>
    </instanceDetails>
  </tModelInstanceInfo>
</tModelInstanceDetails>
```

> **Note:** For each transport, the service endpoint is always stored in the bindingTemplate `accessPoint` field.

The `client-auth` property is present in the `instanceParms` of the HTTP or HTTPS transport attributes whenever authentication is configured. The possible values for `client-auth` are `basic`, `client-cert`, and `custom-token`. Whenever the value is `custom-token`, two additional properties are present: `token-header` and `token-type`.

Because Oracle Service Bus business service definitions do not support custom token authentication in this release, if you import a service from UDDI that has a value of custom-token for `client-auth`, the service is imported as if it does not have any authentication configuration.

Table 41–4 is organized by transport type and lists the `tModelKey` and `instanceParms` used by each of the transports.

Table 41–4   Transport Attributes

Transport	tModelKey	InstanceParms
Email[1]	uddi:uddi.org:transport:smtp	■ Attachment Supported [Boolean] ■ Request Encoding
File	uddi:uddi.org:transport:file	■ File Mask ■ Sort by Arrival [Boolean] ■ Request Encoding
FTP	uddi:uddi.org:transport:ftp	■ File Mask ■ Sort by Arrival [Boolean] ■ Transfer Mode [Text, Binary] ■ Request Encoding
HTTP	uddi:uddi.org:transport:http	■ Client Authentication [None, Basic, Client Cert (HTTP only), and Custom Token] ■ Request Encoding ■ Response Encoding
JEJB	uddi:uddi.org:transport:jejb	■ URI ■ EJB Spec Version ■ Client JAR ■ Home Interface (not published for EJB 3.0) ■ Remote Interface (business interface for EJB 3.0) ■ Method Names
JMS	uddi:uddi.org:transport:jms	■ Destination Type [Queue, Topic] ■ Response Required, Response URI ■ Response Message Type [Bytes, Text] ■ Request Encoding ■ Response Encoding
Local	uddi:uddi.org:transport:local	None
MQ	uddi:bea.org:transport:mq	■ Response Required ■ Response URI ■ Response Correlation Pattern
SB	uddi:bea.org:transport:sb The URI scheme is sb when use ssl is false; sbs when use ssl is true.	None

Table 41–4   (Cont.) Transport Attributes

Transport	tModelKey	InstanceParms
SFTP	uddi:bea.org:transport:sftp	■ File Mask ■ Sort by Arrival [Boolean] ■ Request Encoding ■ Authentication Mode
Tuxedo	uddi:bea.org:transport:tuxedo	■ Response Required ■ Access Point ID ■ Buffer Type ■ Buffer Subtype ■ Classes Jar ■ Field Table Classes ■ View Classes
WS	uddi:uddi.org:transport:http WS uses the HTTP tModelKey	None

[1] The accessPoint in the Binding Template for an email transport uses the standard mailto URL format: `mailto:name@some_server.com`. This is different from the one configured for the proxy service in Oracle Service Bus, which is a URL oriented toward reading email. It is not be possible to derive this mailto URL from the proxy service definition as the server name is not known. For example, if the proxy service is defined to read from a POP3 server, it might be defined with a URL such as mailfrom:pop3.bea.com. When publishing such a proxy service, a dummy server is added. In the above example, the published URL will take the form mailto:some_name@some_server.com.

41.10.3 Service Type Attributes

Table 41–5 provides a high-level description of each of the service types.

Table 41–5   Service Type Attributes

Service	Description
WSDL	WSDL based proxies map to UDDI based on the *Using WSDL in a UDDI Registry, version 2.0.2* technical note at URL: http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v202-20040631.htm.
Any SOAP	A simple marker protocol in the tModel in the bindingTemplate tModelInstanceDetails, as well as in the categoryBag, defines the Any Soap attributes.
Any XML	A simple marker protocol tModel within the bindingTemplate tModelInstanceDetails, as well as in the categoryBag defines the Any XML attributes.

Table 41–5 (Cont.) Service Type Attributes

Service	Description
Messaging Services	A simple marker protocol tModel in the bindingTemplate `tModelInstanceDetails`, defines the messaging services attributes. Unlike the other service types, messaging services have additional configuration information associated with them, which provides detail about the request and response messages. The configuration details are represented as XML data in the `InstanceParms` data for the following tModel reference in the tModelInstanceInfo:

- Input message format (XML, Text, Binary, MFL)
- URL of input message schema in Oracle Service Bus (optional, if input message is XML)
- URL of input message MFL in Oracle Service Bus (if input message is MFL)
- Output message format (none, XML, Text, Binary, MFL)
- URL of output message schema in Oracle Service Bus (optional, if output message is XML)
- URL of output message MFL in Oracle Service Bus (if output message is MFL)

41.11 Canonical tModels Supporting Oracle Service Bus Services

The Oracle Service Bus-UDDI mapping provides a number of canonical tModels that are used to represent Oracle Service Bus metadata and relationships. These tModels must be registered in the UDDI registry to support this mapping. You can create these tModels in Oracle Service Registry under the administrator ID.

Table 41–6 through Table 41–9 provide a summary of the tModels.

Table 41–6 CategorizationGroup tModel Types

Name	Description
bea-com:servicebus:properties	Describes very specific attributes of an Oracle Service Bus service. In the data model it is used in the business service `categoryBag`.

Table 41–7 Categorization tModel Types

Name	Value	Description
bea-com:servicebus:serviceType	WSDL, SOAP, XML, Messaging Service	Describes the service type of the Oracle Service Bus service.
bea-com:servicebus:instance	URL of Oracle Service Bus Administration Console	Describes the service instance in Oracle Service Bus responsible for publishing the service to UDDI.

Table 41–8 Transport tModel Types

Name	Description
uddi-org:jms	Describes the type of transport used by the service. A reference to it is found in the accessPoint attribute of the business service binding template.

Table 41–8   (Cont.) Transport tModel Types

Name	Description
uddi-org:file	Describes the type of transport used to invoke the service. A reference to it is found in the accessPoint attribute of the business service binding template.

Table 41–9   Protocol tModel Types

Name	Description
bea-com:servicebus:anySoap	Describes the type of protocol used to access the service. It designates services that have a SOAP message but not defined by a WSDL or schema. The message body content is determined dynamically by the application.
bea-com:servicebus:anyXML	Describes the type of protocol used to access the service. It designates services having an XML message but not defined by a WSDL or schema. The message body content is determined dynamically by the application.
bea-com:servicebus:messagingService	Describes the type of protocol used to access the service. It designates services where the request message can be any XML (with or without schema), text, binary, or MFL and whose response message can be any of the above or none. The message body content is determined dynamically by the application.

41.12 Example

Example 41–4 is an example of the mapping for a Messaging Service, configured with JMS transport, the request being XML with a schema and the response being a text message.

Example 41–4   Sample Messaging Service Mapping

```
<businessService
  serviceKey="uddi:bea.com:servicebus:Domain:Project:JMSMessaging"
  businessKey="uddi:9cb77770-57fe-11da-9fac-6cc880409fac"
  xmlns="urn:uddi-org:api_v3">
    <name>JMSMessagingProxy</name>
    <bindingTemplates>
      <bindingTemplate
        bindingKey="uddi:4c401620-5ac0-11da-9faf-6cc880409fac"
        serviceKey="uddi:bea.com:servicebus:
          Domain:Project:JMSMessaging">
      <accessPoint useType="endPoint">
        jms://server.com:7001/weblogic.jms.XAConnectionFactory/
              ReqQueue
      </accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="uddi:uddi.org:transport:jms">
          <instanceDetails>
            <instanceParms>
              <ALSBInstanceParms
                xmlns="http://www.bea.com/wli/sb/uddi">
                <property name="is-queue" value="true"/>
                <property name="request-encoding"
                  value="iso-8859-1"/>
                <property name="response-encoding"
```

```
                      value="utf-8"/>
                  <property name="response-required"
                    value="true"/>
                  <property name="response-URI"
                    value="jms://server.com:7001/
                    .jms.XAConnectionFactory/
                      RespQueue"/>
                  <property name="response-message-type"
                    value="Text"/>
                  <property name="Scheme" value="jms"/>
                </ALSBInstanceParms>
              </instanceParms>
            </instanceDetails>
          </tModelInstanceInfo>
          <tModelInstanceInfo
            tModelKey="uddi:bea.com:servicebus:
                protocol:messagingservice">
            <instanceDetails>
              <instanceParms>
                <ALSBInstanceParms xmlns=
                  "http://www.bea.com/wli/sb/uddi">
                    <property name="requestType" value="XML"/>
                    <property name="RequestSchema"
                      value="http://server.com:7001/
                      sbresource?SCHEMA%2FDJS%2FOAGProcessPO"/>
                  <property name="RequestSchemaElement"
                    value="PROCESS_PO_007"/>
                  <property name="responseType" value="Text"/>
                  </ALSBInstanceParms>
              </instanceParms>
            </instanceDetails>
          </tModelInstanceInfo>
        </tModelInstanceDetails>
      </bindingTemplate>
    </bindingTemplates>
  <categoryBag>
  <keyedReferenceGroup tModelKey="uddi:bea.com:servicebus:properties">
    <keyedReference tModelKey="uddi:bea.com:servicebus:servicetype"
          keyName="Service Type"
          keyValue="Mixed" />
    <keyedReference tModelKey="uddi:bea.com:servicebus:instance"
          keyName="Service Bus Instance"
          keyValue="http://cyberfish.bea.com:7001" />
    </keyedReferenceGroup>
  </categoryBag>
</businessService>
```

42

Extensibility Using Java Callouts and POJOs

This chapter describes how to extend the capabilities of Oracle Service Bus by invoking custom Java code from within proxy services. Oracle Service Bus supports a Java exit mechanism through a Java callout action that lets you call out to a Plain Old Java Object (POJO).

Static methods can be accessed from any POJO. The POJO and its parameters are visible in the Oracle Service Bus Administration Console or Oracle Service Bus plug-ins for Eclipse at design time; the parameters can be mapped to message context variables.

You can also use Java callouts to create Java objects to store in the pipeline and to pass Java objects as parameters to other Java callouts.

For information about configuring a Java callout to a POJO, see Section 21.20, "Adding Java Callout Actions."

42.1 Usage Guidelines

The scenarios in which you can use Java callouts in Oracle Service Bus include the following:

- Custom validation—Examples of custom validation include validation against a DTD, or doing cross-field semantic validation in Java.

- Custom transformation—Examples of custom transformations can include converting a binary document to base64Binary, or vice versa, or using a custom Java transformation class.

- Custom authentication and authorization—Examples of custom authentication and authorizations include scenarios in which a custom token in a message must be authenticated and authorized. However, the authenticated user's identity cannot be propagated by Oracle Service Bus to the services or POJOs subsequently invoked by the proxy service.

- Lookups for message enrichment—For example, a file or Java table can be used to look up any piece of data that can enrich a message.

- Binary data access—You can use a Java callout to a POJO to sniff the first few bytes of a binary document to deduce the MFL type. The MFL type returned is used for a subsequent NonXML-to-XML transformation using the MFL Transform action.

- Implementing custom routing rules or rules engines.

- Create a Java object and store it in the pipeline.

- Pass a Java object as a parameter to another Java callout.

- Invoking a remote EJB operation or service with a POJO using a JEJB proxy service.

The input and return types for Java callouts are not restricted. See Section 39.3.5, "Java Content in the body Variable" for more information about storing and passing Java objects in the pipeline.

Enterprise JavaBeans (EJBs) also provide a Java exit mechanism. The use of EJBs is recommended over the use of Java Callouts in the following cases:

- When you already have an EJB implementation; although the JEJB transport lets you invoke EJBs with EJB calls through Oracle Service Bus, letting you leverage Oracle Service Bus functionality such as message routing, UDDI integration, alerts, per-operation monitoring, reporting, and result caching.

- When you require read access to a JDBC database—Although POJOs can be used for this purpose, EJBs were specifically designed for this and provide better support for management of, and connectivity to, JDBC resources.

- When you require write access to a JDBC database or other J2EE transactional resource—EJBs were specifically designed for transactional business logic and they provide better support for proper handling of failures. However, transaction and security context propagation is supported with POJOs, and the JEJB transport provides error handling in transaction contexts.

For outbound messaging, Oracle recommends that you write a custom transport instead of using POJOs or EJBs.

42.2 Working with Streaming Content

You can work with streaming content using Java callouts, both to pass binary-content as an input argument to callout methods and to accept streaming content results from Java callout methods.

42.2.1 Passing Streaming Content to a Java Callout

You can pass binary-content as an input argument to a Java callout method in a streaming fashion. Oracle Service Bus handles this by checking the Java type of the input argument. If the argument is of type `javax.activation.DataSource`, the system creates a wrapper `DataSource` object and gets the `InputStream` from the corresponding source by invoking the `Source.getInputStream()` method. You can call this method as many times as you need in your Java callout code.

In addition, the `getContentType()` method returns the `application/octet-stream` unless the binary content is a paged MIME attachment, in which case the value of the Content-Type header of the corresponding MIME part is used, if present.

Similarly, the `getName()` method returns the string value of the binary-content reference attribute unless the binary content is a paged MIME attachment, in which case the value of the Content-ID header of the corresponding MIME part is used, if available. The `getOutputStream()` method throws the `UnsupportedOperationException`, as required.

After completing, the result is passed to the Java callout method argument. To properly interpret the binary octets in the input stream, the Java callout method might also require the value of the Content-Transfer-Encoding header (for example, to determine whether the encoding is binary, 7bit, 8bit, and so on). You can pass this parameter as a separate argument, as shown in the following:

```
$attachments/*:attachment[1]/*:Content-Transfer-Encoding/text()
```

If the input argument is not a DataSource, Oracle Service Bus converts the argument to a byte[] array.

42.2.2 Streaming Content Results from a Java Callout

You can get streaming content results from a Java callout method. Oracle Service Bus handles this by checking the Java type of the result and then adding the new source to the source repository, setting the appropriate context variable value to the corresponding ctx:binary-content XML element.

> **Note:** To return the contents of a file from a Java callout method, you can use an instance of javax.activation.FileDataSource.

Whenever the Oracle Service Bus pipeline needs the binary contents of the source, it looks up the DataSource object corresponding to the ctx:binary-content element in the repository and invokes the DataSource.getInputStream() method to retrieve the binary octets.

The getInputStream() method might be called multiple times during message processing, for example to accommodate outbound message retries in the transport layer.

42.3 Best Practices

POJOs are registered as JAR resources in Oracle Service Bus. For information about JAR resources, see Chapter 8, "JARs."

In general, Oracle recommends that the JARs are small and simple—any large bodies of code that a JAR invokes or large frameworks that are made use of are best included in the system classpath. If you make a change to the system classpath, you must reboot the server.

Oracle recommends that you put dependent and overlapping classes in the same JAR resource; put them in different JARS if they are naturally distinct. Any change to a JAR causes all the services that reference it to be redeployed—this can be time consuming for your Oracle Service Bus system. The same class can be located in multiple JAR resources without causing conflicts. The JARs are dynamically class loaded when they are first referenced.

A single POJO can be invoked by one or more proxy services. All the threads in the proxy services invoke the same POJO. Therefore, the POJO must be thread safe. A class or method on a POJO can be synchronized, in which case it serializes access across all threads in all of the invoking proxy services. Any finer-grained concurrency (for example, to control access to a DB read results cache and implement stale cache entry handling) must be implemented by the POJO code.

It is generally a bad practice for POJOs to create threads.

43

XQuery Implementation

This chapter describes the custom Oracle XQuery functions available with Oracle Service Bus.

The Oracle XQuery engine fully supports all of the language features that are described in the World Wide Web (W3C) specification for XQuery with one exception: modules. For more information about the XQuery 1.0 and XPath 2.0 functions and operators (W3C Working Draft 23 July 2004), see the following URL:
http://www.w3.org/TR/2004/WD-xpath-functions-20040723/

Oracle Service Bus supports the following XQuery functions:

- A robust subset of the XQuery functions that are described in W3C specification.

- Oracle function extensions and language keywords provided as part of the Oracle XQuery engine—with a small number of exceptions, as described in Section 43.1, "Supported Function Extensions from Oracle."

- Oracle Service Bus-specific function extensions. See Section 43.2, "Function Extensions from Oracle Service Bus."

Note: All of the Oracle function extensions use the following function prefix `fn-bea:` In other words, the full XQuery notation for an extended function is of this format:

`fn-bea: function_name.`

43.1 Supported Function Extensions from Oracle

For descriptions of all Oracle function extensions, see "XQuery Implementation" in the *XQuery and XQSE Developer's Guide* at
http://download.oracle.com/docs/cd/E13162_
01/odsi/docs10gr3/xquery/extensions.html.

Oracle Service Bus supports all Oracle function extensions to XQuery except for the following:

- fn-bea:is-access-allowed

- fn-bea:is-user-in-group

- fn-bea:is-user-in-role

- fn-bea:userid

- fn-bea:async

- fn-bea:timeout

- fn-bea:get-property
- fn-bea:execute-sql()

Oracle recommends that you do not use the following functions in Oracle Service Bus. They are better covered by other language features:

- fn-bea:if-then-else
- fn-bea:QName-from-string
- fn-bea:sql-like

43.2 Function Extensions from Oracle Service Bus

Oracle Service Bus provides the following XQuery functions:

43.2.1 fn-bea:lookupBasicCredentials

The `fn-bea:lookupBasicCredentials` function returns the user name and unencrypted password from a specified service account. You can specify any type of service account (static, pass-through, or user-mapping). See Chapter 17, "Service Accounts."

Use the `fn-bea:lookupBasicCredentials` function as part of a larger set of XQuery functions that you use to encode a user name and password in a custom transport header or in an application-specific location within the SOAP envelope. You do not need to use this function if you only need user names and passwords to be located in HTTP Authentication headers or as WS-Security user name tokens. Oracle Service Bus already retrieves user names and passwords from service accounts and encodes them in HTTP Authentication headers or as WS-Security user name tokens when required.

The function has the following signature:

```
fn-bea:lookupBasicCredentials( $service-account as xs:string ) as
UsernamePasswordCredential
```

where `$service-account` is the path and name of a service account in the following form:

```
project-name[/folder[...]]/service-account-name
```

The return value is an XML element of this form:

```
<UsernamePasswordCredential
   xmlns="http://www.bea.com/wli/sb/services/security/config">
   <username>name</username>
   <password>unencrypted-password</password>
</UsernamePasswordCredential>
```

You can store the returned element in a user-defined variable and retrieve the user name and password values from this variable when you need them.

For example, your Oracle Service Bus project is named `myProject`. You create a static service account named `myServiceAccount` in a folder named `myFolder1/myFolder2`. In the service account, you save the user name of `pat` with a password of `patspassword`.

To get the user name and password from your service account, invoke the following function:

```
fn-bea:lookupBasicCredentials( myProject/myFolder1/myFolder2/myServiceAccount )
```

The function returns the following element:

```
<UsernamePasswordCredential
   xmlns="http://www.bea.com/wli/sb/services/security/config">
   <username>pat</username>
   <password>patspassword</password>
</UsernamePasswordCredential>
```

43.2.2 fn-bea:isUserInGroup

Returns an indicator of whether a given user belongs to a given group (true or false). For example:

```
fn-bea:isUserInGroup($user-name as xs:string, $group-name as xs:string)
```

43.2.3 fn-bea:isUserInRole

Returns an indicator of whether a given user belongs to a given role (true or false). For example:

```
fn-bea:isUserInRole($user-name as xs:string, $role-name as xs:string)
```

43.2.4 fn-bea: uuid

The function `fn-bea:uuid` returns a universally unique identifier. The function has the following signature:

```
fn-bea:uuid() as xs:string
```

You can use this function in the proxy pipeline to generate a unique identifier. You can insert the generated unique identifier into an XML document as an element. You cannot generate a unique identifier to the system variable. You can use this to modify a message payload.

For example, suppose you want to generate a unique identifier to add to a message for tracking purposes. You could use this function to generate a unique identifier. The function returns a string that you can add it to the SOAP header.

43.2.5 fn-bea:execute-sql()

The `fn-bea:execute-sql()` function provides low-level database access from XQuery within Oracle Service Bus message flows--see Section 37.9, "Accessing Databases Using XQuery." The query returns a sequence of flat row elements with typed data.

The function has the following signature:

```
fn-bea:execute-sql( $datasource as xs:string, $rowElemName as xs:QName, $sql as
xs:string, $param1, ..., $paramk) as element()*
```

where

- `$datasource` is the JNDI name of the datasource

- `$rowElemName` is the name of the row element—specify `$rowElemName` as whatever QName you want each element of the resulting element sequence to have

- `$sql` is the SQL statement

- `$param1, ..., $paramk` are 1 to k parameters

- `element()*` represents the sequence of elements returned

The return value is a sequence of flat row elements with typed data and automatically translates values between SQL/JDBC and XQuery data models. Data Type mappings that the XQuery engine generates or supports for the supported databases can be found in the Appendix H, "XQuery-SQL Mapping Reference."

When you execute the `fn-bea:execute-sql()` function from an Oracle Service Bus message flow, you can store the returned element in a user-defined variable.

Use the following examples to understand the use of the `fn-bea:execute sql()` function in Oracle Service Bus:

- Section 43.2.5.1, "Example 1: Retrieving the URI from a Database for Dynamic Routing"

- Section 43.2.5.2, "Example 2: Getting XMLType Data from a Database"

43.2.5.1 Example 1: Retrieving the URI from a Database for Dynamic Routing

Oracle Service Bus proxy services support specification of the URI to which messages are to be routed at runtime (dynamically)—see Section 37.8, "Using Dynamic Routing." Example 43–1 is an example use of the `fn-bea:execute-sql()` function to retrieve the URI from a database in a dynamic routing scenario.

Example 43–1 Get the URI for a Business Service from a Database

```
<ctx:route><ctx:service>
{
    fn-bea:execute-sql(
    'ds.myJDBCDataSource',
    xs:QName('customer'),
    'SELECT targetService FROM DISPATCH_MAPPING WHERE customer_priority=?',
      xs:string($body/m:Request/m:customer_pri/text()))
    )/TARGETSERVICE/text()
}
</ctx:service></ctx:route>
```

In Example 43–1:

- `ds.myJDBCDataSource` is the JNDI name to the data source

- `xs:string($body/m:Request/m:customer_pri/text())` interrogates the request message and populates `customer_priority=?` with the value of `customer_pri` in the message

- `/TARGETSERVICE/text()` is the path applied to the result of the SQL statement, which results in the string (CDATA) contents of that element being returned

- `<ctx:route><ctx:service> ... </ctx:service></ctx:route>` are required elements of the XQuery statement for a dynamic routing scenario

- The following is the table definition for `DISPATCH_MAPPING`:

```
create table DISPATCH_MAPPING
(
  customer_priority varchar2(256),
```

```
      targetService varchar2(256),
      soapPayload varchar2(1024)
);
```

The `DISPATCH_MAPPING` table is populated as shown in Example 43–2:

Example 43–2   DISPATCH_MAPPING Table

```
INSERT INTO DISPATCH_MAPPING (customer_priority, targetService, soapPayload)
VALUES ('0001', 'system/UCGetURI4DynamicRouting_proxy1', '<something/>');
INSERT INTO DISPATCH_MAPPING (customer_priority, targetService, soapPayload)
VALUES ('0002', 'system/UCGetURI4DynamicRouting_proxy2', '<something/>');
```

> **Note:**   The third column in the table (`soapPayload`) is not used in this
> scenario.

Executing the fn-bea:execute-sql for Example 3

If the XQuery in Example 43–1 is executed as a result of a proxy service receiving the
request message in Example 43–3 (the value of `<customer_pri>` in the request message
is `0001`), the URI returned for the dynamic route scenario is

```
system/UCGetURI4DynamicRouting_proxy1
```

(See also Example 43–2.)

Example 43–3   Example Request Message $body

```
<m:Request xmlns:m="http://www.bea.com/alsb/example">
<m:customer_pri>0001</m:customer_pri>
</m:Request>
```

43.2.5.2 Example 2: Getting XMLType Data from a Database

Data Type mappings that the XQuery engine generates or supports for the supported
databases can be found in the Appendix H, "XQuery-SQL Mapping Reference." Note
that the `XMLType` column type in SQL is not supported. However, you can access the
data in an `XMLType` column by using the `getStringVal()` method of the `XMLType` object
to convert it to a String value.

The following scenario outlines a procedure you can use to select data from an
`XMLType` column in an Oracle database.

1. Use an assign action in a proxy service message flow to assign the results of the
    following XQuery to a variable (`$result`).

Example 43–4   Get XMLType Data from a Database

```
fn-bea:execute-sql(
    'ds.myJDBCDataSource',
    'Rec',
    'SELECT a.purchase_order.getStringVal() purchase_order from datatypes a'
)
```

where:

- `ds.myJDBCDataSource` is the JNDI name to the data source

- `Rec` is the `$rowElemName`—therefore, Rec is the QName given to each element
   of the resulting element sequence

- `select a.purchase_order.getStringVal() ...` is the SQL statement that uses the `getStringVal()` method of the `XMLType` object to convert it to a String value

- `datatypes` is the table from which the value of the XML is read (the `datatypes` table in this case contains one row)

Note: The following is the table definition for the `dataty.pes` table:

```
create table datatypes
(
  purchase_order xmltype
);
```

---

2. Use a replace action to replace the node contents of `$body` with the results of the `fn-bea:execute-sql()` query (assigned to `$result` in the preceding step):

```
Replace [ node contents ] of [ undefined XPath ] in [ body ] with
[ $result/purchase_order/text() ]
```

The following listing shows `$body` after the replacement.

Note: The `datatypes` table contains one row (with the purchase order data); the row contains the XML represented in Example 43–5.

Example 43–5 $body After XML Content is Replaced with Result of fn-bea:execute-sql()

```
<soap-env:Body>
  <openuri:orders xmlns:openuri="http://openuri.com/">
    <openuri:order>
      <openuri:customerID>123</openuri:customerID>
      <openuri:orderID>123A</openuri:orderID>
    </openuri:order>
    <openuri:order>
      <openuri:customerID>345</openuri:customerID>
      <openuri:orderID>345B</openuri:orderID>
    </openuri:order>
    <openuri:order>
      <openuri:customerID>789</openuri:customerID>
      <openuri:orderID>789C</openuri:orderID>
    </openuri:order>
  </openuri:orders>
</soap-env:Body>
```

43.2.6 fn-bea:serialize()

You can use the `fn-bea:serialize()` function if you need to represent an XML document as a string instead of as an XML element. For example, you may want to exchange an XML document through an EJB interface and the EJB method takes String as argument. The function has the following signature:

```
fn-bea:serialize($input as item()) as xs:string
```

43.3 Creating and Using Custom XPath Functions

You can create and use your own custom XPath functions in both inline XQuery expressions and in XQuery resources. For more information, see "Creating and Using

Custom XPath Functions" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

Part III

Operations Guide

This part contains information and procedures for performing operational tasks in Oracle Service Bus. Chapters include:

44

Introduction to the Operations Guide

This part describes the operational functions you can perform in Oracle Service Bus, such as monitoring services, reporting, endpoint URI management, using smart search, tracing, and throttling.

This part is intended for the following audience:

- Operational Specialists: Responsible for monitoring services, servers and alerts in Oracle Service Bus.

- Architects: Responsible for designing the security architecture of the enterprise system.

To best understand this guide you must be familiar with Oracle Service Bus resources, proxy services, and business services, described in Part II, "User Guide".

For high-level architectural information about Oracle Service Bus, see *Oracle Fusion Middleware Concepts and Architecture for Oracle Service Bus*.

45

Roles in Oracle Service Bus

This chapter describes the default security roles in Oracle Service Bus and shows how they correspond to Oracle WebLogic Server default roles.

Oracle Service Bus supports various roles. The role assigned to a user determines the tasks that a user can perform. You can assign roles to users to secure resources and services in the Oracle Service Bus Administration Console by restricting access.

This chapter includes the following sections:

- Section 45.1, "Default Roles in Oracle Service Bus"

- Section 45.2, "Relation Between Roles in WLS and Roles in Oracle Service Bus"

- Section 45.3, "How to Create Roles in Oracle Service Bus"

You can also restrict the user interfaces that should be made available to a given role depending on the privileges of the role.

45.1 Default Roles in Oracle Service Bus

By default, IntegrationAdmin, IntegrationDeployer, IntegrationMonitor, IntegrationOperator are predefined roles in Oracle Service Bus. The following section describes the various roles available in Oracle Service Bus and their functionality.

45.1.1 IntegrationAdmin

The IntegrationAdmin role is an administrative security role. As an IntegrationAdmin, you can access Oracle Service Bus Administration Console. Users assigned to this role can access all resources and services in Oracle Service Bus. This role is granted to users requiring administrator privileges in Oracle Service Bus Administration Console.

In Oracle Service Bus, you can assign the IntegrationAdmin role by assigning the IntegrationAdmins parent group when you create or reconfigure a user. For more information about creating a user in Oracle Service Bus, see Section 25.3, "Adding Users.".

Users who are assigned this role can perform the following tasks in Oracle Service Bus Administration Console.

- Create or commit session

- Create, edit, or delete resources and projects

- View the available users and groups in Oracle Service Bus

- View and configure monitoring, reporting, and tracing for business and proxy services

- Import or export resources

- View and configure UDDI registries

- Publish and import resources from registries

45.1.2 IntegrationDeployer

The IntegrationDeployer role is assigned to users who deploy services. An IntegrationDeployer can access Oracle Service Bus Administration Console to create and deploy resources and services. Also in this role, you can access the existing resources and services in the Oracle Service Bus.

When a user is created or reconfigured in Oracle Service Bus, IntegrationDeployer role is granted by assigning the IntegrationDeployers parent group. For more information about how to create a user in the Oracle Service Bus, see Section 25.3, "Adding Users."

Users who are assigned this role can perform all tasks that can be performed by a user in the IntegrationAdmin role. For more information about tasks performed by an user in the IntegrationAdmin role, see Section 45.1.1, "IntegrationAdmin."

45.1.3 IntegrationMonitor

The IntegrationMonitor role is granted to users who monitor resources and services in Oracle Service Bus Administration Console. Users assigned to this role can also monitor violations to Service Level Agreements (SLAs), and the alerts from the message flow pipeline.

When a user is created or reconfigured in Oracle Service Bus Administration Console, the IntegrationMonitor role is assigned to users by assigning the IntegrationMonitors parent group. For more information about how to create a user in Oracle Service Bus Administration Console, see Section 25.3, "Adding Users."

Users who are assigned this role can perform the following tasks:

- View dashboard for SLA alerts and pipeline alerts

- Use SmartSearch to view business services, proxy services, alert destination and SLA alert rules

- View details of existing users and groups

- View details of resources

45.1.4 IntegrationOperator

The IntegrationOperator role is granted to users, who perform day-to-day operations in Oracle Service Bus Administration Console. IntegrationOperators can perform the day-to-day operations on the resources in Oracle Service Bus Administration Console. This role can also perform certain monitoring tasks and session management.

When a user is created or reconfigured in Oracle Service Bus Administration Console, the IntegrationOperator role is granted by assigning the IntegrationOperators parent group. For more information about how to create a user in Oracle Service Bus Administration Console, see Section 25.3, "Adding Users."

Users who are assigned this role can perform the following tasks:

- View configuration details of all resources

- View and configure monitoring, tracing, logging, and reporting for business services and proxy services

- Edit and update dashboard settings

- Add, update, and delete alert rules

- Add, view, delete, and edit alert destinations

- View and purge SLA alerts for business services and proxy services

- View and purge pipeline alerts for proxy services

- Use SmartSearch to view and edit operational settings for business services, proxy services, alert destination, and SLA alert rules

- Use global settings to enable or disable monitoring, pipeline alerting, SLA alerting, reporting, and logging at a global level

- View the status of all the servers associated with the domain

- View and purge message reports

- View the UDDI registries that have been configured for the domain

- View, the auto-publish status and auto-import status of business services and proxy services

- View security configurations of users and groups

For more information about tasks you can perform in each of these roles, see "Configuring Administrative Security" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

45.2 Relation Between Roles in WLS and Roles in Oracle Service Bus

Roles in Oracle Service Bus Administration Console are related to corresponding roles in the Oracle WebLogic Server Administration Console. Table 45–1 gives different roles available in Oracle Service Bus Administration Console and the corresponding roles in the Oracle WebLogic Server Administration Console.

Table 45–1   Relationship Between Roles in WLS and Roles in Oracle Service Bus

Roles in Oracle Service Bus	Roles in WLS
IntegrationAdmin	Administrator
IntegrationDeployer	Deployer
IntegrationMonitor	Monitor
IntegrationOperator	Operator

Users belonging to the Administrator role in Oracle WebLogic Server are automatically included in the IntegrationAdmin group in Oracle Service Bus Administration Console. The converse however, is not true.

> **Note:**   A user can also be associated with multiple roles. For example, a user can be associated with IntegrationAdmin role in Oracle Service Bus Administration Console must posses the Administrator role in Oracle Service Bus to access the Oracle WebLogic Server Administration Console.

45.3 How to Create Roles in Oracle Service Bus

You can create new roles in Oracle Service Bus if you possess Administrator role in the Oracle WebLogic Server Administration Console. An administrator can create new roles from the Global Roles page and customize the role by editing the conditions for the new role in the Global Role Conditions page. For more information about creating and customizing roles, see Section 25.11, "Adding Roles."

46

Monitoring Oracle Service Bus at Runtime

The chapter describes the Oracle Service Bus (OSB) monitoring framework, including statistics, alerts, operational settings, and the OSB Administration Console dashboard.

Oracle Service Bus enables you to monitor and collect runtime information required for system operations. Oracle Service Bus aggregates runtime statistics, which you can view on the dashboard. The dashboard lets you monitor the health of the system and notifies you when alerts are generated in your services. With this information, you can quickly and easily isolate and diagnose problems as they occur.

This section provides information on the following topics:

- Section 46.1, "What is Service Monitoring?"
- Section 46.2, "Aggregation Intervals"
- Section 46.3, "What are the Consequences of Renaming or Moving a Service?"
- Section 46.4, "What Statistics Are Available for Oracle Service Bus Services?"
- Section 46.5, "Accessing Statistical Information for Services"
- Section 46.6, "The Role of Alerts in Service Monitoring"
- Section 46.7, "What are Operational Settings for a Service?"
- Section 46.8, "SLA Alerting Functionality in Oracle Service Bus"
- Section 46.9, "The Oracle Service Bus Dashboard"

46.1 What is Service Monitoring?

You can monitor Oracle Service Bus at runtime to know how many messages in a particular service have processed successfully and how many have failed.

The Oracle Service Bus monitoring framework provides access to information about the number of messages that were processed successfully or failed, which project the service belongs to, the average execution time of message processing, and the number of alerts associated with the service. Using the Oracle Service Bus Administration Console you can view monitoring statistics for the period of the current aggregation interval or for the period since you last reset statistics for this service or since you last reset statistics for all services. Using the public APIs you can access only the statistics since the last reset.

46.1.1 About the Oracle Service Bus Monitoring Framework

Monitoring in Oracle Service Bus involves monitoring of the operational resources, servers, and Service Level Agreements (SLAs). Figure 46–1 is an illustration of the architecture of the Oracle Service Bus monitoring framework.

Figure 46–1 Monitoring Framework in Oracle Service Bus
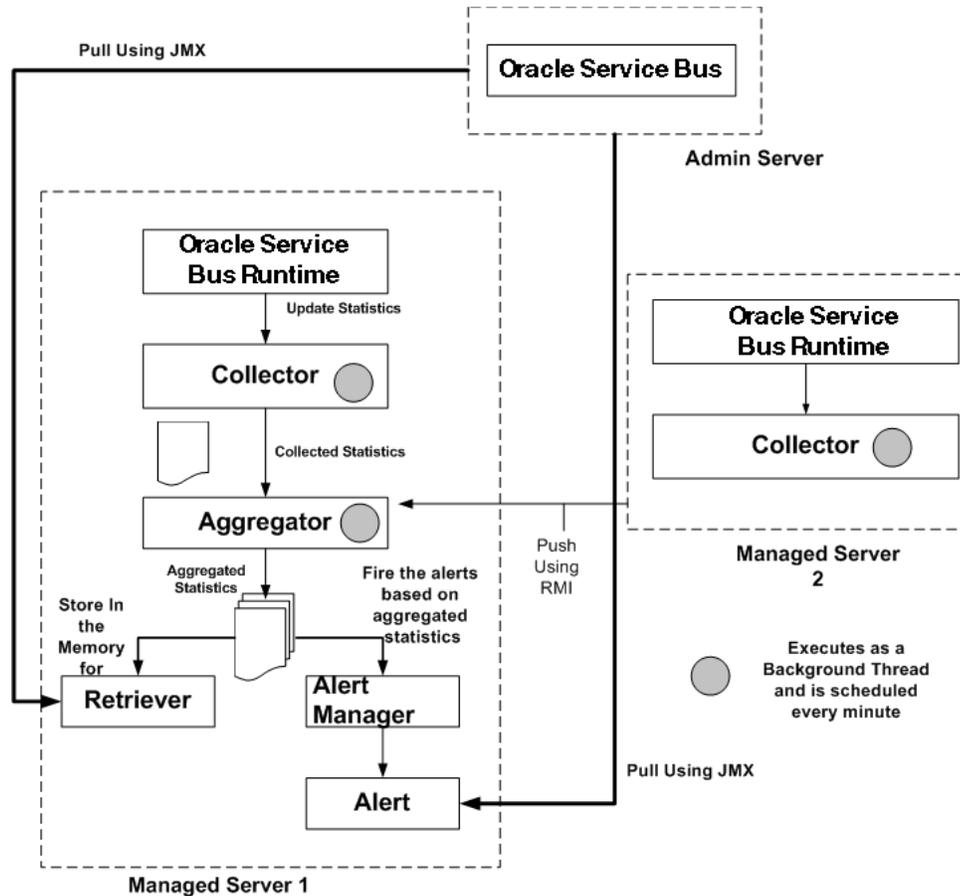
Oracle Service Bus monitoring architecture consists of the following components:

- Collector–Each Managed Server in a cluster hosts a Collector. The Collector collects statistics on operational resources at regular intervals of time, which is managed in a RMI object. It also keeps samples history within the aggregation interval for the collected statistics.

  Oracle Service Bus runtime invokes a collector at the beginning of each minute. At every system-defined checkpoint interval, it stores a snapshot of current statistics into a persistent store for recovery purposes and sends the information to the Aggregator in raw format, as raw format is optimized for fast collection and small footprint.

  > **Note:** An operational resource is defined as the unit for which statistical information can be collected by the monitoring subsystem. Operational resources include proxy services, business services, service level resources such as Web Services Definition Language (WSDL) operations, flow components in a pipeline, and endpoint URIs.

- Aggregator–The Aggregator is present only on only one Managed Server. The server on which this resides is selected arbitrarily when you generate domain using the config wizard. It aggregates all the statistics that are collected from all Managed Servers across all Managed Servers in a cluster.

  Oracle Service Bus runtime invokes the aggregator twenty-five seconds past each minute, to enable collectors to collect data and send it to the aggregator. At system-defined checkpoint intervals, each Managed Server in the cluster sends a snapshot of its contributions to the Aggregator. Data structures in aggregator are optimized for aggregating and retrieving data.

- Retriever–The Retriever retrieves the statistics that are stored in the memory. This is present only in that Managed Server, which contains the aggregator.

- Alert Manager–The alert manager fires alerts based on the aggregated statistics. This is present only in that Managed Server, which contains the aggregator.

The Collector collects the updated statistics from Oracle Service Bus runtime and sends it to Aggregator.The Aggregator aggregates the statistics over the aggregation interval. The aggregated statistics are pushed to the Alert Manager. The Alert Manager triggers alerts based on these statistics. The aggregated statistics are also stored and can be retrieved by the Retriever. The following steps are executed when you monitor a service in Oracle Service Bus runtime:

In a cluster, all the statistics which are collected in the Managed Servers are pushed to the aggregator. You can access the cluster wide statistics from the Service Health tab. For more information, see Section 46.5.3, "How to Access Statistics in a Cluster." You can also access statistics using APIs. For more information, see Section 46.5.2, "How to Access Statistical Information Using the JMX Monitoring APIs."

The alerts are pulled from the Managed Server that hosts the aggregator, and they are displayed in the Oracle Service Bus Administration Console.

46.2 Aggregation Intervals

In Oracle Service Bus, the monitoring subsystem collects statistics, such as message count over an aggregation interval. The aggregation interval is the time period over which statistical data is collected and displayed in the Oracle Service Bus Administration Console. Statistics which are not based on an aggregation interval are meaningless. In addition to statistics collected over well-defined aggregation interval you can also collect cumulative statistics.

46.2.1 The Refresh Rate of Monitoring Data

Aggregation interval is a moving window, which always refers to an interval of time in minutes, hours or days. It does not move with infinite granularity or precision, but at regular intervals of time called the sampling interval. This enables an aggregation interval to move smoothly and produce accurate statistics.

Figure 46–2   Illustration of Aggregation Interval and Sampling Interval
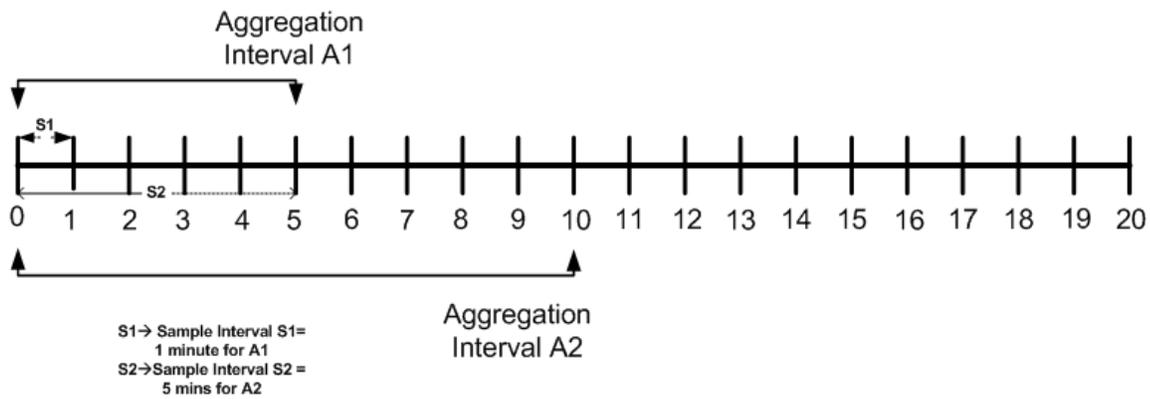
Figure 46–2 is an illustration of the of Aggregation interval. For example aggregation interval `A1` is set at five minutes and aggregation interval `A2` has been set at ten minutes. Oracle Service Bus runtime collects statistics for the service with aggregation interval `A1` for every minute (`S1`). It aggregates the statistics at the end of the aggregation interval.

Similarly for aggregation interval `A2` it collects statistics for every five minutes (`S2`). Intervals S1 and S2 are called sampling intervals. For more information about sample interval, see Section 46.2.2, "Sample Intervals Within Aggregation Intervals."

46.2.2 Sample Intervals Within Aggregation Intervals

In Oracle Service Bus runtime statistics are computed at regular intervals, within every aggregation interval. These regular sub intervals are known as the sample interval. The duration of the sample interval depends on the aggregation interval. Table 46–1 gives the length of sample interval for different aggregation intervals:

Table 46–1   Sample Interval

Aggregation Interval	Sample Interval
1, 2, 3, 4, and 5 minutes	1 minute
10, 15, 20, 25, and 30 minutes	5 minutes
40, 50, and 60 minutes	10 minutes
90 and 120 minutes	30 minutes
3, 4, 5, and 6 hours	1 hour
8, 10, and 12 hours	2 hours
16, 20, and 24 hours	4 hours
2, 3, 4, 5, 6, and 7 days	1 day

46.2.3 How to Set the Aggregation Interval for Monitoring Data

Aggregation interval in Oracle Service Bus has the following properties:

- You can track statistics for a service over only one aggregation interval.

- You cannot set an arbitrary value for an aggregation interval. You must choose from one of the values in the list.

- You can set the aggregation interval for the following:

> – A service–You must set the aggregation interval for a service in Operation tab of View a Proxy service or View a Business Service page. For more information about how to set aggregation interval for a service, see Section 26.10, "Setting the Aggregation Interval for a Service."
>
> – An alert rule–You must set the aggregation interval for an alert rule by editing Conditions on View Alert Rule Details page. For more information about how to edit conditions for an alert rule, see Section 26.23.2, "Defining Alert Rule Conditions."

You can only specify an aggregation interval less than or equal to seven days.

46.2.4 What are the Consequences Of Changing Aggregation Interval Of A Service?

When you modify the aggregation interval of a service, the statistics of the service in the current aggregation interval is reset. However, the status of the endpoint URI for the service remains unaffected by the change in the aggregation interval. A running count metrics of the service is not reset when modify the aggregation interval.

46.3 What are the Consequences of Renaming or Moving a Service?

When you rename or move a service within Oracle Service Bus, all the monitoring statistics that have been collected in the Oracle Service Bus Administration Console are lost. All current aggregation interval and cumulative metrics are reset and the service is monitored from start. If endpoint URI for a service was marked offline before it was renamed or moved, then after you rename or move the service, the URIs are marked online again and the status of the URI is displayed as online.

46.4 What Statistics Are Available for Oracle Service Bus Services?

You can monitor services in Oracle Service Bus and collect statistics for all services. Monitoring system in Oracle Service Bus supports the following types of statistics:

- Counter–A counter simply keeps track of the count of events in Oracle Service Bus runtime such as number of messages received and number of failovers. This is scalar and takes on integral values.

- Interval–An interval keeps track of time elapsed between two well-defined events. This tracks the total, average, minimum, and maximum of such events in Oracle Service Bus. This takes on integral and non-integral values.

- Status Type–A Status statistic keeps track of the status. Using this you can keep track of the initial status and the current status of the object.

For more information about different types of statistics, see Appendix F, "Monitoring Statistics in Oracle Service Bus." For more information on how to access statistics using APIs, see Section 46.5.2, "How to Access Statistical Information Using the JMX Monitoring APIs."

46.5 Accessing Statistical Information for Services

You can access the statistical information for a service through the Oracle Service Bus Administration Console or directly by using the JMX monitoring APIs. This section describes accessing the information through the Oracle Service Bus Administration Console and the JMX monitoring APIs. For more information about accessing the statistical information through JMX monitoring APIs, see Appendix D, "JMX Monitoring API."

46.5.1 How to Access Service Statistics from the Oracle Service Bus Administration Console

You can access the service statistics from the Oracle Service Bus Administration Console for a stand alone server or a cluster. This section describes how to access statistics for a standalone server. For information on how to access service statistics for a cluster, see Section 46.5.3, "How to Access Statistics in a Cluster."

In the you can access the statistical information for services in the current aggregation interval or for time interval since the last reset. The Service Health tab of dashboard provides the statistical information for current aggregation interval and since last reset.

For more information, see Section 46.9.1, "How to Access Service Statistics for the Current Aggregation Interval" and Section 46.9.2, "How to Access Running Count Statistics for Services."

46.5.2 How to Access Statistical Information Using the JMX Monitoring APIs

You can also access statistical information directly from a program using the Java Management Extensions (JMX) monitoring APIs. Using the JMX monitoring APIs you can access only the running count statistics. The JMX monitoring APIs provide an efficient lower level support for bulk operations. For more information about using JMX monitoring APIs, see Appendix D, "JMX Monitoring API."

46.5.3 How to Access Statistics in a Cluster

In a cluster environment, statistics are available at individual Managed Server level and the cluster level. In Service Health tab choose Cluster or the name of a Managed Server from the Server list, to view statistics for the cluster or the individual Managed Server.

To get detailed statistics for a particular service in a cluster, access the service monitoring details page for the service from the Service Health tab. On the Service Monitoring Details page, you can access the cluster wide statistics by setting the Server list to Cluster. By setting it to the individual Managed Server value, statistics pertaining to that specific server can be viewed.

Set Display Statistics to Current Aggregation Interval to view cluster statistics in the current aggregation interval or Since Last Reset to view the running count statistics for the cluster.

46.5.4 How to Reset Statistics

You can reset the statistics of business services and proxy services from the Service Health tab of the dashboard or from the Service Monitoring Details page for a service. The following table describes how to reset statistics in each case.

To Reset Statistics from	Description
Service Health Tab	Click the Service Health tab to go to the Service Health page. Click Reset Statistics icon to reset statistics for a service. Click **Reset** All Statistics link to reset statistics for all the services for which monitoring is enabled.
	Note: The reset Statistics Icon is available only when you set Display Statistics to Since Last Reset.JMS
	For more information about how to reset statistics, see Section 26.7, "Resetting Statistics for Services."

To Reset Statistics from	Description
Service Monitoring Details Page	Click the name of the service in the alert history table or on the Service Health page to go to the Service Monitoring Details page. In the Service Monitoring Details page in the Display Statistics field select Since Last Reset. Click **Reset** Statistics to reset the statistics for the given service.

46.5.4.1 What are the Consequences of Resetting the Statistics?

When you reset statistics for a service in the Oracle Service Bus Administration Console, all the statistics collected for the service since the last reset is lost. You cannot undo this action. The status of endpoint URIs is not reset when you reset statistics.

46.6 The Role of Alerts in Service Monitoring

SLA alerts are raised in Oracle Service Bus to indicate potential violation of the Service Level Agreements (SLAs). You can use alerts for:

- Monitoring and generating email notification of WS-Security errors.

- Monitoring the number of messages passing through a particular pipeline.

- Detecting the violation of service level agreements with third-party products.

- Detecting a non-responsive endpoint.

Pipeline alerts can be raised in the message flow of the proxy service. You can use the alerts in a message flow for:

- Detecting errors in a message flow.

- Indicating business occurrences.

46.6.1 Assigning Severity for Alerts

You can configure the severity of an alert in an alert rule for SLA alerts or in the `Alert` action of a message flow of a proxy service. The severity level of alerts is user configurable and has no absolute meaning. You can configure alerts with one of the following levels of severity:

- Normal

- Warning

- Minor

- Major

- Critical

- Fatal

The alert destinations are notified when an alert is raised. For more information in alert destinations, see Section 46.6.7, "What are Alert Destinations?"

46.6.2 What are SLA Alerts?

SLA alerts are automated responses to violations of Service Level Agreements (SLAs). These alerts are displayed on the Oracle Service Bus dashboard. They are generated when the service violates the service level agreement or a predefined condition.

To raise an SLA alert, you must enable SLA alerting both at the service level and at the global level. For more information about how to configure operational settings for services, see Section 46.7.1, "How to Configure the Operational Settings for a Service." The Alert History panel contains a customizable table displaying information about violations or occurrences of events in the system.

You must define alert rules to specify unacceptable service performance according to your business and performance requirements. Each alert rule lets you specify the aggregation interval for that rule when configuring the alert rule. This aggregation interval is not affected by the aggregation interval set for the service.

For more information about aggregation interval, see Section 46.2, "Aggregation Intervals." Alert rules also allow you to send notifications to the configured alert destinations. For information on defining alert rules, see Section 26.23, "Creating and Editing Alert Rules."

46.6.2.1 A Sample Use Case for SLA Alerts

Consider the following use case to verify the service level agreements:

Assume that a particular proxy service is generating SLA alerts due to slow response time. To investigate this problem, you must log in to the Oracle Service Bus Administration Console and review the detailed statistics for the proxy service. At this level, you can able to identify that, a third-party Web service invocation stage in the pipeline is taking a lot of time and is the actual bottleneck.

You can use these alerts as the basis for negotiating SLAs. After successfully renegotiating SLAs with the third-party Web service provider, you must configure alert metrics to track the Web service provider's compliance with the new agreement terms.

46.6.3 What are Pipeline Alerts?

Pipeline alerts can be generated in a message flow whenever you define an `Alert` action available under the reporting category in the message flow.

You can also define conditions under which a pipeline alert is triggered using the conditional constructs available in the pipeline editor such as XQuery Editor or an if-then-else construct, and you select an Alert Destination resource to sent the alerts to.

You can have complete control over the alert body including the pipeline, and context variables. Also you can extract the portions of the message. For more information about how to configure `Alert` actions in a stage, see Section 21.25, "Adding Alert Actions." When the alert action is executed the alerts are notified to the appropriate alert destinations.

You can obtain an integrated view of all the pipeline alerts generated by a service on the dashboard page in the Oracle Service Bus Administration Console.

46.6.3.1 A Sample Use Case for Pipeline Alerts

Consider the following use case for pipeline alerts:

Consider a case when you want to be notified when special business conditions are encountered in a message flow. You can configure an alert action is a message flow to raise alerts when such predefined conditions are encountered. You can also configure email alert destination to receive an email notification of the alert. You can also send the details to the email recipient in the form of payload.

For example, in the case of a proxy service that routes orders to a purchase order website, and you want to be notified when an order exceeding $10 million is routed.

For this you must configure an alert action in the appropriate place in the pipeline, with the condition and configure email alert destination with the email information and use it as the target destination in the alert action. You can also include the details of the order in the form of payload.

You can also use pipeline alerting to detect errors in a message flow. For example, in the case of a proxy service that validates the input document, you want to be notified when the validation fails so that you can contact the client to fix the problem.

For this you must configure an alert action within the error handler for the message flow of the proxy service. In the action you can include the actual error message in the fault variable and other details in the SOAP header, to be sent as the payload. You can also configure additional alert destinations using an alert destination resource in the alert action.

46.6.4 How to View or Delete SLA Alerts

In the Oracle Service Bus Administration Console the extended alert history page for the SLA alerts contains information about all the SLA alerts that have been generated in the domain. You can view all the alerts that were triggered or search for specific alerts from the table. For more information about data displayed in the extended SLA alert history page, see Section 26.3, "Locating Alerts."

You can delete the alerts from the Extended Alert History page or the View Alert Details page. You can filter your search using the Extended Alert History Filters pane. For more information on how to filter your search, see Section 46.6.6.1, "How to Filter a Search for SLA Alerts."

To view a pie or bar chart of the alerts, click View Bar Chart or View Pie Chart in the page. Click **Purge SLA Alert History** to delete all the SLA alerts. You can also purge the alerts based on the date and time they were raised.

46.6.5 How to View or Delete Pipeline Alerts

The extended alert history page for the pipeline alerts contains information about all the pipeline alerts that have been generated in the domain. You can view all the alerts that were triggered or search for specific alerts from the table. For more information about data displayed in the extended pipeline alert history page, see Section 26.3, "Locating Alerts."

You can delete the alerts from the Extended Pipeline Alerts page or from the View Alert Details page. For more information on how to filter your search, see Section 46.6.6.2, "How to Filter a Search for Pipeline Alerts." To view a pie or bar chart of the alerts, click View Bar Chart or View Pie Chart link in the page. Click **Purge Pipeline Alert History** to delete all the pipeline alerts. You can also purge the alerts based on the date and time they were raised.

46.6.6 How to Filter a Search for Specific Alerts

Use Extended Alert History to filter a search for specific alerts. The following sections describe how to search for SLA alerts and pipeline alerts using extended alert history filters.

46.6.6.1 How to Filter a Search for SLA Alerts

Use Extended Alert History Filters pane to filter a search for SLA alerts. Table 46–2 describes the various criteria on which you can filter SLA alerts.

Table 46–2   Search Criteria for SLA Alerts

Search Criterion	Description
Date Range	Use this to search for pipeline alerts that were generated in the given interval of time. You can set the interval in one of the following ways: ■ All ■ Set timestamp interval in MM/DD/YY HH:Min:SS AM/PM format. ■ Alerts generated during the given time interval in the format days–hours–minutes
Alert Severity	Specify the level of severity. The search result includes all the alerts that have the specified level of severity and above.
Service	Use this to search for a specific service.
Service Type	This is updated automatically when you search for a specific service.
Alert Name	Use this to search by alert name.

46.6.6.2 How to Filter a Search for Pipeline Alerts

Use Extended Alert History Filters pane to filter a search for pipeline alerts. Table 46–3 describes the various criteria on which you can filter pipeline alerts.

Table 46–3   Search Criteria for Pipeline Alerts

Search Criterion	Description
Date Range	Use this to search for pipeline alerts that were generated in the given interval of time. You can set the time interval in one of the following ways: ■ All ■ Set timestamp interval in MM/DD/YY HH:Min:SS AM/PM format. ■ Alerts generated during the given time interval in the format days–hours–minutes
Alert Severity	Specify the level of severity. The search result includes all the alerts that have the specified level of severity and above.
Service	Use this to search for a specific service.
Service Type	This is updated automatically when you search for a specific service.
Alert Summary	Use this to search by alert summary.

46.6.7 What are Alert Destinations?

Alert destinations are resources to which alerts are sent. Alert rules require that you select a predefined alert destination for handling alerts. Alert destinations give you the flexibility configure whether alerts are sent to SNMP traps, are collected for reporting, are logged to the local server's alert log, and sent to email recipients or JMS destinations.

With alert logging enabled, the Oracle Service Bus Administration Console provides information about the alerts generated due to SLA violations or as a result of alert actions configured in the pipeline. The dashboard page displays the overall health of

Oracle Service Bus. It provides an overview of the state of the system comprising server health, services health, and alerts.

For more information about how to interpret the information on the dashboard, see Section 46.9, "The Oracle Service Bus Dashboard."

46.6.7.1 Email

Email alert destinations allow you to receive messages when alerts are raised in the Oracle Service Bus Administration Console. To configure this alert destination you have to use the SMTP server global resource or a JavaMail session in the Oracle WebLogic Server. For more information on configuring a default SMTP Server resource, see Section 31.8, "Configuring a Default SMTP Server." For more information about configuring JavaMail sessions, see "Configure Access to JavaMail" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

The SMTP server global resource captures the address of the SMTP server, port number, and if required, the authentication credentials. The authentication credentials are stored inline and are not stored as a service account. The alert manager makes use of the email alert destination to send the outbound email messages when both pipeline alerts and SLA alerts are generated. When an alert is delivered, an email metadata consisting of the details about the alert is prefixed to the details of the payload that is configured.

You can specify the email ID of the recipients in the Mail Recipients field. For more information about configuring an email alert destination, see Section 6.3.1, "Adding Email Recipients."

46.6.7.2 SNMP Traps

The Simple Network Management Protocol (SNMP) traps allow any third-party software to interface monitoring service level agreements within Oracle Service Bus. By enabling the notification of alerts using SNMP, Web Services Management (WSM) and the Enterprise Service Management (ESM) tools can monitor SLA violations and pipeline alerts by monitoring alert notifications.

SNMP is an application-layer protocol which allows the exchange of information on the management of a resource across a network. It enables you to monitor a resource and, if required, take some action based on the data obtained from the resource.

Both the SNMP version 1 and SNMP version 2 are supported by Oracle Service Bus. SNMP includes the following components:

- Managed Resource
- Management Information Base(MIB)
- SNMP Agent
- SNMP Manager
- Network Management System (NMS)

For more information, see Appendix E, "SNMP Components."

46.6.7.3 Reporting

The Reporting destination lets you send notifications of pipeline alerts or SLA alerts to the custom reporting provider that can be developed using the reporting APIs provided with Oracle Service Bus. This allows third parties to receive and process alerts in custom Java code.

46.6.7.4 Alert Logging

Each alert destination lets you configure whether the alerts sent to that destination are logged. Logged alerts are sent to the local alert log. Each Oracle Service Bus server has its own alert log. In a cluster, the Admin Server collects the alert logs from all Managed Servers and aggregates the alerts in the Oracle Service Bus Administration Console.

46.6.7.5 JMS

Java Messaging Service (JMS) is another destination for pipeline alerts and SLA alerts. You must configure a JNDI URL for the JMS destination for alerts. When you configure an alert rule to post a message to a JMS destination, you must create a JMS connection factory and a queue or topic, and target them to the appropriate JMS server in the Oracle WebLogic Server Administration Console.

For information on how to do this, see Configuring a JMS Connection Factory and JMS Resource Naming Rules for Domain Interoperability in "Methods for Configuring JMS System Resources" in *Oracle Fusion Middleware Configuring and Managing JMS for Oracle WebLogic Server*. When you define the JMS alert destination you can either use a destination queue or a destination topic. The message type can be bytes or text. For more information about how to configure JMS alert destination see Section 6.3.2, "Adding JMS Destinations."

46.7 What are Operational Settings for a Service?

Operational settings enable you to control the state of a service in the Oracle Service Bus Administration Console. Table 46–4 describes operational settings for services in the Oracle Service Bus Administration Console.

Table 46–4   Operational Settings for Services in the Administration Console

Operational Settings	Usage	Default Value When a Service is Created
State	Use this to enable or disable a service.	Enabled
Monitoring	Use this to enable service monitoring at a specific level or above. You can also use this to disable service monitoring for proxy services.	Disabled
Aggregation Interval	Use this to set the aggregation interval for the service.	10 minutes
SLA Alerts	Use this to enable SLA alerting for services at a specific level of severity or above. You can also use this to disable SLA alerting for a service.	Enabled
Pipeline Alerts	Use this to enable pipeline alerting for proxy services at a specific severity level or above. You can also use this to disable pipeline alerting for proxy services.	Enabled at Normal level or higher
Reports	Use this to enable or disable message reporting for proxy services.	Enabled at Normal level or higher

Table 46–4   (Cont.) Operational Settings for Services in the Administration Console

Operational Settings	Usage	Default Value When a Service is Created
Logs	Use this to enable logging at a specific severity level or above. You can also use this to disable logging for proxy services.	Enabled at Debug level or higher
	In proxy service log actions, the severity level of the log actions must match the Logging severity level on the proxy service operational settings.	
	Note: To see log data in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:	
	■ Minimum severity to log: Info	
	■ Log file: Info	
	■ Standard out: Info	
	For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.	
Execution Tracing	Use this to enable or disable execution tracing for proxy services.	Disabled
	Note: To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:	
	■ Minimum severity to log: Info	
	■ Log file: Info	
	■ Standard out: Info	
	For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.	
Message Tracing	Use this to enable or disable message tracing for proxy services at a specific detail level or above. You can also set the payload limit (in kilobytes) and the default encoding.	Disabled
	Note: See note for Execution Tracing.	
Offline Endpoint URIs	Use this to enable or disable non responsive endpoints for business services. You can also specify the interval of time to wait before retrying the offline endpoint URI. You can enable or disable offline URIs for business services only.	Disabled
Throttling State	Use this to enable or disable throttling for a business service.	Disabled
Maximum Concurrency	Use this to restrict the number of messages that can be concurrently processed by a business service.	0

Table 46–4   (Cont.) Operational Settings for Services in the Administration Console

Operational Settings	Usage	Default Value When a Service is Created
Throttling Queue	Use this to restrict the maximum number of messages in the throttling queue.	0
Message Expiration	The maximum time interval (in milliseconds) for which a message can be placed in throttling queue.	0
Result Caching State	Use this to enable or disable result caching for a business service	Enabled Result caching is globally enabled by default

The operational settings at the service level is overridden by the global settings. For more information about configuring operational settings globally, see Section 46.7.2, "How to Configure the Operational Settings at the Global Level."

46.7.1 How to Configure the Operational Settings for a Service

You can enable or disable the operational settings for an individual service from the Operation Settings view of the View a Proxy Service (see Figure 46–3) or View a Business Service page (see Figure 46–4). For more information, see Section 19.1, "Creating and Configuring Business Services" and Section 20.1, "Creating and Configuring Proxy Services."

Some operational settings such as service state, monitoring, SLA alerting, and pipeline alerting can enabled or disabled through public APIs. For more information, see *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

Figure 46–3   View a Proxy Service Operational Settings

Figure 46–4   View a Business Service Operational Settings

You can perform the following operational settings for proxy services and business services:

- State
- Aggregation Interval
- Monitoring
- SLA Alerting
- Pipeline Alerting
- Message Reporting
- Tracing
- Pipeline Logging
- URI Offline Interval
- Throttling settings

For a detailed description of the usage of each operational setting for business services, see Section 26.9, "Configuring Operational Settings for Business Services." For a detailed description of the usage of each operational setting for proxy services, see Section 26.8, "Configuring Operational Settings for Proxy Services."

46.7.2 How to Configure the Operational Settings at the Global Level

You can access the Global Settings page from the operations module. You can use the Global Settings page (see Figure 46–5) to configure the operational settings for services. Table 46–5 describes the usage of the operational settings at the global level.

Table 46–5   Usage of Operational Settings at the Global Level

Operational Settings	Usage
Monitoring	Use this to enable monitoring for all services in a domain. Click the check box associated with **Enable Monitoring** to enable or disable monitoring for all the services in the domain.
SLA Alerting	Use this to enable SLA alerting for all services in a domain. Click the check box associated with **Enable SLA Alerting** to enable or disable monitoring for all the services in the domain.
Pipeline Alerting	Use this to enable pipeline alerting for proxy services in a domain. Click the check box associated with **Enable Pipeline Alerting** to enable or disable monitoring for all the services in the domain.
Message Reporting	Use this to enable message reporting for proxy services in a domain. Click the check box associated with **Enable Reporting** to enable or disable monitoring for all the services in the domain.
Logging	Use this to enable logging for proxy services in a domain. Click the check box associated with **Enable Logging** to enable or disable monitoring for all the services in the domain.
Result Caching	Use this to enable or disable result caching for business services in a domain. Even though you can configure business services to use result caching, this global option must be enabled for result caching to work. If you disable result caching globally, cached results for all business services enabled with result caching are flushed.

Figure 46–5   Global Settings Page

For more information, see Section 27.2, "Enabling Global Settings."

> **Notes:** The Enable Monitoring option lets you enable or disable monitoring of all services that have individually been enabled for monitoring. If monitoring for a particular service has not been enabled, you must first enable it and set the aggregation interval on the Manage Monitoring page before the system starts collecting statistics for that service.
>
> Enable or disable these settings at the global level in conjunction with the settings at the service level to effectively enable or disable them. The operational settings at the global level supersede the operational settings at the service level.

46.7.3 Updates to Operational Settings During Import of Oracle Service Bus Configurations

When a service is overwritten by the way of importing configuration from a config jar, the operational settings of this service can be also be overwritten. To preserve the operational settings during import, you must set the Preserve Operational Settings flag to true while importing the service. For more information, see Section 46.8.4, "What Happens to Alert Rules When You Import Oracle Service Bus Configurations?"

46.7.4 Updates to Global Settings During the Import of Oracle Service Bus Configurations

When you import Oracle Service Bus configurations, if the config jar that is being imported also contains the global settings of the domain from which it is being imported, then these domain level settings can get overwritten. In order to prevent this, set Preserve Operational Settings flag to true while importing the service.

46.7.5 How to Preserve Operational Settings During the Import of Oracle Service Bus Configuration Through APIs

You can preserve operational settings during import of Oracle Service Bus configurations using APIs. For more information, see ALSBConfigurationMBean documentation in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*. Modify the MBean as shown in Example 46–1 to preserve the during the import.

Example 46–1   Preserve Operational Settings During the Import of Oracle Service Bus Configurations Through APIs

```
/**
 // Imports a configuration jar file, applies customization, activates it and exports the resources
again
 // @throws Exception
 /
static private void simpleImportExport(String importFileName, String passphrase) throws Exception {
SessionManagementMBean sm = ... // obtain the mbean to create a session;
// obtain the raw bytes that make up the configuration jar file
 File importFile = new File(importFileName);
 byte[] bytes = readBytes(importFile);
// create a session
 String sessionName = "session." + System.currentTimeMillis();
 sm.createSession(sessionName);
// obtain the ALSBConfigurationMBean that operates on the
 // session that has just been created
 ALSBConfigurationMBean alsbSession = getConfigMBean(sessionName);
```

```
// import configuration into the session. First we upload the
 // jar file, which will stage it temporarily.
 alsbSession.uploadJarFile(bytes);
// then get the default import plan and modify the plan if required
 ALSBJarInfo jarInfo = getImportJarInf();
 ALSBImportPlan importPlan = jarInfo.getDefaultImportPlan();
// preserve operational values
 importPlan. setPreserveExistingOperationalValues(true);
// Modify the plan if required and pass it to importUploaeded method
 ImportResult result = alsbSession.importUploaded(importPlan);
// Pass null to importUploaded method to mean the default import plan.
 //ImportResult result = alsbSession.importUploaded(null);
// print out status
 if (result.getImported().size() > 0) {
     System.out.println("The following resources have been successfully imported.");
     for (Ref ref : result.getImported()) {
         System.out.println("\t" + ref);
     }
 }
if (result.getFailed().size() > 0) {
     System.out.println("The following resources have failed to be imported.");
     for (Map.Entry e : result.getFailed().entrySet()) {
         Ref ref = e.getKey();
         // Diagnostics object contains validation errors
         // that caused the failure to import this resource
         Diagnostics d = e.getValue();
         System.out.println("\t" + ref + ". reason: " + d);
     }
// discard the changes to the session and exit
     System.out.println("Discarding the session.");
     sm.discardSession(sessionName);
     System.exit(1);
}
// peform the customization to assign/replace environment values and
 // to modify the references.
...
// activate the session
 sm.activateSession(sessionName, "description");
// export information from the core data
 ALSBConfigurationMBean alsbcore = getConfigMBean(null);
 //export the information at project level, pass only a collection of project refs to this method
 byte[] contentsProj = alsbcore.exportProjects(Collections.singleton(Ref.DEFAULT_PROJECT_
REF),null);
// the byte contents can be saved as jar file
}
```

46.8 SLA Alerting Functionality in Oracle Service Bus

In Oracle Service Bus you must define conditions based on which alerts are raised. The conditions are configured in an SLA alert rule. The alert rule also configures the severity level and an alert destination for an alert.

46.8.1 How to Configure SLA Alert Rules

SLA alerts are automated responses to SLAs violations, which are displayed on the dashboard. You must define alert rules to specify unacceptable service performance according to your business and performance requirements. When you configure an alert rule, you must specify the aggregation interval. The alert aggregation interval is not affected by the aggregation interval set for the service. For more information about

aggregation interval, see Section 46.2, "Aggregation Intervals."

Creating an alert rule involves the following steps:

- General Configuration–defines the name, description, summary, duration, severity, frequency, state of the enabled alert rule and other general characteristics.

- Define Condition–defines one or more conditions that trigger the alert rule. Additionally, you must define the aggregation interval for the condition on this page.

> **Note:** You can create alert rules even if you have not enabled for monitoring for a service.

For more information about creating an alert rule, see Section 26.23, "Creating and Editing Alert Rules."

For a service for which monitoring is enabled, Alert rule is evaluated at discrete intervals. Once an alert rule is created it is first evaluated at the end of the aggregation interval, and after that at the end of each sample interval. For example, if the aggregation interval of an alert rule is five minutes, it is evaluated five minutes after it is created, and then every minute after that (since sample interval for five minutes, is one minute).

If a rule evaluates to false no alert is generated. If the rule evaluates to true the alert generation is governed by the Alert Frequency. If the frequency is `Every Time`, an alert is generated every time an alert rule evaluates to true. If the frequency is `Notify Once`, an alert is generated only if no alert is generated in the previous evaluation. In other words, an alert is generated the first time the alert rule evaluates to true and no more notifications are generated until the condition resets itself and evaluates to True again.

46.8.2 How to Lookup or Edit Existing Alert Rules

The View Alert Rule Details page displays complete information about a specific alert rule, as shown in Figure 46–6. You can view the details of the alert rule in this page. You can edit an alert rule configuration from this page. For more information about how to edit an alert rule, see Section 26.23, "Creating and Editing Alert Rules."

Figure 46–6   View Alert Rule Details Page

General Configuration	
Rule Name	newbsRule
Rule Description	
Alert Summary	
Alert Destination	default/NewalertDest
Start Time (HH:MM)	
End time (HH:MM)	
Rule Expiration Date (MM/DD/YYYY)	
Rule Enabled	true
Alert Severity	Fatal
Alert Frequency	Every Time
Stop Processing More Rules	false
Conditions	
Condition Expression	Aggregation Interval 0 Hour(s) and 10 Minutes Success Ratio (%) = 100

Back | Edit

46.8.3 How to Rename Alert Rules

You can rename the alert rules from the SLA Alert Rules tab of View a Business Service or View a Proxy Service page. To rename an alert rule click Rename Alert Rule icon. Enter the new name for the alert rule in New Alert Rule Name field of the Rename Alert Rule window. Click **Rename**. Click **Update** and activate the session to complete. For more information, see Section 26.21, "Viewing Alert Rules." The Rename icon for the renamed alert is now disabled.

46.8.3.1 What are the Consequences of Renaming an Alert Rule?

When alerts are triggered, they are listed on the alert history page. Click **View Alert Rule Details** action icon to access the alert rule page. However, when you rename an alert rule, you cannot access the alert rule by clicking the View Alert Rule Details action from the alert history page, for the alerts that were raised before it was renamed.

You can access the Alert Details page from the alert history page for the alerts that are raised before-and after renaming the alert rule. The alert name is greyed in the Alert Details page for the alerts that were raised before the alert rule was renamed. When you rename an alert details icon for the renamed alert gets disabled.

Similar limitations exist when you attempt to access the alert rule page by clicking the alert name link on the alert details page. The alert name generated by alerts rules that are later renamed refers to an outdated name. You can view the old alert rule, but the name is grayed out indicating that the alert rule has been renamed.

When you rename an alert rule, the conditions on which a rule is based are preserved. The aggregation interval of the alert rule is also preserved. The alert is raised at the end of the first aggregation interval after the alert rule is renamed. For example, consider an alert rule a1 with aggregation interval five minutes. If the alert rule is renamed to a2 after two minutes of execution the next alert under the name a2 is generated three minutes after the is renamed.

46.8.4 What Happens to Alert Rules When You Import Oracle Service Bus Configurations?

You can preserve the alert rule configurations when you import Oracle Service Bus configurations. When you import Oracle Service Bus configurations, the operational settings are preserved. When services with alert rules exists in a jar that you import but does not exist in the target domain, then these services along with the alerts rules are imported as is.

However, if the same service exists in the target domain as well, then the import behavior is governed by the state of the Preserve Operational Settings during the import operation. For more information on how to preserve operational settings, see Section 46.7.3, "Updates to Operational Settings During Import of Oracle Service Bus Configurations."

46.9 The Oracle Service Bus Dashboard

The Oracle Service Bus dashboard displays service health, server health and details of all the alerts that have been triggered in Oracle Service Bus runtime. The dynamic refresh of this display is controlled by the Dashboard Refresh Rate setting in the User Preferences page. The default option for this setting is No Refresh. These alerts can be the result of SLA violations or pipeline alerts.

Service Level Agreements (SLAs) are agreements that define the precise level of service expected from the business and proxy services in Oracle Service Bus. Pipeline alerts are defined in the message flow for business purposes such as record the number of message that flow through the message pipeline, to track occurrences of certain business events, or to report errors but not for the health of the system.

Each row of the table displays the information that you have configured, such as the severity, timestamp, and associated service. Clicking the Alert Name link displays Alert Details page for more details about the SLA alert. This helps you to analyze the cause of the SLA alert. Clicking the Alert Summary link displays the Alert Details for more details about the pipeline alert. This helps you to analyze the cause of the SLA alert.

From the dashboard, you can drill-down into the system and easily find specific information, such as the average execution time of a service, the date and time an alert occurred, or the duration for which server has been running.

The following sections helps you to understand the information displayed on Oracle Service Bus dashboard.

Figure 46–7   Oracle Service Bus Dashboard for SLA Alerts
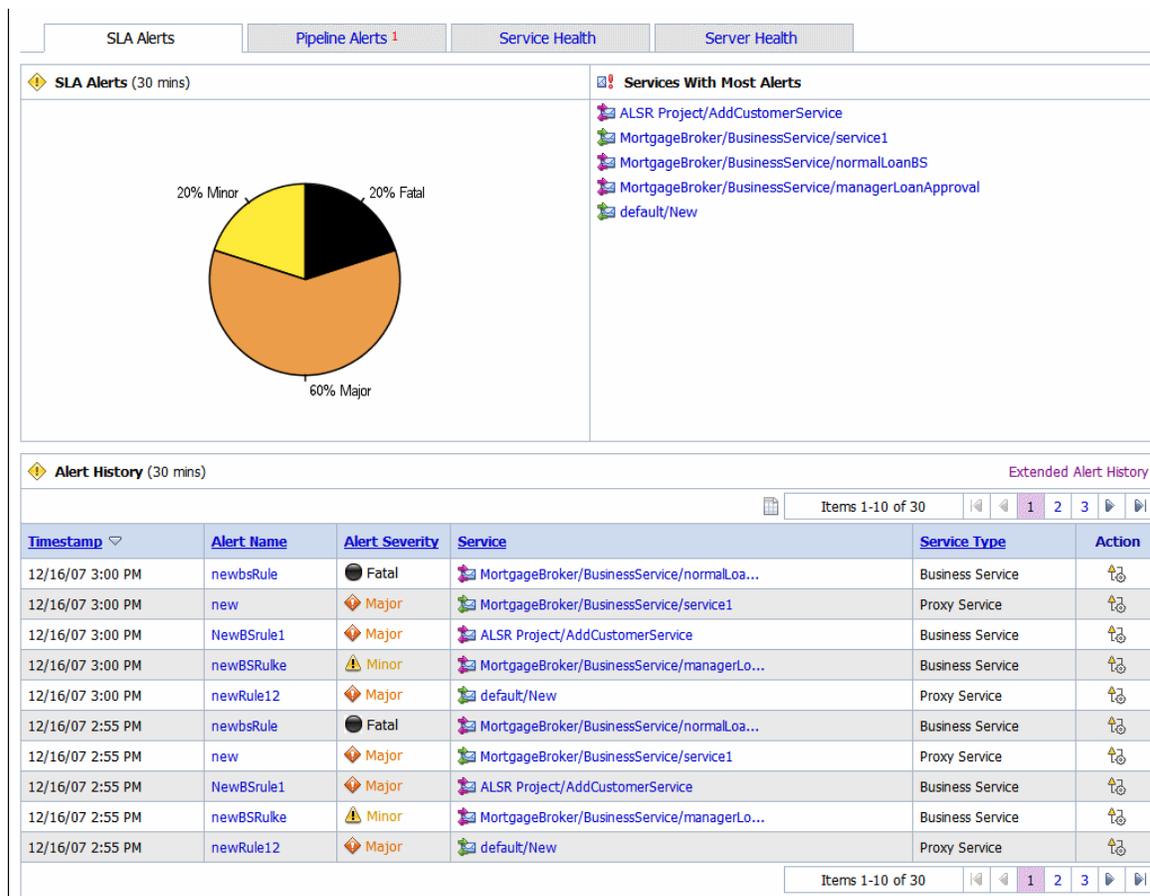
46.9.1 How to Access Service Statistics for the Current Aggregation Interval

Click the Service Health tab to access the Service Health page. The Service Health page is displayed.

This is a dynamic view of statistical data collected by each service. This view is available when you select Current Aggregation Interval in the Display Statistics field. The aggregation interval displayed in this view determines the statistics that are displayed. For example, if the aggregation interval of a particular service is twenty minutes, that service's row displays the data collected in the last twenty minutes.

From this page you can view all services or search for services based on the given criteria. For more information about the statistics displayed in this page, in the Current Aggregation Interval view, see Section 26.11, "Viewing Service Metrics."

The Service Monitoring Details page provides you with two views of detailed information about a specific service. Figure 46–8 shows the Service Monitoring Details page for a business service in the current aggregation interval. Figure 46–9 shows the Service Monitoring Details page for a proxy service.

To access this page click the name of the service in the Service With Most Alerts section, Alert History table, or extended alert history table for SLA alerts and pipeline alerts. Also the name of the service in Service Health tab is a link to Service Monitoring Details page.

Figure 46–8   Service Monitoring Details for Business Service–Current Aggregation Interval

Figure 46–9 Service Monitoring Details for Proxy Services–Current Aggregation Interval

This is a dynamic view of statistical data collected by each service. This view is available when you select Current Aggregation Interval in the Display Statistics field. The aggregation interval displayed in this view determines the statistics that are displayed. For example, if the aggregation interval of a particular service is twenty minutes, this page displays the data collected in the last twenty minutes for that service.

For more information on different tabs available for a business service, see Section 46.9.2.1, "Service Metrics," Section 46.9.2.2, "Operations," and Section 46.9.2.5, "Endpoint URIs." For more information on different tabs available for a proxy service, see Section 46.9.2.1, "Service Metrics," Section 46.9.2.3, "Pipeline Metrics," Section 46.9.2.4, "Action Metrics," and Section 46.9.2.2, "Operations."

For more information about the statistics displayed in this page, in the Current Aggregation Interval view, see Section 26.11, "Viewing Service Metrics."

46.9.2 How to Access Running Count Statistics for Services

The running count statistics for a service are statistics that are available since the last reset.

Figure 46–10   Service Health Tab—Since Last Reset

This view is a running count of the service health metrics. This view is available when you select Since Last Reset in the Display Statistics field. The statistics displayed in each row are for the period since you last reset the statistics for an individual service or since you last reset the statistics for all services.

You can also reset statistics for selected services or for all services. For more information about the statistics displayed in this page, in the Since Last Reset view, see Section 26.11, "Viewing Service Metrics."

Figure 46–11   Service Monitoring Details Page for a Business Service—Since Last Reset

Figure 46–12   Service Monitoring Details Page for a Proxy Service–Since Last Reset

This view is a running count of the service monitoring metrics. This view is available when you select Since Last Reset in the Display Statistics field. The statistics displayed in each row are for the period since you last reset the statistics for an individual service or since you last reset the statistics for all services.

From this page you can view all services or search for services based on the given criteria. You can also reset statistics for this service. For more information about the statistics displayed in this page, in the Since Last Reset view, see Section 26.11, "Viewing Service Metrics."

You have the following tabs in the Service Monitoring Details page for each of the views:

46.9.2.1 Service Metrics

The Service Metrics (see Figure 46–13) view displays the metrics for a proxy service or a business service.

Figure 46–13   Service Monitoring Details Page for a Business Service-Service Metrics Tab

Service Metrics	Operations	Endpoint URIs

General	
SLA Alert Count	0 Alerts ; Normal 10/20/09 2:55 PM
Min Response Time	235 msecs
Max Response Time	235 msecs
Overall Avg. Response Time	235 msecs
Message Count	2
Error Count	1
Failover Count	0
Success Ratio	50.0%
Failure Ratio	50.0%
WS Security Errors	0
Validation Errors	N/A
Result Caching	
Hit Count	0
Hit Ratio	0%
Throttling	
Min Throttling Time	0 msecs
Max Throttling Time	0 msecs
Average Throttling Time	0 msecs

The Service Metrics tab displays the following types of metrics:

■ General–This section enables you to quickly view the status of the alerts and service level statistics for the service in the current aggregation interval. When you view the service level statistics for the time interval since the last reset, this displays all the metrics since they were last rest. For more information about the metrics displayed in this view, see Section 26.11, "Viewing Service Metrics."

■ Throttling –This section enables to view the throttling statistics for a business service. You can also see the minimum and maximum throttling time in milliseconds. For more information on throttling statistics, see Section 26.11, "Viewing Service Metrics."

46.9.2.2 Operations

These metrics are displayed for WSDL based services for which you have defined operations. The Operations tab (see Figure 46–14) displays the statistics for the operation defined in a WSDL based service. For more information statistics displayed in this tab, see Section 26.12, "Viewing Operations Metrics for WSDL-Based Services."

Figure 46–14   Service Monitoring Details Page-Operation View

Service Metrics	Operations	Endpoint URIs				
				Items 1-1 of 1	◁ ◁ 1 ▷ ▷	
Operation Name			**Message Count**	**Error Count**	**Min Response Time**	**Max Response Time**	**Avg. Resp. Time**
addCustomer			0	0	0 msecs	0 msecs	0 msecs
				Items 1-1 of 1	◁ ◁ 1 ▷ ▷	

Back   Reset Statistics   Refresh

46.9.2.3 Pipeline Metrics

This view (see Figure 46–15) gives information on various components of the pipeline of the service. The Pipeline Metrics tab is available only for proxy services. For more information about the statistics displayed in this tab, see Section 26.13, "Viewing Pipeline Metrics."

Figure 46–15   Service Monitoring Details Page-Pipeline Metrics View for Proxy Services

Service Metrics	Pipeline Metrics	Action Metrics	Operations			
				Items 1-1 of 1 ◁ ◁ 1 ▷ ▷		
Component Name	**Message Count**	**Error Count**	**Min Response Time**	**Max Response Time**	**Avg. Resp. Time**	
Route to Loan Processing Service	0	0	0 msecs	0 msecs	0 msecs	
				Items 1-1 of 1 ◁ ◁ 1 ▷ ▷		

Back   Reset Statistics   Refresh

46.9.2.4 Action Metrics

This view (see Figure 46–16) presents information on actions in the pipeline of the service, displayed as a hierarchy of nodes and actions. Figure 46–16, for example, shows the Routing Table action nested under the route node Route to Loan Processing Service.

Similarly, actions nested under various branches of the Routing Table action are displayed accordingly. The table also contains entries that mark branches, in this case, the rows labeled Case and Default Case.

Figure 46–16   Service Monitoring Details Page-Action Metrics View for Proxy Services

Service Metrics	Pipeline Metrics	Action Metrics	Operations

Node	**Count**	**Avg. Time**	**Total Time**
Route to Loan Processing Service	---	---	---
Routing Table	3	100 msecs	302 msecs
Case	---	---	---
Routing	3	93 msecs	280 msecs
Default Case	---	---	---
Routing	0	0 msecs	0 msecs

Back   Reset Statistics   Refresh

The Action Metrics tab is available only for proxy services. For more information about the statistics displayed in this tab, see Section 26.14, "Viewing Action Metrics."

46.9.2.5 Endpoint URIs

The Endpoint URIs tab of the Service Monitoring page for a business service gives statistics of the various endpoint URIs configured for a business service and their status. For more information about the statistics displayed in this view, see Section 26.15, "Viewing Business Services Endpoint URIs Metrics."

Figure 46–17   Service Monitoring Details Page-Endpoint URI for Business Services

Service Metrics	Operations	Endpoint URIs						
							Items 1-1 of 1 ◁ ◁ 1 ▷ ▷▷	
Endpoint URI		**Message Count**	**Error Count**	**Min Response Time**	**Max Response Time**	**Avg. Resp. Time**	**Status**	**Action**
http://localhost:7001/njws_basic_ejb/NormalSimpleBean		2	0	162 msecs	235 msecs	198 msecs	Online	🗑
					Items 1-1 of 1 ◁ ◁ 1 ▷ ▷▷			

46.9.3 Viewing SLA Alerts in the Dashboard

You can view the details of SLA alerts in the SLA Alerts tab of the dashboard. Table 46–6 describes the dashboard for SLA alerts:

Table 46–6   Oracle Service Bus Dashboard for SLA Alerts

Section	Description
SLA Alerts	The pie chart shows the distribution of SLA alerts based on their severity for the duration set for alert history in the dashboard settings page. The severity level of alerts is user configurable and has no absolute meaning. For more information about alert severity, see Section 46.6.1, "Assigning Severity for Alerts."
	Click a specific area in the pie chart to display the Extended SLA Alert History page for alerts for the chosen level of severity and alert history duration.
Services With Most SLA Alerts	This section lists all the services with most SLA alerts in the current aggregation interval.
Alert History	This section gives details for all the SLA alerts generated during the alert history duration. For more information, see Section 46.9.4.1, "Viewing the Alert History for Pipeline Alerts."

46.9.3.1 Viewing the Alert History for SLA Alerts

The Alert History (Figure 46–7) for SLA alerts table shows all the SLA alerts, which have occurred in the alert history duration you have set in the User Preferences page. For more information about alert history table, see Section 26.1, "Viewing SLA Alerts."

To view a complete list of alerts, click Extended Alert History. For more information about Extended Alert History, see Section 46.6.4, "How to View or Delete SLA Alerts."

46.9.4 Viewing Pipeline Alerts in the Dashboard

You can view the pipeline alerts in the Pipeline Alerts tab of the dashboard (see Figure 46–18). Table 46–7 describes the dashboard for pipeline alerts.

Figure 46–18   Oracle Service Bus Dashboard for Pipeline Alerts
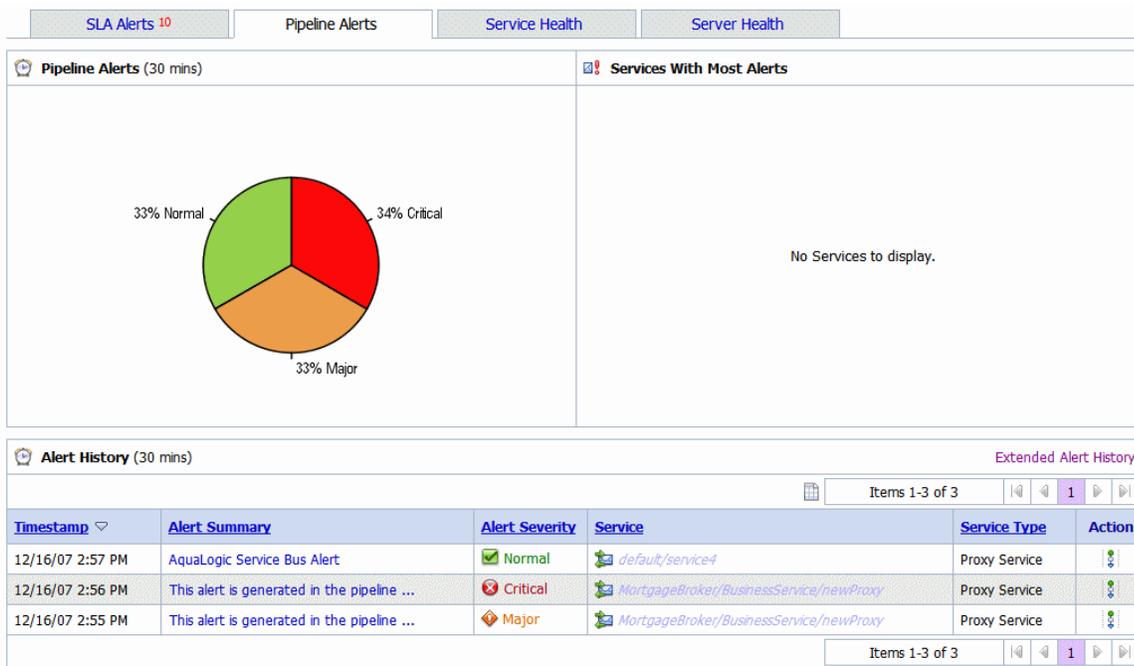
Table 46–7   Dashboard for Pipeline Alerts

Section	Description
Pipeline Alerts	The pie chart shows the distribution of pipeline alerts based on their severity for the duration set for alert history in the dashboard settings page. The severity level of alerts is user configurable and has no absolute meaning. For more information about alert severity, see Section 46.6.1, "Assigning Severity for Alerts."
	Click a specific area in the pie chart to display the Extended pipeline Alert History page for alerts for the chosen level of severity and alert history duration.
Service With Most Alerts	This section lists all the services with most pipeline alerts in the current aggregation interval.
Alert History	This section gives details for all the pipeline alerts generated during the alert history duration. For more information, see Section 46.9.4.1, "Viewing the Alert History for Pipeline Alerts."

46.9.4.1 Viewing the Alert History for Pipeline Alerts

The Alert History (Figure 46–7) for pipeline alerts table shows all the pipeline alerts, which have occurred in the alert history duration you have set in the User Preferences page. For more information about Alert History table, see Section 26.2, "Viewing Pipeline Alerts."

To view a complete list of alerts, click Extended Alert History. For more information about Extended Alert History, see Section 46.6.5, "How to View or Delete Pipeline Alerts."

46.9.5 Viewing Server Health in the Dashboard

You can view the server summary in Server Health tab of the dashboard.

Figure 46–19   Server Health
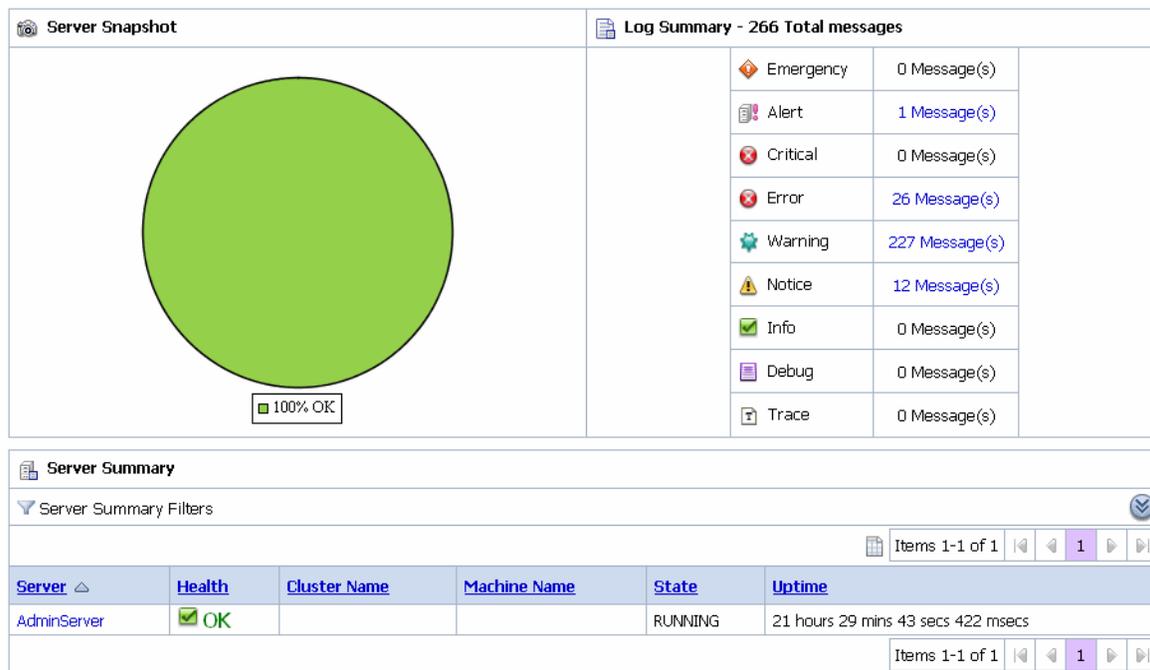
46.9.5.1 Viewing Log Summary

The Log Summary section in the Server Health tab displays the summary log for the servers associated with the domain. The domain log file provides a central location from which to view the overall status of the domain. Each server instance forwards a subset of its messages to a domain-wide log file.

By default, servers forward only messages of severity level `Notice` or higher. You can modify the set of messages that are forwarded. For more information, see "Understanding WebLogic Logging Services" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

If you configure the logging action in a pipeline, the log is forwarded to the Admin Server log. You can view the logging messages in the Server Health tab of the Oracle Service Bus Administration Console.

In a cluster, log action messages are forwarded to the Managed Server, so by default you cannot view these logging messages in the Oracle Service Bus Administration Console. To view managed server logs in the Oracle Service Bus Administration Console, configure Oracle WebLogic Server to forward these messages to the domain log. For information on how to do this, see "Create Log Filters" in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

To see the number of messages currently raised by the system, click the View Log Summary link in the Server Summary panel. A table is displayed that contains the number of messages grouped by severity, as shown in Figure 46–19.

You can view the log summary only if you posses administrator privileges in the Oracle WebLogic Server Console.

Table 46–8 describes the status messages in the log summary.

Table 46–8   Log Summary Messages

Message	Description
Emergency	This indicates that the server is in an unusable state. This severity indicates a severe system failure.
Alert	This indicates that a particular service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; immediate attention of the administrator is required to resolve the problem.
Critical	This indicates that a system or service error has occurred. The system can recover but there might be a momentary loss or permanent degradation of service.
Error	This indicates that a user error has occurred. The system or application can handle the error with no interruption. Limited degradation of service may occur.
Warning	This indicates that a suspicious operation or configuration has occurred. However, normal operations may not be affected.
Notice	This is an informational message with a higher level of importance than Info messages.
Info	This reports normal operations; a low-level informational message.
Debug	This shows the number of debug messages logged.

This display is based on the health state of the running servers, as defined by the WebLogic Diagnostic Service. For more information about the WebLogic Diagnostic Service, see *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

To view the domain log for a particular status of alert message, click the number corresponding with the status of alert message. shows an example of a domain log file displayed in the Oracle Service Bus Administration Console.

Figure 46–20   Domain Log File Entries

For more information about domain log file, see Section 26.18, "Viewing Domain Log Files." For more information, see Message Attributes in "Understanding WebLogic Logging Services" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log*

Messages for Oracle WebLogic Server. To display details of a single log file on the page, select the appropriate log, then click the View. You can also customize the Domain Log File Entries table to view the following additional information:

- Machine

- Server

- Thread

- User ID

- Transaction ID

- Context ID

- Timestamp

For additional description of these information, see Section 26.20, "Viewing Details of Domain Log Files.". For more information about how to customize the Domain Log File Entries table, see Section 26.19, "Customizing Your View of Domain Log File Entries."

46.9.5.2 Viewing Server Summary

You can view the Server Summary in the Server Health tab of the dashboard. In a single node domain, the Server Summary displays the summary of the Admin Server. In a cluster domain, it displays the health of all the servers in a cluster, in case of a cluster environment. For more information on Server Summary, see Section 26.16, "Viewing Server Information."

46.9.5.3 Viewing Server Details

To access this page, click the name of a server under server summary or click the name of a server in the Servers Summary page.

This page enables you to view more server monitoring details.

The information displayed on this page is a subset of the Monitoring tab in the Oracle Service Bus Administration Console Server Settings page. Table 46–9 describes the available information.

Table 46–9   Server Information

Information	Description
General	This provides general runtime information about the server. Click Advanced to view more information, such as Oracle WebLogic Server version or operating system name.
Channels	This provides monitoring information about each channel.
Performance	This provides information about the performance of the server.
Threads	This provides current runtime characteristics and statistics for the server's active executable queues.
Timers	This provides information about the timer used by the server.
Workload	This provides statistics for Work Managers, constraints, and policies configured on the server.
Security	This statistics for Work Managers, constraints, and policies configured on the server.
JMS	This lets you monitor JMS information about the server.

Table 46–9   (Cont.) Server Information

Information	Description
JTA	This provides the summary of all transaction information for all resource types on the server.

47

Reporting

This chapter describes the Oracle Service Bus (OSB) reporting framework, including the default JMS reporting provider and the reporting module in the OSB Administration Console.

Oracle Service Bus delivers message data and alerts to one or more reporting providers. Message data can be captured from the body of the message and from other variables associated with the message, such as header or inbound variables. Alert data contains information about Service Level Agreement (SLA) violations that you can configure to monitor proxy services. You can use the message or alert data delivered to the reporting provider for functions such as tracking messages or regulatory auditing.

Oracle Service Bus includes a JMS reporting provider for message reporting. The Reporting module in the Oracle Service Bus Administration Console displays the information captured from this reporting provider. If you do not wish to use the JMS Reporting Provider that is provided with your Oracle Service Bus installation, you can untarget it and create your own reporting provider using the Reporting Service Provider Interface (SPI). If you configure your own reporting provider for messages, no information is displayed in the Oracle Service Bus Administration Console. You must to create your own user interface. To capture SLA data, you must to create a reporting provider for alerts.

This chapter contains the following sections:

- Section 47.1, "About the Oracle Service Bus Reporting Framework"

- Section 47.2, "About the JMS Reporting Provider"

- Section 47.3, "How to Configure a Database for the JMS Reporting Provider Store"

- Section 47.4, "How to Enable Message Reporting"

- Section 47.5, "How to Stop a Reporting Provider"

- Section 47.6, "How to Untarget a JMS Reporting Provider"

- Section 47.7, "Using the Reporting Module in the Oracle Service Bus Administration Console"

47.1 About the Oracle Service Bus Reporting Framework

Oracle Service Bus contains an extensible framework for creating one or more reporting providers for messages or alerts.

To enable message reporting you must first create a Report action in the message flow for the proxy service. The Report action lets you extract information from each message and write it to the Oracle Service Bus Reporting Data Stream. You do not need to configure a report action for alert reporting. Alert data is always available in

the Reporting Data Stream. For more information, see Section 47.4, "How to Enable Message Reporting."

Information you need for creating your own reporting provider is located in `com.bea.wli.reporting` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*. The Java API Reference provides information about what you need to do for implementing a reporting provider, including how to package it, where it goes, how to deploy it, and the order of deployment. The reporting schema is MessageReporting.xsd, located in *OSB_ORACLE_HOME*/lib/sb-schemas.jar.

Figure 47–1 shows the reporting framework.

Figure 47–1 Reporting Framework
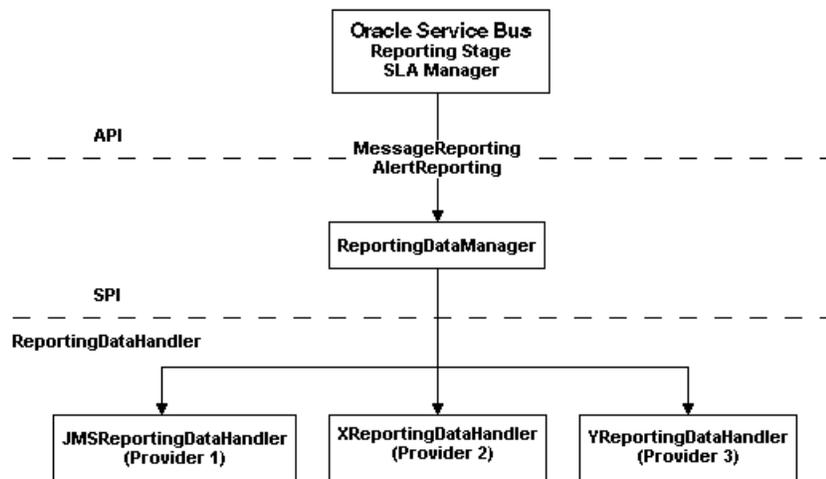
As shown in the Figure 47–1, both report messages and alerts are exported to reporting data streams. In the Report stage, information is extracted by the Report action from each message and written to the Reporting Data Stream with metadata that adheres to MessageReporting.xsd.

Similarly, the SLA Manager uses Reporting Data Manager APIs to write to the Alert Reporting Stream with metadata that adheres to the AlertReporting.xsd. To develop a reporting provider for alerts or your own message reporting provider, you need to implement ReportingDataHandler interface and use the ReportingDataManager class.

The ReportingDataHandler interface takes the reporting or alert data stream and processes it. It can either process or store a stream, or do both in a relational database, file, JMS queue, and so on. Depending on which stream you want to use, you need to implement the appropriate handle methods to process the data stream:

- Message Reporting Stream—the Report action of Oracle Service Bus runtime uses the following two handle methods to write to the Message Reporting Stream:

  ```
  handle(com.bea.xml.XmlObject metadata, String s)
  handle(com.bea.xml.XmlObject metadata, com.bea.xml.XmlObject data)
  ```

- Alert Reporting Stream—the Alert Manager uses the following `handle` method to write to the Alert Reporting Stream:

  ```
  handle(com.bea.xml.XmlObject metadata, String data)
  ```

The ReportingDataManager is a local server object that keeps a registry of reporting providers. Reporting providers implement the ReportingDataHandler interface. The ReportingDataManager provides operations to:

- Add and remove reporting data handlers.

- Export reporting data stream using various handle operations.

47.2 About the JMS Reporting Provider

The JMS Reporting Provider provides a pluggable architecture to capture the reporting information from each message using a Report action in a proxy service message flow. The default JMS Reporting Provider is automatically configured when you create an Oracle Service Bus domain. All messages across the cluster are aggregated and stored in a JMS Reporting Provider Data Store in a database specific format. This provider displays information from the JMS Reporting Provider Data Store.

> **Note:** If you do not want to use the default Oracle Service Bus JMS Reporting Provider, you should untarget the reporting provider and its corresponding data source in your domain, as described in Section 47.6, "How to Untarget a JMS Reporting Provider."

The JMS Reporting Provider consists of a producer and a consumer, which are decoupled to improve scalability. The producer is a JMS producer and the Message Driven Bean (MDB) acts as the JMS consumer, as shown in Figure 47–2.

Figure 47–2 JMS Reporting Provider
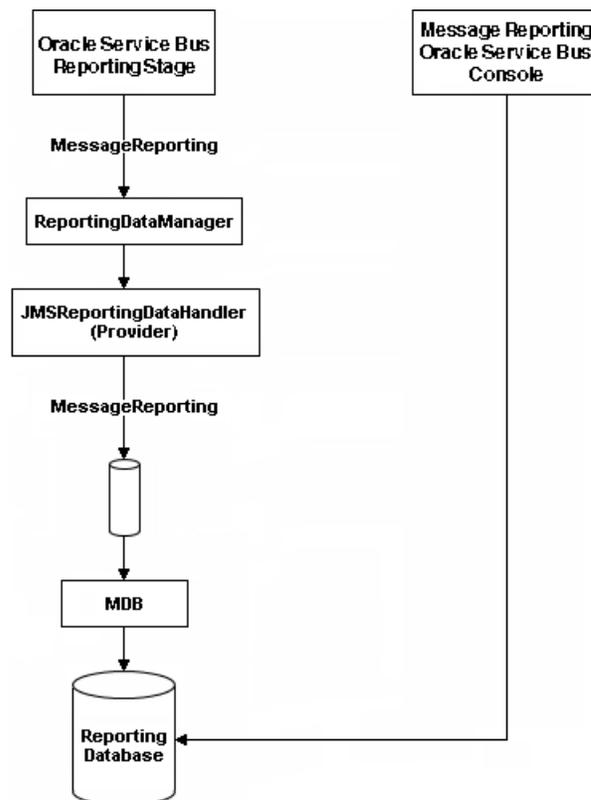
The Reporting stage contains the Report actions that collect the reporting information and dispatch the reporting stream to the JMS Reporting Provider through various handle operations in the ReportingDataManager. The JMSReportingDataHandler is the JMS producer of the reporting provider. The JMSReportingDataHandler takes the

reporting stream and logs the information to a JMS queue. The MDB listens to the JMS reporting queue, which processes the message asynchronously, and stores the data in the JMS Reporting Provider Data Store.

While the JMS Reporting provider processes reports generated by a Report action, it ignores reports generated due to SLA and Pipeline alerts. When writing a custom reporting provider, however, you are not restricted to that behavior. Depending on your business requirements you may want to process all or a subset of reporting data delivered by the reporting framework to your custom reporting provider.

47.2.1 Reporting Actions in Global Transactions

The JMS reporting provider uses a default queue connection factory that allows proxy service pipelines to execute reporting actions in the context of global transactions. The connection factory, defined in the jmsResources module, is called wli.reporting.jmsprovider.ConnectionFactory.

If you do not want report actions to execute in the context of a global transaction, modify the connection factory in the Oracle WebLogic Server Console by choosing **Services** > **Messaging** > **JMS Modules** > **jmsResources**, selecting the **Transactions** tab, and clearing the **XA Connection Factory Enabled** option.

Following is the logic that a proxy service pipeline uses when handling report actions using this connection factory:

- If a global transaction exists and the connection factory is XA-enabled (the default), the report is generated within the context of that global transaction.

- If a global transaction exists and the connection factory is not XA-enabled, the transaction is suspended and the report is generated within a local transaction.

- If a global transaction does not exist, the report is generated within a local transaction.

47.3 How to Configure a Database for the JMS Reporting Provider Store

Oracle Service Bus requires a database for the JMS Reporting Provider Data Store. The Apache Derby database that is installed with Oracle WebLogic Server is intended for development only, not for a production environment.

In a production environment you must use one of the supported databases. For the latest information about supported databases, see Supported Databases and Drivers in *Oracle Fusion Middleware Supported System Configurations* at:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

> **Note:** If you do not want to use the default Oracle Service Bus JMS Reporting Provider and hence do not configure a database, you should untarget the reporting provider and its corresponding data source in your domain, as described in

47.3.1 Configure the Reporting Data Source for Transactions

Make sure the JMS reporting provider data source supports transactions in one of the following ways. Incorrect configuration could result in exceptions.

- Configure the data source to use a supported XA driver.

- If you use a non-XA driver for the data source, be sure to select the following options in the data source configuration:

    – Supports Global Transactions

    – Logging Last Resource

       For more information about Logging Last Resource (LLR), see "Understanding the Logging Last Resource Transaction Option" in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

47.3.2 How to Configure a Database in a Development Environment

When you create an Oracle Service Bus domain, the Configuration Wizard does not create database tables automatically. In a development environment, the default JMS Reporting Provider checks whether tables exist for the specified database at runtime. If tables do not exist, the Reporting Provider creates them; if they do exist, the Reporting Provider uses them.

You can specify which database is used by the JMS Reporting Provider in one of the following ways:

- Run the reporting SQL scripts in your Oracle Service Bus domain. The scripts are located in *OSB_ORACLE_HOME*/dbscripts.

- When you create a domain in the Configuration Wizard, customize the JDBC settings on the reporting provider. For more information, see "Installing and Configuring Oracle Service Bus" in the *Oracle Fusion Middleware Installation Guide for Oracle Service Bus Installation Guide*. If you are using Apache Derby in your development environment, the default database with Oracle WebLogic Server, you need not specify a database in the Configuration Wizard.

47.3.3 How to Configure a Database for Production

For complete information about configuring a database for production, see the *Oracle Fusion Middleware Deployment Guide for Oracle Service Bus* in the following chapters:

- "Configuring a Single-Server Deployment"

- "Configuring a Clustered Deployment"

47.4 How to Enable Message Reporting

To receive report messages from either the JMS Reporting Provider, which is provided with Oracle Service Bus installation, or your reporting provider, you must first create a Report action in the message flow for the proxy service. The Report action lets you extract information from each message and write it to the Oracle Service Bus Reporting Data Stream. In the Report action, you must specify the information you want to extract from the message and add to the Oracle Service Bus Reporting Data Stream.

When configuring a Report action, use key values to extract key identifiers from the message. You can configure multiple keys. Information can be captured not only from the body of the message but any other variable associated with the message, such as header or inbound variables. For more information about message variables, see Chapter 39, "Message Context."

You can use any XML elements as a key:

```
<?xml version="1.0" encoding="utf-8"?>
<poIncoming>
```

```
    <areacode>408</areacode>
    <item-quantity>100</item-quantity>
    <item-code>ABC</item-code>
    <item-description>Medicine</item-description>
</poIncoming>
```

For example, you can specify the key as the itemcode, the value as `.//item-code` (an XPath expression), and the variable as message body (body), as shown in Figure 47–3.

Figure 47–3   Key Name and Value

If you are using the default JMS Reporting Provider, the keys and associated values are displayed in the Report Index column of the Summary of Messages table. If you configure multiple keys, the key-value pairs are displayed in the Report Index column with each key-value separated by a comma, as shown in Figure 47–4.

Figure 47–4   Keys and Associated Values Display

For information about creating a Report action or on how to view the Summary of Messages page, see the following in *Using the Oracle Service Bus Administration Console*:

- Section 21.27, "Adding Report Actions"
- Section 28.1, "Viewing Reporting Messages"

47.5 How to Stop a Reporting Provider

If you wish to stop a reporting provider when the server is running in the Oracle Service Bus domain, do the following:

1. Start the Oracle WebLogic Server Administration Console.

2. After logging into the Oracle WebLogic Server Administration Console, in the Domain Structure, click Deployments. The Summary of Deployments page is displayed.

3. In the Deployments table, select the check box next to the reporting provider you wish to stop.

4. Click **Stop** and after the list is displayed, choose the appropriate command.

5. After the Stop Application Assistant page is displayed, click Yes. The Deployments table shows that the state of the reporting provider is now Prepared.

47.6 How to Untarget a JMS Reporting Provider

If you do not want to use the default Oracle Service Bus JMS Reporting Provider (or a custom reporting provider), you should untarget it in your domain.

When you create an Oracle Service Bus domain, the Configuration Wizard, by default, includes the JMS Reporting Provider as a target in the deployment list. Untargeting the default JMS Reporting Provider and its associated data source lets you avoid benign JMS reporting errors at server startup. The reporting provider is the only Oracle Service Bus module that persists data in a database.

If you disable the reporting provider in the development environment, you can safely suppress Apache Derby database startup when you start the domain with the following command:

```
startWebLogic.cmd(.sh) noderby
```

This section provides different options for untargeting a reporting provider.

> **Note:** If no reporting provider exists, you can still define a Report action. However, no data is be written.

47.6.1 How to Untarget the Default JMS Reporting Provider Before Domain Creation

To untarget the default JMS Reporting Provider in the Oracle Fusion Middleware Configuration Wizard before a domain is created:

1. In the Configuration Wizard, select optional configurations for Deployments and Services and JMS File Store.

2. In the related wizard pages that follow, for each server, deselect the "JMS Reporting Provider" application and the "wlsbjmsrpDataSource" JDBC system resource.

When the wizard creates the domain, neither resource is targeted at server startup.

47.6.2 How to Untarget the JMS Reporting Provider when the Server is Running

To untarget a reporting provider when an Oracle Service Bus domain is running, do the following:

1. Start the Oracle WebLogic Server Administration Console and log in.

2. From the left panel, under Domain Structure, click **Deployments**. The Summary of Deployments page is displayed.

3. In the Deployments table, click the reporting provider you wish to untarget. The Settings page for the Reporting Provider is displayed.

4. Click the **Targets** tab.

5. Clear the appropriate check box.

6. Click **Save**. A message is displayed indicating that the settings have been successfully updated.

7. After you untarget the reporting provider, untarget the data source used by the reporting provider, as follows:

> **Note:** This step is only required for reporting providers that use their own data sources. To untarget the default JMS reporting provider in Oracle Service Bus installation you must perform the following steps.

   a. In the left panel, under Domain Structure, select **Services** > **JDBC** > **Data Sources**.

   b. In the Summary of JDBC Data Source page, click the name of the data source you wish to untarget. The Settings page for the data source is displayed.

   c. Click the **Targets** tab.

   d. Clear the appropriate check box.

   e. Click **Save**. A message is displayed indicating that the settings have been successfully updated.

   f. To activate the changes, in the Change Center, click Activate Changes.

47.6.3 How to Untarget the JMS Reporting Provider When the Server in Not Running

If the Oracle Service Bus domain is not running, you can use the WebLogic Scripting Tool (WLST) to remove the JMS Reporting Provider from the Oracle Service Bus domain. For more information about WLST, see *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

To untarget a reporting provider, complete the following steps:

1. If you have not already set up your environment to use WLST, see "Main Steps for Using WLST" in "Using the WebLogic Scripting Tool" in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

2. Invoke WLST Offline.

   ```
   C:>java com.bea.plateng.domain.script.jython.WLST_offline
   ```

3. To read the domain that was created using the Configuration Wizard execute:

   ```
   wls:/offline>readDomain("C:/oracle/user_projects/domains/base_domain")
   ```

4. To untarget the reporting provider data source execute:

   ```
   wls:/offline/base_domain>unassign("JdbcSystemResource", "wlsbjmsrpDataSource",
   "Target", "AdminServer")
   ```

5. To the reporting provider application execute:

   ```
   wls:/offline/base_domain>unassign("AppDeployment", "JMS Reporting Provider",
   "Target", "AdminServer")
   ```

6. To update the domain execute:

   ```
   wls:/offline/base_domain>updateDomain()
   ```

7. To close the domain execute:

   ```
   wls:/offline/base_domain>closeDomain()
   ```

8. Exit from the WLST command prompt execute:

   ```
   wls:/offline>exit()
   ```

After the Oracle Service Bus JMS reporting provider is untargeted, the Reporting module in the Oracle Service Bus Administration Console indicates that the reporting provider is not deployed.

> **Note:** In a cluster, the JMS Reporting Provider is targeted to Cluster. Therefore in a cluster, to view and purge messages, you must configure at least one Managed Server to run with the Administration server. If no Managed Servers are running, the Oracle Service Bus Administration Console displays a message that the reporting provider is unavailable.

47.7 Using the Reporting Module in the Oracle Service Bus Administration Console

The reporting module in the Oracle Service Bus Administration Console displays the information collected by the JMS Reporting Provider Data Store. The first page of the Reporting module, called the Summary of Messages, displays a table containing the extracted information and other information, such as the time the message was written to the database and the service with which the message is associated. You can customize the display of information on this page by filtering and sorting the data. You can also drill down to view detailed information about specific messages, including error information.

The Reporting module provides a purge function to help you manage your message data. You can purge all of the messages from the reporting datastore or base the purge on a time-range.

The JMS Reporting Provider Data Store requires a database. An evaluation version of the Apache Derby database is installed with Oracle WebLogic Server. You can use Derby for a development environment but not for production. Oracle Service Bus also supports databases from other vendors. Be sure to apply standard database administration practices to the database hosting the JMS Reporting Provider Data Store. For more information, see Section 47.3, "How to Configure a Database for the JMS Reporting Provider Store."

For more information about the reporting module, see Chapter 28, "Reporting."

47.7.1 Viewing the Summary of Message Reports

When you click Message Reports in the Reporting module of the Operations navigation bar, the Summary of Messages page is displayed. This page contains a table that provides a list of report messages sorted by the database timestamp.

Figure 47–5   Summary of Messages

If the messages are not filtered, the Summary of Messages table displays up to 100 of the latest messages based on the database timestamp. If you filter the messages, up to 1000 messages are displayed.

> **Note:**   The filter is not reset automatically. The filter remains in effect until you update it or reset it.

Table 47–1 describes the information in the Summary of Message Reports.

Table 47–1   Summary of Message Reports

Column Name	Description
Report Index	Displays the key-value pairs extracted from the message context variables or the message payload. For more information, see Section 47.2, "About the JMS Reporting Provider."
DB TimeStamp	Displays the time when the message was written to the database.
Inbound Service	Displays the inbound service associated with the message. Click the name of the service to go to View a Proxy Service page.
Error Code	Displays the error code associated with this message, if it exists. For more information about error codes, see Chapter 24, "Proxy Services: Error Handlers."

To search for specific messages, click Filter in the Summary of Messages Table to filter the display of messages.

You can filter report messages for a specified period of time, by the name of a service, by error code, and by report index. After you filter the messages, the title of the page changes to Summary of Filtered Messages. For information about using the Summary of Messages filter, see Section 28.1, "Viewing Reporting Messages."

To view more information about a report message, click the name of the message in the Report Index column. The View Message Details page is displayed.

47.7.2 Viewing Message Details

The View Message Details page displays complete information about the report messages.

Table 47–2 through Table 47–7 describe the information displayed in Report Message Detail Page.

Table 47–2   General Configuration Message Information

Message Information	Description
Message ID	An unique identifier for this message.
Database Timestamp	The time when the message was written to the database.
Time at point of Logging	The date and time, on the server machine, that the message was reported.
Server name	The name of the server from which this message was generated.
State	The state of the pipeline from which this message was generated. The pipeline can be in one of the following states: ■ REQUEST: indicates that the reporting action was executed in a request pipeline. ■ RESPONSE: indicates that the reporting action was executed in a response pipeline. ■ ERROR: the action was running in the service-level error handler.
Node Name	The pipeline node from which this message was generated.
Pipeline Name	The pipeline from which this message was generated.
Stage Name	The stage from which this message was generated.

Table 47–3   Inbound Service Message Information

Message Information	Description
Name	The inbound proxy service associated with this message. An inbound proxy service exchanges messages with client applications. The name is a link to the View a Proxy Service page. For more information about this page, see Section 20.5.1, "View a Proxy Service - Configuration Details Page."
URI	The URI associated with the proxy service.
Operation	The inbound operation associated with this message. Operations are the tasks performed by a pipeline or route node in the message flow associated with the service.

Table 47–4   Outbound Service Message Information

Message Information	Description
Name	The outbound business service associated with this message. An outbound business service exchanges messages with an Oracle Service Bus proxy service. Click the link to go to View Business Service Details page. For more information about this page, see Section 19.5.1, "View a Business Service - Configuration Details Page."
URI	The URI to the outbound business service end point.

Table 47–4   (Cont.) Outbound Service Message Information

Message Information	Description
Operation	The name of the operation invoked on the outbound service. Operations are the tasks performed by a pipeline or route node in the message flow associated with the service.

Table 47–5   Report Index Message Information

Message Information	Description
Report Text Index	Displays the key-value pairs extracted by a Report Action from the message context variables or the message payload. For more information, see Section 47.2, "About the JMS Reporting Provider."

Table 47–6   Fault Message Information

Message Information	Description
Error Code	The code associated with the error, if any. For more information, see Chapter 24, "Proxy Services: Error Handlers."
Reason	The reason the error occurred, if any.
Detail	The fault details associated with the error. These details, if present, are typically a stack trace of where a particular fault occurred. The stack trace may be truncated due to a size limitation in the database. The limit is 2048 characters.

Table 47–7   Report Body Message Information

Message Information	Description
Detail	Opens a new browser window that displays the report body in a browser. You can use an XQuery expression in a Report action to capture the report body text. For more information, see Section 21.27, "Adding Report Actions" and Section 23.1, "Creating and Editing Inline XQuery and XPath Expressions."

Detail opens a browser window that displays the report body in a browser. You can use an XQuery expression in a Report Action to capture the report body text.

47.7.3 How to Purge Messages from the Reporting Data Store

You can purge all of the messages from the reporting datastore or base the purge on a range of time. Message purging is an asynchronous process that occurs in the Oracle Service Bus Administration Console. This feature enables you to work with the Summary of Messages page in the Oracle Service Bus Administration Console while the purge occurs in the background.

The duration of time it takes a purge to complete depends on how many messages are in the purge queue. The deletion of messages is slowed if you search for reporting messages during the purge process. Moreover, the Summary of Messages page may display incorrect data as some data may not yet be purged.

Because the purge process is asynchronous and occurs in the background, the Oracle Service Bus Administration Console does not display any messages to indicate that a

purge is in process. However, if another user attempts to start a purge when a purge is in progress, the following message is displayed:

```
A Purge job is already running. Please try later.
```

48

Managing Endpoint URIs for Business Services

This chapter describes how to manage endpoint URIs in Oracle Service Bus business services, including configuring retries, marking non-responsive endpoints as offline, viewing endpoint metrics, and triggering alerts based on endpoint status.

An endpoint URI is the URL of an external service that is accessed by a business service. In Oracle Service Bus you must define at least one endpoint URI for a business service. When you define multiple endpoint URIs for a business service you must define one of the following load balancing algorithm:

- Round robin
- Random
- Random-weighted
- None

The load balancing algorithm controls the manner in which business service tries to access the endpoint URI. The status of the endpoint URI can be online or offline. For more information, see Section 26.9, "Configuring Operational Settings for Business Services."

This chapter contains the following sections:

- Section 48.1, "How to Configure a Business Service to Perform Retries"
- Section 48.2, "How to Mark a Non-Responsive URI Offline"
- Section 48.3, "Metrics for Monitoring Endpoint URIs"
- Section 48.4, "How to Mark an Offline URI as Online"
- Section 48.5, "How to Generate Alerts Based on Endpoint URI Status"

48.1 How to Configure a Business Service to Perform Retries

You can define the retry option for business services. The retry option specifies the maximum number of times a business service can attempt to access endpoint URIs after an initial failure. For example, consider the behavior of a business service B with endpoint URIs `eu1`, `eu2`, and `eu3`, when the retry count is set to `1`, `2`, and `4`.

When Retry Count = 1 – If business service B fails to process a request or is unable to access the endpoint URI `eu1`, it tries to process the request with `eu2` (retry 1). If the retry fails then the business service returns failure. The business service does not retry the third endpoint URI `eu3`.

When Retry Count = 2 – If business service B fails to process a request or is unable to access the endpoint URI `eu1`, it tries to process the request with `eu2` (retry 1). If the retry fails then the business service tries to process the request with `eu3` (retry 2). If the retry fails then the business service returns failure.

When Retry Count = 4 – If business service B fails to process a request or is unable to access the endpoint URI `eu1`, it tries to process the request with `eu2` (retry 1). If the retry fails then the business service tries to process the request with `eu3` (retry 2). Then the business service waits for a interval you have configured for retry iteration interval (in seconds) before trying `eu1` (retry 3). If this fails the business service retries `eu2` (retry 4). If the retry fails then the business service returns failure.

If the retry count is set to `0`, then the business service does not retry after the failure.

> **Note:** The order in which a business service retries the endpoints is controlled by the load balancing algorithm.

48.1.1 How to Suppress Retries in Case of Application Errors

A business service fails to process a request due to communication or application errors.

Communication errors occur due to random network problems. Retrying such requests with another endpoint URI can be successful. Application errors occur when a request is malformed or due to errors, and cannot be processed by any of the endpoints. You can turn off retry behavior for the application errors by setting Retry Application Errors to No in the Transport Configuration page for a business service, depending on the transport used.

48.2 How to Mark a Non-Responsive URI Offline

A communication error occurs each time a business service tries to access a non-responsive URI. You can configure a business service to mark non-responsive URIs offline. Doing so prevents a business service from repeatedly attempting to access a non-responsive URI and therefore avoids these communication errors.

To do so, you must enable the Offline Endpoint URIs operational setting for the business service. You can mark an endpoint URI offline temporarily or permanently as described in the following sections.

Mark an endpoint URI offline temporarily if you want the business service to automatically retry the same endpoint after a short interval of time; mark it offline permanently if you want the business service to treat the endpoint URI as offline until it is reset manually.

48.2.1 How to Mark an Endpoint URI Offline Temporarily

Mark an endpoint URI offline temporarily if you want the business service to automatically retry the same endpoint after a specified interval of time.

To mark an endpoint URI offline temporarily, you can specify a Retry Interval value in the Offline Endpoint URI operational setting for the business service. On encountering a communication error, the endpoint URI status is changed to Offline. When the retry interval has passed and this business service attempts to process a new request, it tries to access this endpoint URI. If this attempt is successful, then the endpoint URI is marked online again. If the attempt to access the endpoint URI fails, then the URI is marked offline again for the duration of the retry interval, and the cycle is repeated.

This configuration can be useful for the case in which a communication error is temporary and corrects itself. For example, when an endpoint becomes temporarily overloaded communication errors occur, but reverts to normal operation without requiring manual intervention.

48.2.2 How to Mark an Endpoint URI Offline Permanently

Mark an endpoint URI offline permanently if you want a business service to treat the endpoint URI as offline until you reset it manually.

To mark an endpoint URI offline permanently, you specify a Retry Interval value of `0` hours `0` min `0` sec in the Offline Endpoint URI operational setting for the business service. On encountering a communication error, the endpoint URI status is changed to Offline and remains offline until you mark the endpoint URI online again.

For more information, see Section 26.9, "Configuring Operational Settings for Business Services" and Chapter 46, "Monitoring Oracle Service Bus at Runtime."

This configuration is useful for a case in which a communication error is caused by a problem with the endpoint URI that must be resolved by manual intervention.

48.3 Metrics for Monitoring Endpoint URIs

You can monitor the metrics using the Oracle Service Bus Administration Console or the JMX monitoring APIs. For information on using the Oracle Service Bus Administration Console, see Section 46.5.1, "How to Access Service Statistics from the Oracle Service Bus Administration Console." For information on using the JMX monitoring APIs, see Appendix D, "JMX Monitoring API."

In the Oracle Service Bus Administration Console, the endpoint URI metrics are available on the Endpoint URIs tab within the Service Monitoring Details page for a service. This includes count, response time, and endpoint URI status metrics.

For more information about the statistics displayed in this view, see Section 26.15, "Viewing Business Services Endpoint URIs Metrics."

You can monitor the endpoint URIs using URI status statistics and URI level statistics. The following items describe the expected behavior when you monitor endpoint URIs:

- You can obtain the statistics only when you enable monitoring for a business service.

- When you rename or move a service, the URI level statistics is reset.

- When you change the aggregation interval, all the URI level statistics except the URI status are reset.

- When you reset statistics for the service (or reset all statistics), all the URI level statistics except the URI status are reset.

- When you add a new URI to an existing business service, the metrics for the new URI are collected automatically.

48.3.1 Endpoint URI Status

The Status statistic on the Oracle Service Bus Administration Console indicates whether the endpoint URI is online or offline. You can also obtain the status of an endpoint URI using the JMX monitoring APIs. Table 48–1 describes the possible states of an endpoint URI.

Table 48–1   Status of Endpoint URIs

Status	Description
Online	Implies the URI is online on a given server. In a cluster it implies that the URI is online for all servers.
Offline	Implies the URI is offline on a given server. In a cluster it implies that the URI is offline for all servers.
Partial	Implies that at least one server in the cluster reports a problem for that URI. This metric is available for clusters only.

> **Note:**   When a URI is associated with more than one business service, the same endpoint URI can have a different status for each of the business services.

48.3.2 Endpoint URI Performance Metrics

The endpoint URI performance metrics provide information on how many messages have been processed by a given endpoint and how many failed and their response times. You can use the following metrics for monitoring the endpoint URIs:

- Message Count
- Error Count
- Average Response Time
- Min Response Time
- Max Response Time

For more information about these statistics, see Section 26.15, "Viewing Business Services Endpoint URIs Metrics."

The following are the important properties of the endpoint URI statistics:

- You can obtain the statistics only when you enable monitoring for a business service.
- When you rename or move a service or change the aggregation interval, the URI level statistics is reset.
- When you add a new URI to an existing business service, the metrics for the new URI are collected automatically.

48.4 How to Mark an Offline URI as Online

You can mark an endpoint URI that is offline as online using the Oracle Service Bus Administration Console or by using the public APIs.

In the Oracle Service Bus Administration Console, you can mark an offline endpoint URI as online from the Service Monitoring Details page. Click the **Click to mark this endpoint URI online** icon in the Actions column of the Endpoint URIs tab. For more information, see Section 26.15, "Viewing Business Services Endpoint URIs Metrics."

All the endpoint URIs are marked online when:

- You add them to a business service
- You restart a server

- You enable a disabled service

- You rename or move a service

- A business service is able to successfully access the URI after the retry interval you have configured is past.

You can also use APIs to mark an offline endpoint URI as online. This is useful when the you have not enabled monitoring for a business service but you require to mark its endpoint URIs online. For more information, see com.bea.wli.monitoring.ServiceDomainMBean in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

When you mark an endpoint URI online in a cluster domain, it is marked online on all the Managed Servers.

48.5 How to Generate Alerts Based on Endpoint URI Status

If an endpoint URI is not accessible, the business service trying to access it receives a communication error.

In addition to configuring a business service to take a non responsive URI offline, as described in Section 48.2, "How to Mark a Non-Responsive URI Offline" you can raise an alert when a system encounters non-responsive URIs. You do this by configuring SLA alert rules based on endpoint URI status.

48.5.1 How to Configure an Alert Rule Based on Endpoint URI Status

You can configure the alert rule for a business service, based on the status of the Endpoint URI. Complete the tasks as described in Section 26.23.1, "Configuring General Information for Alert Rules."

Then complete the following tasks in the Alert Rule Conditions Configuration page, shown in Figure 48–1.

Figure 48–1   Alert Rule Condition Configuring page

1. In the Simple Expression section of the Alert Rule Conditions Configuration page, choose Status in the first list.

   The endpoint URI status based alert rule condition is comprised of:

2. A state transition clause–The state transition clause supports notification when any endpoint URI or all endpoint URIs change state from online to offline, or from offline to online. Choose one of the following options to identify the status for which you want to create a notification:

   - All URIs offline

   - All URIs online

   - Any URI offline

- Any URI online

For example, consider a business service for which two alert rules are configured, one based on `All URIs offline = True` condition and another on `Any URI offline = True` condition. If an alert based on `All URIs offline = True` condition is generated then it signifies a severe problem because all requests to this service are likely to fail until the situation is resolved.

However, if an alert based on `Any URI offline = True` is generated, it implies that the other endpoint URIs are responsive and subsequent requests may not fail.

> **Note:** All alert rules are independently evaluated. If alerts based on both (any or all URI) clauses have been configured for the same business service, it is likely that both alerts are generated simultaneously when the last endpoint URI is marked offline.
>
> If a business service has only one URI, the `All URIs offline = True` and `Any URI offline = True` clauses mean the same thing and so they behave in an identical manner.
>
> The evaluation of an alert rule condition based on a transition from offline to online behaves in a similar fashion except that it tracks any or all endpoint URIs being marked back to online state.

3. A server clause–The server clause lets you specify an alert trigger when a state transition occurs on any or all servers. The server clause is significant only in a cluster domain with multiple Managed Servers. Choose one of the following options to create the expression:

- Evaluate on all servers

- Evaluate on any server

A communication error can occur due to network problem on a computer hosting a Managed Server. Such an event is interpreted by the business service as the endpoint URI being non-responsive (although the remote endpoint being accessed is responsive). A communication error can also occur because the endpoint URI is not responding.

In the first case, the URIs are marked offline on only one server (on the computer with network problems) and online on all the other servers in the cluster. An alert condition based on `Evaluate on any server` clause generates an alert, but an alert condition based on the `Evaluate on all servers` clause does not generate an alert.

For the second case, the URI is marked offline on all the Managed Servers (one by one as each server tries to access that endpoint). As each Managed Server marks the endpoint URI offline, the alert rule condition based on `Evaluate on any servers` is met and an alert is generated. When the endpoint URI is marked offline on the last of the servers in the cluster domain, the alert rule condition based on `Evaluate on all servers` is also met and this alert is also generated.

> **Notes:** All alert rules are independently evaluated. If you have configured alerts based on both (any or all server) clauses for the same business service, it is likely that both alerts are generated simultaneously when the endpoint URI is marked offline on the last server in the cluster.
>
> In a single server domain, the `Evaluate on any server` and `Evaluate on all servers` clauses mean the same thing and behave in an identical manner.

When designing the alert rules for your system, you must choose one or more combinations of the clauses in accordance with your requirements.

You must set any one of the conditions to be True or False. These conditions can be evaluated on all servers or any server in a cluster.

> **Note:** To ensure that you do not miss any alert that is triggered due to frequent changes in the status of the URI, Oracle recommends that you set the aggregation interval for alert rules based on the status of the URI to one minute. For more information on aggregation interval, see Section 46.2, "Aggregation Intervals."

49

Execution Tracing

This chapter describes how to enable and use execution tracing for services in the Oracle Service Bus Administration Console.

Oracle Service Bus lets you trace messages without having to shut down the server. This feature is useful in both a development and production environments. Execution tracing allows administrators, support engineers, and systems engineers to troubleshoot and diagnose a message flow in one or more proxy services.

For example, if one of your proxy services is failing and you want to find out at which stage the problem exists, you can enable execution tracing for that proxy service. After tracing is enabled, the system logs various details extracted from the message flow such as stage name, name of the pipeline, and route node name.

The entire message context is also printed, including headers and message body. When a fault occurs in the message flow, additional details such as error code and reason are logged. Execution tracing occurs at the beginning and end of each component in the message flow, which includes stages, pipelines, and nodes. Actions are not traced individually.

49.1 How to Enable or Disable Execution Tracing

You can enable execution tracing from View a Proxy Service page of the Oracle Service Bus Administration Console.

> **Note:** To see tracing in the log file or standard out (server console), Oracle WebLogic Server logging must be set to the following severity levels:
>
> - Minimum severity to log: Info
>
> - Log file: Info
>
> - Standard out: Info
>
> For information on setting log severity levels, see "Using Log Severity Levels" in *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

To configure execution tracing:

1. Select the proxy service you want. For more information on how to find a proxy service, see Section 27.1.2, "Finding Proxy Services."

2. On the Operational Settings tab, select the **Enabled** option for Execution Tracing.

> **Note:** To enable execution tracing for all proxy services, use Smart Search to find all proxy services. With all proxy services displayed, select the **Exe. Tracing** option at the top of the table.

3. Click **Update** and activate the session.

You can view the execution tracing status for a proxy service in the Operation Settings tab of the View a Proxy Service page, Summary of All Services table, and Summary of Proxy Services table. For more information about Summary of All Services, see Section 51.3, "Managing Operational Settings for All Services," and for more information about Summary of Proxy Services, see Section 51.4, "Managing Operational Settings for Proxy Services."

49.2 How to Access Tracing Information

The tracing information is stored in the server directory logs at the following location:

DOMAIN_HOME/*domain*/servers/*server_name*/logs/*server_name*.log

Figure 49–1 shows a sample of the tracing log.

Figure 49–1   Tracing Log Example

```
weblogic.application.utils.StateMachineDriver.nextState(StateMachineDriver.java:26)
>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <Log Management> <svaidyan02> <xbusServer>
<[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS
Kernel>> <> <> <1167381864275> <BEA-170027> <The server initialized the domain log
broadcaster successfully. Log messages will now be broadcasted to the domain log.>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <WebLogicServer> <svaidyan02> <xbusServer> <Main
Thread> <<WLS Kernel>> <> <> <1167381864976> <BEA-000365> <Server state changed to ADMIN>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <WebLogicServer> <svaidyan02> <xbusServer> <Main
Thread> <<WLS Kernel>> <> <> <1167381864996> <BEA-000365> <Server state changed to RESUMING>
####<Dec 29, 2006 2:14:28 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381868541> <BEA-090171> <Loading the identity certificate and private key stored under
the alias DemoIdentity from the jks keystore file
C:\bea2613a\WEBLOG~1\server\lib\DemoIdentity.jks.>
####<Dec 29, 2006 2:14:29 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381869643> <BEA-090169> <Loading trusted certificates from the jks keystore file
C:\bea2613a\WEBLOG~1\server\lib\DemoTrust.jks.>
####<Dec 29, 2006 2:14:29 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381869713> <BEA-090169> <Loading trusted certificates from the jks keystore file
C:\bea2613a\JROCKI~1\jre\lib\security\cacerts.>
####<Dec 29, 2006 2:15:32 PM IST> <Warning> <Server> <svaidyan02> <xbusServer>
<DynamicSSLListenThread[DefaultSecure[1]]> <<WLS Kernel>> <> <> <1167381932743> <BEA-002611>
<Hostname "svaidyan02.apac.bea.com", maps to multiple IP addresses: 192.168.1.5,
172.22.56.120>
####<Dec 29, 2006 2:15:32 PM IST> <Notice> <Server> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381932753> <BEA-002613> <Channel "Default[2]" is now listening on 127.0.0.1:7021 for
```

> **Note:** The execution tracing pattern in the server log is identical to the execution tracing in the Test Console. For more information on tracing in the Test Console, see Section 33.1.4, "Tracing Proxy Services."

50

Throttling

This chapter describes how to enable and use throttling in Oracle Service Bus.

Throttling, which lets you control the amount of message traffic to a business service, helps improve performance and stability by preventing message overload on high-traffic business services.

This chapter contains the following sections:

- Section 50.1, "How to Enable Throttling"
- Section 50.2, "What are the Operational Settings for Throttling?"
- Section 50.3, "What Metrics are Available for Throttling?"
- Section 50.4, "How to Access Throttling Metrics"
- Section 50.5, "How to use Throttling Metrics for Alerting"
- Section 50.6, "How to use Throttling for Business Services with Multiple Endpoint URIs"

50.1 How to Enable Throttling

You must enable throttling for a business service from the Operational Settings tab of the View a Business Service page. In this tab, under Throttling select the **Enable** check box for the **Throttling State**. When you enable throttling for a business service you must specify values for **Maximum Concurrency**. You can also specify the **Throttling Queue** and **Message Expiration** for the business service. For more information, see Section 50.2, "What are the Operational Settings for Throttling?" and Section 26.9, "Configuring Operational Settings for Business Services."

Table 50–1 provides some important definitions in throttling.

Table 50–1   Important Definitions

What is …	Definition
Throttling queue	The priority queue in which the messages are enqueued when business services reach their maximum concurrency. Messages with higher priority are processed first. Messages are processed on a first-in first-out basis, if they have the same priority. You can assign priority to messages using the routing options.
	Notes:
	■ The greater the integer for priority, the higher is the priority of the message.
	■ You can configure only one queue for a business service
	A throttling queue is an in-memory queue. Messages that are placed in this queue are not recoverable when a server fails or when you restart a server.
	When you delete or rename a business service, all the messages in the throttling queue are discarded.
Message priority	The priority of a message in the throttling queue.
Expired message	A message that has been in the throttling queue for an interval greater than the value of message expiration. For more information on message expiration, see Table 50–2.

50.2 What are the Operational Settings for Throttling?

Table 50–2 describes the operational settings for throttling.

Table 50–2   Operational Settings for Throttling

Operation Setting	Use this to …
Throttling State	Enable or disable throttling for a business service.
	Note: When you disable throttling for a business service at runtime, all messages in the throttling queue are processed without throttling.
Maximum Concurrency	Restrict the number of messages that can be concurrently processed by a business service. This must be a positive integer. When this threshold is reached for a business service, all the incoming messages for the business service are placed in a throttling queue until the business service can accept more messages.
	Any change to this setting affects both new messages and those already in the queue. When you increase the value, the Oracle Service Bus runtime allows more messages to be sent to the business service after processing those in the queue first. When you decrease the value, the Oracle Service Bus runtime places any new messages in a throttling queue until the concurrency setting goes below the new threshold, if you have defined a throttling queue. If you have not defined a throttling queue, the messages are discarded.
	In a cluster environment, the number of messages that can be concurrently processed by a business service is equally divided among the Managed Servers.
	Note: You can set this operational setting only if you enable throttling for the business service.

Table 50–2   (Cont.) Operational Settings for Throttling

Operation Setting	Use this to …
Throttling Queue (operational setting)	Restrict the number of messages in the throttling queue. The length of the throttling queue must be positive integer. All the incoming messages beyond the maximum concurrency limit for the business service are placed in the throttling queue. When the queue is full, the message in the queue with the lowest priority will be removed from the queue if a new incoming message has a higher priority. If you set this length to be equal to zero, it implies that a throttling queue does not exist for the business service. Any change to this setting is dynamically implemented. When you decrease the value for this setting, all the messages beyond the new length are discarded. **Note**: You can set this operational setting only if you enable throttling for the business service. In a cluster environment, this is equally divided among the Managed Servers.
Message Expiration	Restrict the maximum time (in milliseconds) spent by a message in the throttling queue of a business service. This must be a positive integer. When this time has elapsed, the message is removed from the queue. These messages are referred to as expired messages. If the message expiration is set to zero for a service, the messages in throttling queue for this service will never expire. When you increase the value for this setting, the expiration time for the new messages and the messages that are already present in the queue is increased. When you decrease the value, all the messages that have exceeded the new value are immediately discarded. **Note**: You can set this operational setting only if you enable throttling for the business service.

50.3 What Metrics are Available for Throttling?

Table 50–3 describes the metrics that are available for throttling.

Table 50–3   Throttling Metrics

Metrics	Description
Max Throttling time	The maximum time spent by all messages in the throttling queue in milliseconds. If this value is greater than zero for a business service, it indicates that messages are placed in the queue for the business service.
Min Throttling time	The minimum time spent by all messages in the throttling queue in milliseconds. If this value is equal to zero some messages are not placed in the throttling queue.
Average Throttling time	The average the time spent by all messages in the throttling queue in milli seconds.

For more information about throttling metrics, see Section 50.4, "How to Access Throttling Metrics."

50.4 How to Access Throttling Metrics

You can access the throttling metrics from the Service tab of the Service Monitoring Details page in the Oracle Service Bus Administration Console. You can obtain the metrics for the current aggregation interval and for the interval since the last reset. For more information, see Section 46.5.1, "How to Access Service Statistics from the Oracle Service Bus Administration Console."

You can also access the metrics using JMX Monitoring APIs. For more information, see Section 46.9.3, "Viewing SLA Alerts in the Dashboard."

50.5 How to use Throttling Metrics for Alerting

You can define an SLA alert rule based on throttling metrics. You can define the alert rule based on Average Throttling Time. For more information, see Section 26.23, "Creating and Editing Alert Rules."

50.6 How to use Throttling for Business Services with Multiple Endpoint URIs

This topic provides information on using throttling for business services with multiple endpoint URIs.

In Oracle Service Bus, a business service can be associated with multiple endpoint URIs. For more information on endpoint URIs, see Chapter 48, "Managing Endpoint URIs for Business Services." When you associate a business service with multiple URIs, you must configure the maximum concurrency for the business service and not the individual URI. The maximum concurrency for each URI is set internally depending on the overall maximum concurrency and the load balancing weight, based on the following equation:

```
URI-specific max_concurrency = [User configured max_concurrency] X [weight]
```

For example, consider a business service B with three endpoint URIs eu1, eu2, and eu3. The load balancing algorithm is defined as random-weighted. The weights of the URIs are 1, 2, and 3 respectively. Assuming that you have defined a maximum concurrency of 10 for the business service, the URI specific maximum concurrency is 10, 20, and 30. The effective maximum concurrency of the business service B is 60. If the last endpoint URI that has a weight of 3 is offline, the effective maximum concurrency of the business service is 30.

> **Notes:** The weights for the URI when the load balancing is round robin or random is 1. When the load balancing is None the weight of the primary URI is 1 and the weight of the backup URI is 0. The weight of the backup URI becomes 1 when the primary URI goes offline.
>
> Messages for which the endpoint URI is overridden in routing options are not throttled. Messages are also not throttled with session stickiness enabled.

50.6.1 What Happens to Retried Messages During Throttling?

When failover is enabled on a service, retried messages are not throttled. The message is sent to the next URI regardless of the operational settings for throttling.

Messages that are expired or that are discarded, because the throttling queue is full or because the service reached its maximum concurrency, are not retried.

51

Managing Operational Settings Using Smart Search

This chapter describes how to use Smart Search in Oracle Service Bus. Smart Search lets you locate and manage resources such as proxy services, business services, alert rules for Service Level Agreements (SLA) violations, and alert destinations.

You can use Smart Search to find resources based on various criteria regardless of your role in Oracle Service Bus. For more information about tasks, which can be performed by each role Oracle Service Bus, see Chapter 45, "Roles in Oracle Service Bus."

You can also use results of a search to change the operational settings at the service level. For effectively enabling or disabling any operational setting, you must enable or disable the setting both at the global level and the service level. For more information, see Section 46.7.1, "How to Configure the Operational Settings for a Service."

You can access Smart Search from the Configuration module on the Operations navigator bar (see Figure 51–1).

Figure 51–1 Accessing Smart Search

There are the two levels of smart search available in Oracle Service Bus–basic search and advanced search. Using basic search you can search using the following basic criteria–Type, Name, and Path. Using advanced search, you can use additional filters to customize your search for each type. The following sections describe:

- Section 51.1, "Using Basic Search"

- Section 51.2, "Using Advanced Search"
- Section 51.4, "Managing Operational Settings for Proxy Services"
- Section 51.5, "Managing Operational Settings for Business Services"
- Section 51.6, "Managing Operational Settings for Split-Joins"
- Section 51.7, "Managing Operational Settings for Alert Destinations"
- Section 51.8, "Managing Operational Settings for SLA Alert Rules"

51.1 Using Basic Search

Basic Search helps you to search for resources using basic criteria such as resource type or the name of a resource. Table 51–1 describes the basic criteria you can use with the search functionality.

Table 51–1   Criteria for Basic Search

Criteria	Usage
Type	This search criterion is mandatory. Use the Type field to specify the type of resource. The field has the following options: - All Services: Choose this type when you search for both proxy services and business services. - Proxy Services: Choose this type when you search for proxy services only. - Business Services: Choose this type when you search for business services only. - Alert Destinations: Choose this type when you search for alert destinations. - SLA Alert Rules: Choose this type when you search for SLA alert rules. **Note**: To view all the resources of a given type, choose the resource type in the Type field and click View All.
Name	Optional. Enter the name of specific resource you want to find.
Path	Optional. Enter a path in the Path field to specify a location (path) of the resource.

You can use any one or a combination of these criteria.

51.2 Using Advanced Search

Use the advanced search if you want to configure your search with additional criteria. To use these filters, click Open Advanced Search Filters icon.The following sections describe the usage of advanced search filters in smart search functionality to manage operational settings.

51.3 Managing Operational Settings for All Services

You can search for proxy services and business services in the Oracle Service Bus Administration Console using filters in Smart Search. This section describes finding, viewing, and editing proxy services and business services.

51.3.1 Finding Services Using Search Filters

Table 51–2 describes the usage advanced filters to customize your search for both proxy services and business services.

Table 51–2   Using Advanced Filters to Find Proxy Services and Business Services

Filter	Usage
Service State	Choose Enabled or Disabled from the. Choose All to ignore this filter.
Service Monitoring	Choose Enabled or Disabled from the. Choose All to ignore this filter
Service SLA Alerts	Choose Enabled or Disabled from the. Choose All to ignore this filter.
Service Message Tracing	Choose Enabled or Disabled from the. Choose All to ignore this filter.

Click **Search** to find all services using the set criteria or click View All to view all the services.

51.3.2 Viewing and Editing Operational Settings

You can view results of the search for proxy service and business service in the Summary of All Services table.

Use this table to enable or disable services, monitoring, and SLA alerting functionality for both business service and proxy services. You can also use this to enable or disable message tracing, pipeline alerting, message reporting, execution tracing, and pipeline logging for proxy services. To enable or disable select the check box in the appropriate field and click Update.

The runtime effects of these settings also depend on corresponding settings at the global level. For more information about Global Settings, see Section 46.7.2, "How to Configure the Operational Settings at the Global Level." You can update the information for one or more services concurrently using this table.

The following information is displayed for all services:

Table 51–3   Understanding Summary of All Services

Property	Description
Name	The name assigned to the service. The name is a link to the View Proxy Service page for proxy services or View Business Services page for business services.
Path	The project associated with the service. If the service resides in a project folder, this folder is also listed. The path is displayed in the format: project-name/root-folder/ . . ./parent-folder The path is a link to the corresponding path in the Project Explorer.
Type	The type of the parent service: proxy service or business service.
State	The state of the service–Enabled or Disabled.

Table 51–3   (Cont.) Understanding Summary of All Services

Property	Description
Monitoring	The monitoring status of the service. Enabled or Disabled, and the level enabled at and above: ■ Service Monitoring ■ Pipeline Monitoring ■ Action Monitoring
SLA Alerts	The SLA alerting status. Enabled or Disabled, and the level enabled at and above: ■ Normal (N) ■ Warning (W) ■ Minor (Mn) ■ Major (Mj) ■ Critical (C) ■ Fatal (F)
Msg. Tracing	The message tracing status–Enabled or Disabled.
Pipeline Alerts	For proxy services only. The pipeline alerting status–Enabled or Disabled, and the level enabled at and above: ■ Normal (N) ■ Warning (W) ■ Minor (Mn) ■ Major (Mj) ■ Critical (C) ■ Fatal (F)
Reports	For proxy services only. The message reporting status of the service: Enabled or Disabled.
Logs	For proxy services only. The logging status–Enabled or Disabled, and the severity level at which it is enabled–Debug (D), Info (I), Warning (W), or Error (E).
Exe. Tracing	For proxy services and Split-Joins only. The tracing status–Enabled or Disabled
Caching	For business services only. Whether result caching is enabled or disabled.
Actions	For proxy services: The View Message Flow icon is a link to the pipeline for that proxy service.

Click **Reset** to discard any changes in the summary table and refresh the page with currently stored settings.

> **Note:**   You can edit the operational settings depending on the privileges of your role. For more information about roles, see Chapter 45, "Roles in Oracle Service Bus."

51.4 Managing Operational Settings for Proxy Services

You can search for proxy services in the Oracle Service Bus Administration Console using filters in Smart Search. This section describes finding, viewing, and editing proxy services.

51.4.1 Finding Proxy Services Using Search Filters

Table 51–4 describes advanced filters to customize your search for proxy services.

Table 51–4   Using Advanced Filters to Search for Proxy Services

Filter	Usage
State	Choose Enabled or Disabled. Choose All to ignore this filter.
Monitoring	Choose Enabled or Disabled. Choose All to ignore this filter.
SLA Alerts	Choose Enabled or Disabled. Choose All to ignore this filter.
Message Tracing	Choose Enabled or Disabled. Choose All to ignore this filter.
Pipeline Alerts	Choose Enabled or Disabled. Choose All to ignore this filter.
Reports	Choose Enabled or Disabled. Choose All to ignore this filter.
Logs	Choose Enabled or Disabled. Choose All to ignore this filter.
Execution Tracing	Choose Enabled or Disabled. Choose All to ignore this filter.

Click **Search** to find all proxy services using the set criteria or click View All to view all the services.

51.4.2 Viewing and Editing Operational Settings

You can view results of the search for proxy service in the Summary of Proxy Services table.

The Summary of Proxy Services view displays State, Monitoring, SLA Alerts, Message Tracing, Pipeline Alerts, Reports, Logs, Execution Tracing, Path, Type, Name, and Actions. For more information about the fields displayed in the summary of proxy services, see Section 51.3.2, "Viewing and Editing Operational Settings."

Use this table to enable or disable proxy services, monitoring, and SLA alerting, message tracing, pipeline alerting, message reporting, pipeline logging, and execution tracing. To enable or disable click the check box in the appropriate field and click Update. The runtime effects of these settings also depend on corresponding settings at the global level. For more information about Global Settings, see Section 46.7.2, "How to Configure the Operational Settings at the Global Level." You can update the information for one or more proxy services concurrently using this table.

Click **Reset** to discard any changes in the summary table and refresh the page with currently stored settings.

> **Note:**   You can edit the pipeline message flow depending on the privileges of your role. For more information about roles, see Chapter 45, "Roles in Oracle Service Bus."

51.5 Managing Operational Settings for Business Services

You can search for business services in the Oracle Service Bus Administration Console using filters in Smart Search. This section describes finding, viewing, and editing business services.

51.5.1 Finding Business Services Using Search Filters

Table 51–5 describes the filters you can use to customize your search for business services.

Table 51–5   Using Advanced filters to Search for Business Services

Filters	Usage
Service State	Choose Enabled or Disabled. Choose All to ignore this filter.
Monitoring	Choose Enabled or Disabled. Choose All to ignore this filter.
SLA Alerts	Choose Enabled or Disabled. Choose All to ignore this filter.
Message Tracing	Choose Enabled or Disabled. Choose All to ignore this filter.
Caching	Choose Enabled or Disabled. Choose All to ignore this filter.

Click **Search** to find all services using the set criteria or click View All to view all the services.

51.5.2 Viewing and Editing Operational Settings

You can view results of the search for business service in the Summary of Business Services table.

The Summary of Business Services view displays State, Monitoring, SLA Alerts, Message Tracing, Name, Path, and Type. For more information about the fields displayed in the summary of business services, see Section 51.3.2, "Viewing and Editing Operational Settings."

Use this table to enable or disable services, monitoring, SLA alerts, and message tracing. To enable or disable, select the check box in the appropriate field and click Update. The runtime effects of these settings also depend on corresponding settings at the global level. For more information about Global Settings, see Section 46.7.2, "How to Configure the Operational Settings at the Global Level." You can update the information for one or more business services concurrently using this table.

Click **Reset** to discard any changes in the summary table and refresh the page with currently stored settings.

> **Note:** You can edit the operational settings for a business service depending on the privileges of your role. For more information about roles, see Chapter 45, "Roles in Oracle Service Bus."

51.6 Managing Operational Settings for Split-Joins

You can search for Split-Joins in the Oracle Service Bus Administration Console using filters in Smart Search. This section describes finding, viewing, and editing Split-Joins.

51.6.1 Finding Split-Joins Using Search Filters

Table 51–6 describes the filters you can use to customize your search for Split-Joins.

Table 51–6   Using Advanced filters to Search for Split-Joins

Filters	Usage
Logs	Choose Enabled or Disabled. Choose All to ignore this filter.
Execution Tracing	Choose Enabled or Disabled. Choose All to ignore this filter.

Click **Search** to find all services using the set criteria or click View All to view all the services.

51.6.2 Viewing and Editing Operational Settings

You can view results of the search for Split-Joins in the Summary of Split-Joins table.

The Summary of Split-Joins view displays Name, Path, Type, Logs, and Execution Tracing. Use this table to view the Split-Join configuration details, modify operational settings, and enable/disable logs and execution tracing. To enable or disable, select the check box in the appropriate field and click Update. The runtime effects of these settings also depend on corresponding settings at the global level. For more information about Global Settings, see Section 46.7.2, "How to Configure the Operational Settings at the Global Level." You can update the information for one or more Split-Joins concurrently using this table.

Click **Reset** to discard any changes in the summary table and refresh the page with currently stored settings.

> **Note:**   You can edit the operational settings for a Split-Join depending on the privileges of your role. For more information about roles, see Chapter 45, "Roles in Oracle Service Bus."

51.7 Managing Operational Settings for Alert Destinations

You can search for alert destinations in the Oracle Service Bus Administration Console using advanced filters in Smart Search. You can view and delete alert destinations using smart search.

51.7.1 Finding Alert Destinations using Search Filters

Table 51–7 describes the filters you can use to customize your search for alert destinations.

Table 51–7   Using Advanced Filters to Search for Alert destinations

Filter	Usage
Target	You can choose from one of the following options: ■ SNMP Trap ■ Reporting ■ Email ■ JMS Only alert destinations with at least one of the selected targets are displayed. By default Target filter is not applied.

Table 51–7   (Cont.) Using Advanced Filters to Search for Alert destinations

Filter	Usage
Search Pattern	The system uses the string to search all the Description fields of the Alert Destinations, as well as the specific detailed fields of the email and JMS destinations. If the string appears in any of the Alert Destination fields, the Alert Destinations matching the search criteria are displayed.
	Click **Search** to find all alert destinations using the set criteria or click View All to view all the alert destinations. For more information about alert destinations, see Section 46.6.7, "What are Alert Destinations?."

51.7.2 Viewing and Deleting Alert Destinations

You can view results of the search for alert destinations in the Summary of Alert Destinations table.

Table 51–8 describes the information is provided in the Summary of Alert Destination table.

Table 51–8   Summary of Alert Destination

Column Name	Description
Name	This displays the names of the alert destinations that satisfy the search criteria.
Path	This displays the location of the resource in the Oracle Service Bus domain.
Options	This displays the actions that can be performed on the alert destination. You can delete an alert destination from this field.

You can delete one or more alert destination concurrently using this table.To delete an alert destination click the check box associated with the alert destination and click Delete.

Notes: To create an alert destination click the path to view the corresponding project folder. Select Alert Destination in Create Resource to create a new alert destination.

To reconfigure or view the details of an alert destination click the alert destination to go to the Alert destination configuration page.

51.8 Managing Operational Settings for SLA Alert Rules

You can search for SLA alert rules in the Oracle Service Bus Administration Console using additional filters in Smart Search. This section describes finding, viewing, and configuring alert rules using smart search.

51.8.1 Finding SLA Alert Rules Using Search Filters

Table 51–9 describes the filters you can use to customize your search for SLA ALert Rules:

Table 51–9   Using Advanced Filters to Search for SLA Alert Rules

Filters	Usage
Parent Service	You can base the search for SLA alert rules on the proxy service or business service associated with the SLA alert rule. Use this filter to override the path specified in the basic search.
Service Type	You can specify the type of the parent service criterion. The parent service can be one of the following options: ■ All Services: The parent service is either a proxy service or a business service. ■ Proxy Services ■ Business Service **Note**: When you specify the value for Service Type, Parent Service is reset.
Rule State	You can customize the search based on the state of the SLA alert rule. The SLA alert rule can be in either of the following states: ■ Enabled: Use this to search for all the SLA alert rules that are enabled. ■ Disabled: Use this to search for all the SLA alert rules that are disabled. Choose All to ignore this filter.
Severity	You can customize the search based on the severity of the SLA alerts. To do so, set the level of severity in the Severity field.

Click **Search** to find all services using the set criteria or click View All to view all the services.

> **Note:**   Select the **or above** check box to restrict your search to the specified severity level or above (listed from the most inclusive to the most restrictive level): Normal, Warning, Minor, Major, Critical, and Fatal. For example, to search for alert rules with severity levels equal to Major, Critical, and Fatal set severity equal to Major and click the check box associated with **or above**.

51.8.2 Viewing and Configuring SLA Alert Rules

You can view results of the search for SLA alert rules in the SLA Alert Rules table. You can update the information for one or more alert rule concurrently using this table. To enable or disable the alert rule select the associated check box and click Update.

Click **Reset** to discard any changes in the summary table and refresh the page with currently stored settings.

Table 51–10 describes information displayed in the SLA Alert Rules Summary table:

Table 51–10   SLA Alert Summary Table

Property	Description
Name	The name of the alert rule. Click the name to go to the View Alert Rule Details page. For more information, see Section 46.8.2, "How to Lookup or Edit Existing Alert Rules."
SLA State	The status of the alert rule: Enabled or Disabled.

Table 51–10   (Cont.) SLA Alert Summary Table

Property	Description
Description	This field is hidden by default. A description of the alert rule.
Service Name	The name of the parent service. The name is a link to the Operational Settings page.
Path	The project associated with the parent service of the alert rule. If the parent service of the alert rule resides in a project folder, this folder is also listed. The path is displayed in the format: project-name/root-folder/ . . ./parent-folder The path is a link to the corresponding path in the Project Explorer.
Severity	The severity of the alert that is triggered by this rule: Normal, Warning, Minor, Major, Critical, or Fatal.
Aggr Interval	The aggregation interval in terms of hours and minutes.
Expiration Date	The date when this alert rule is no longer in effect.
Stop Processing	Displays Yes or No. This field is hidden by default.
Frequency	The frequency of this alert: ■ Every Time ■ Notify Once

> **Note:**   You can enable or disable an alert rule depending on the privileges of your role. For more information about roles, see Chapter 45, "Roles in Oracle Service Bus." For more information about enabling and disabling alert rules, see Section 46.7.1, "How to Configure the Operational Settings for a Service."

52

User Preferences

This chapter describes how to customize your home page and other settings in the Oracle Service Bus Administration Console.

The home page is the page you wish to access first when you log in to the Oracle Service Bus Administration Console. To set your preferences for the home page, click User Preferences in the Configuration module of the Operations navigator bar. Your user preferences are preserved across different sessions.

Table 52–1 describes the preferences you can set from the User Preferences page.

Table 52–1   User Preferences

User Preference	Description
Home Page	You can set the home page to one of the following options: ■ Operations: Choose this option to set one of the following pages as your home page: Dashboard Smart Search Global Settings User Preferences Message Reports
Home Page	■ Resource Browser: Choose this option to set one of the following pages as your home page: Proxy Services Business Services Split-Joins WSDLs XML Schemas WS-Policies JCA Bindings XQueries XSLTs MFLs Service Account Service Key Providers Jars Alert Destinations

Table 52–1 (Cont.) User Preferences

User Preference	Description
Home Page	Project Explorer: Choose this option to view the list of projects in the Oracle Service Bus Administration Console.
Home Page	Security Configuration: Choose this option to set one of the following pages as your home page: Users Groups Roles
Home Page	System Administration: Choose this option to set one of the following pages as your home page: Import Resources Export Resources UDDI Registries Import From UDDI Auto Import Status Publish to UDDI Auto-Publish Status JNDI Providers SMTP Servers Proxy Servers Find and Replace Create Customization File Execute Customization File **Note**: The pages that can be set for these options and the default setting for these options vary depending on the role assigned to the user.
Display Search Filters	Select Yes to display all search filters in the Oracle Service Bus Administration Console. The default setting is No.
Display Stage Comments	Set this to display stage annotations or not. The default setting is No.
Display Resource Metadata	Set this to display resource metadata or not. The default setting is Yes.
Dashboard Refresh Rate	Use this to set the refresh rate of the dashboard. You can set this to No refresh, 1, 2, 3, 4, 5, 10, 20, 30, or 60 minutes. The default value for this is No refresh. For more information about Refresh Rate of the Dashboard, see Section 46.2.1, "The Refresh Rate of Monitoring Data."
Alert History Duration	You can set the alert history duration to 30 minutes, 1, 2, 3, or 6 hours. The default value is 30 minutes. For more information about Alert History, see Section 46.9.3.1, "Viewing the Alert History for SLA Alerts" and Section 46.9.4.1, "Viewing the Alert History for Pipeline Alerts."

Part IV

Appendix

This part contains miscellaneous administration information and reference material. Chapters include:

- Appendix A, "Error Codes"
- Appendix B, "Debugging Oracle Service Bus"
- Appendix C, "Oracle Service Bus APIs"
- Appendix D, "JMX Monitoring API"
- Appendix E, "SNMP Components"
- Appendix F, "Monitoring Statistics in Oracle Service Bus"
- Appendix G, "Auditing your Oracle Service Bus System"
- Appendix H, "XQuery-SQL Mapping Reference"
- Appendix I, "Interoperability with WSRP"

A
Error Codes

This chapter describes the Oracle Service Bus errors that can occur when a proxy service is being executed. The error codes associated with these errors surface inside the element of the `fault` context variable. You can access the value using the following XQuery statement:

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

Errors are accompanied by details specific to the error inside the `fault` element.

Table A–1 through Table A–5 list Oracle Service Bus error codes. For error schema examples, see Section A.1, "Error Details Schemas."

Table A–1   Oracle Service Bus Error Codes - Transport Runtime Errors (BEA-380000 to BEA-380099)

Error Code	Error Message
BEA-380000	General Transport error
	XML Details: "An Error Response Was Received" (when occurring in publish action)

Table A–2   Oracle Service Bus Error Codes - Message Flow Runtime Errors (382000...382499)

Error Code	Error Message
BEA-382000	General runtime error
BEA-382030	General parse failure from binding layer (for example, message to XML service is not XML)
BEA-382031	WS-I compliance failure
BEA-382032	Message must be a soap:Envelope
	XML Details: "A Non-SOAP or Invalid Envelope Was Received"
BEA-382033	A soap:Envelope must contain a soap:Body
BEA-382040	Failed to assign value to context variable "{0}". Value must be an instance of {1}
BEA-382041	Failed to assign value to context variable "{0}". Variable is read-only.
BEA-382042	Failed to assign value to context variable "{0}". {1}
BEA-382043	Failed to update the value of context variable "{0}": {1}
BEA-382045	Failed to initialize the value of context variable "{0}": {1}

Table A–2   (Cont.) Oracle Service Bus Error Codes - Message Flow Runtime Errors (382000...382499)

Error Code	Error Message
BEA-382046	Failed to marshall the value of context variable "{0}" to XML: {1}
BEA-382100	General binding error while processing inbound request
BEA-382101	General binding error while preparing inbound response
BEA-382102	General binding error while preparing outbound request
BEA-382103	General binding error while processing outbound response
BEA-382104	Failed to prepare request metadata for service {0}
BEA-382105	Failed to prepare response metadata for service {0}
BEA-382150	Failed to dispatch request to service {0}
BEA-382151	Cannot dispatch to unknown service: {0}

Table A–3   Oracle Service Bus Error Codes - Action Errors (382500...382999)

Error Code	Error Message
BEA-382500	ALSBConfigurationMBean service callout action received SOAP Fault response. XML details: "A SOAP Fault Response was Received"
BEA-382501	Oracle Service Bus service callout action received an unrecognized response. XML details: "An Unrecognized Response was Received"
BEA-382502	Oracle Service Bus service callout has received an error response from the server XML details: "An Unknown Error Response Was Received"
BEA-382505	Oracle Service Bus Validate action validation failed. XML details: "Validation Failed"
BEA-382510	Oracle Service Bus Assign action failed updating variable "{0}": {1}
BEA-382511	Oracle Service Bus Delete action failed updating variable "{0}": {1}
BEA-382512	Oracle Service Bus Insert action failed updating variable "{0}": {1}
BEA-382513	Oracle Service Bus Replace action failed updating variable "{0}": {1}
BEA-382514	Oracle Service Bus Rename action failed updating variable "{0}": {1}
BEA-382515	Callout to java method "{0}" resulted in exception: {1}
BEA-382516	Failed to evaluate expression for callout to java method "{0}". Argument index: {1}, exception: {2}
BEA-382517	Failed to assign the result of Java callout to variable. Method: {0}, variable: {1}, exception: {2}
BEA-382518	Security exception while calling to java method "{0}". Service account: {1}. {2}

Table A–3 (Cont.) Oracle Service Bus Error Codes - Action Errors (382500...382999)

Error Code	Error Message
BEA-382600	Oracle Service Bus publish action received an error response: {0}

Table A–4 Oracle Service Bus Error Codes - Security Errors (386000...386999)

Error Code	Error Message
BEA-386000	General security error. Possible reasons include the following: ■ Custom token or custom user name/password authentication is configured but authentication already took place through WS-Security.
BEA-386100	Authentication failed. Possible reasons include the following: ■ The XPath is configured to act on the body or header but there is no such message part (the binding layer returns null). ■ The XPath engine throws an exception when evaluating the XPath expression. ■ The result of the user name XPath or password XPath is a node that is not of type TEXT or ATTR. ■ The XPath returns more than one node. ■ A user name/password was retrieved but authentication failed. ■ A custom token was retrieved but identity assertion failed (the CSS identity assertion service throws LoginException or any other exception).
BEA-386101	Missing authentication token. Possible reasons include the following: ■ The user name XPath or password XPath returns an empty result or empty string. ■ The custom token XPath returns an empty result, null attribute value, or null text-value.
BEA-386102	Message-level authorization denied. Possible reasons include the following: ■ The message level access control policy denies access to the proxy service.
BEA-386103	The proxy service operation selection algorithm cannot determine the operation name from the request or returns an invalid operation (one which is not in the WSDL or null). Possible reasons include the following: ■ An error occurs while computing the operation. ■ The operation selection algorithm returns null. ■ The operation selection algorithm returns an operation that is not of the of the operations declared by the WSDL.
BEA-386200	General web service security error
BEA-386201	A web service security fault occurred[<error-code from WSS fault>][<fault-string from WSS-fault>] XML details: "A WS-Security Fault Occurred"

Table A–4   (Cont.) Oracle Service Bus Error Codes - Security Errors (386000...386999)

Error Code	Error Message
BEA-386400	General outbound web service security error
BEA-386401	Failed to convert outbound message to SOAP
BEA-386402	Cannot determine the outbound operation
BEA-386420	A web service security error occurred while producing security header
BEA-386440	A web service security error occurred while processing the security header
BEA-386460	Web Service Security policy validation error

Table A–5   Oracle Service Bus Error Codes - UDDI Errors (394500...394999)

Error Code	Error Message
BEA-394500	An error was encountered while importing a resource
BEA-394501	An error was encountered while querying the UDDI registry for business entities
BEA-394502	An error was encountered while initializing the UDDI service
BEA-394503	An error was encountered while initializing the inquiry port
BEA-394504	An error was encountered while publishing this service
BEA-394505	Failed to connect to the UDDI registry
BEA-394506	An error was encountered while querying the UDDI registry for business services
BEA-394507	The registry name contains characters that are not allowed
BEA-394508	The publish URL was missing the UDDI registry configuration
BEA-394509	The Service Account configured is not valid
BEA-394510	A resource could not imported while importing a service
BEA-394511	A validation exception occurred while creating a service definition while importing a service
BEA-394512	A generic error was encountered while importing a resource
BEA-394513	An internal error was encountered while importing a resource
BEA-394514	Failed to create an FTP endpoint configuration when importing a service
BEA-394515	Failed to create an Email endpoint configuration when importing a service
BEA-394516	A possible syntax error was encountered in the access point URL when importing a service
BEA-394517	An error was encountered when parsing the InstanceParms while importing a service

Table A–5 (Cont.) Oracle Service Bus Error Codes - UDDI Errors (394500...394999)

Error Code	Error Message
BEA-394518	Failed to create an file endpoint configuration when importing a service

A.1 Error Details Schemas

Below are schema definitions for XML snippets that accompany the error codes listed in Table A–1. These XML snippets appear inside the fault context variable under the `<ctx:details>` element. These schemas do not all come from a single namespace. Also, while a given error code typically implies a particular XML snippet, some XML snippets are used by several error codes.

```
<xs:schema targetNamespace="http://www.oracle.com/wli/sb/stages/transform/config"
        xmlns:tns="http://www.oracle.com/wli/sb/stages/transform/config"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:ref="http://www.oracle.com/wli/sb/reference"
        xmlns:sdk="http://www.oracle.com/wli/sb/stages/config"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
```

A SOAP Fault Response was Received

```
<xs:element name="ReceivedFaultDetail" type="tns:ReceivedFaultDetail"/>
<xs:complexType name="ReceivedFaultDetail">
    <xs:sequence>
        <xs:element name="faultcode" type="xs:QName"/>
        <xs:element name="faultstring" type="xs:string"/>
        <xs:element name="detail" minOccurs="0" >
            <xs:complexType mixed="true" >
                <xs:sequence>
                    <xs:any namespace="##any" minOccurs="0"
                            maxOccurs="unbounded" processContents="lax" />
                </xs:sequence>
                <xs:anyAttribute namespace="##any" processContents="lax" />
            </xs:complexType>
        </xs:element>

        <!-- HTTP-specific status/response code -->
        <xs:element name="http-response-code" type="xs:int" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

An Unrecognized Response was Received

```
<xs:element name="UnrecognizedResponseDetail"
            type="tns:UnrecognizedResponseDetail"/>
<xs:complexType name="UnrecognizedResponseDetail">
    <xs:sequence>
        <xs:element name="detail" minOccurs="0" type="xs:string" />

        <!-- HTTP-specific status/response code -->
        <xs:element name="http-response-code" type="xs:int" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

An Unknown Error Response Was Received

```
<xs:element name="ErrorResponseDetail" type="tns:ErrorResponseDetail"/>
<xs:complexType name="ErrorResponseDetail">
```

```
                   <xs:sequence>
                        <xs:element name="detail" minOccurs="0" type="xs:string" />

                        <!-- HTTP-specific status/response code -->
                        <xs:element name="http-response-code" type="xs:int" minOccurs="0"/>
                   </xs:sequence>
              </xs:complexType>
```

Validation Failed

```
      <xs:complexType name="ValidationFailureDetail">
           <xs:sequence minOccurs = "0" maxOccurs="unbounded">
                <xs:element name="message" type="xs:string"/>
                <xs:element name="xmlLocation" type="xs:anyType"/>
           </xs:sequence>
      </xs:complexType>
</xs:schema>
<schema targetNamespace="http://www.oracle.com/wli/sb/errors"
        xmlns:err="http://www.oracle.com/wli/sb/errors"
        xmlns:tc="http://www.oracle.com/wli/sb/transports"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
```

A Non-SOAP or Invalid Envelope Was Received

```
      <element name="InvalidEnvelope">
        <complexType>
           <sequence>
                <element name="localpart" type="NCName"/>
                <element name="namespace" type="anyURI" minOccurs="0"/>
           </sequence>
        </complexType>
      </element>
```

A WS-Security Fault Occurred

```
      <element name="WebServiceSecurityFault">
        <complexType>
           <sequence>
                <element name="faultcode" type="QName"/>
                <element name="faultstring" type="string"/>
                <element name="detail" minOccurs="0">
                     <complexType mixed="true">
                        <sequence>
                             <any namespace="##any" minOccurs="0"
                                  maxOccurs="unbounded" processContents="lax"/>
                        </sequence>
                        <anyAttribute namespace="##any" processContents="lax"/>
                     </complexType>
                </element>
           </sequence>
        </complexType>
      </element>
```

An Error Response Was Received

```
      <element name="ErrorResponseDetail" type="err:ErrorResponseDetail"/>
      <complexType name="ErrorResponseDetail">
        <sequence>
             <!-- Response metadata -->
             <element name="response-metadata" type="tc:ResponseMetaDataXML"
```

```
                        minOccurs="0" />
            </sequence>
        </complexType>
    </schema>
```

B

Debugging Oracle Service Bus

This chapter describes how to enable debugging for different modules in Oracle Service Bus.

This chapter includes the following sections:

- Section B.1, "Debugging Oracle Service Bus"
- Section B.2, "Debugging Proxy Services and Split-Joins in Eclipse"

B.1 Debugging Oracle Service Bus

You can enable and disable debugging by modifying the corresponding entries in the following debug XML files, which are located in the root directory of your Oracle Service Bus domain:

- `alsbdebug.xml`—Contains Oracle Service Bus related debug flags
- `configfwkdebug.xml`—Contains configuration related debug flags

If the XML files are not in the root directory or if they have been deleted, they are created again without any contents when the server starts. Example B–1 provides an example of the contents of the `alsbdebug.xml` file with debugging disabled for all modules (all entries set to `false`).

Example B–1   alsbdebug.xml File

```
<java:sb-debug-logger xmlns:java="java:com.bea.wli.debug">
  <java:alsb-stages-transform-runtime-debug>true</java:alsb-stages-transform-runtime-debug>
  <java:alsb-alert-manager-debug>false</java:alsb-alert-manager-debug>
  <java:alsb-credential-debug>false</java:alsb-credential-debug>
  <java:alsb-jms-reporting-provider-debug>false</java:alsb-jms-reporting-provider-debug>
  <java:alsb-management-credential-debug>false</java:alsb-management-credential-debug>
  <java:alsb-management-dashboard-debug>false</java:alsb-management-dashboard-debug>
  <java:alsb-management-debug>false</java:alsb-management-debug>
  <java:alsb-management-user-mgt-debug>false</java:alsb-management-user-mgt-debug>
  <java:alsb-module-debug>true</java:alsb-module-debug>
  <java:alsb-monitoring-aggregator-debug>false</java:alsb-monitoring-aggregator-debug>
  <java:alsb-monitoring-debug>false</java:alsb-monitoring-debug>
  <java:alsb-pipeline-debug>true</java:alsb-pipeline-debug>
  <java:alsb-security-wss-debug>true</java:alsb-security-wss-debug>
  <java:alsb-service-account-manager-debug>false</java:alsb-service-account-manager-debug>
  <java:alsb-service-provider-manager-debug>false</java:alsb-service-provider-manager-debug>
  <java:alsb-service-repository-debug>false</java:alsb-service-repository-debug>
  <java:alsb-service-security-manager-debug>false</java:alsb-service-security-manager-debug>
  <java:alsb-service-validation-debug>false</java:alsb-service-validation-debug>
  <java:alsb-test-console-debug>false</java:alsb-test-console-debug>
```

```
  <java:alsb-transports-debug>true</java:alsb-transports-debug>
  <java:alsb-uddi-debug>true</java:alsb-uddi-debug>
  <java:alsb-wsdl-repository-debug>false</java:alsb-wsdl-repository-debug>
  <java:alsb-wspolicy-repository-debug>true</java:alsb-wspolicy-repository-debug>
  <java:alsb-security-encryption-debug>false</java:alsb-security-encryption-debug>
  <java:alsb-security-module-debug>false</java:alsb-security-module-debug>
  <java:alsb-sources-debug>false</java:alsb-sources-debug>
  <java:alsb-proxy-server-manager-debug>true</java:alsb-proxy-server-manager-debug>
  <java:alsb-custom-resource-debug>false</java:alsb-custom-resource-debug>
  <java:alsb-mqconnection-debug>false</java:alsb-mqconnection-debug>
  <java:alsb-throttling-debug>true</java:alsb-throttling-debug>
  <java:alsb-flow-resource-debug>false</java:alsb-flow-resource-debug>
  <java:alsb-flow-transport-debug>false</java:alsb-flow-transport-debug>
  <java:alsb-flow-deployment-debug>false</java:alsb-flow-deployment-debug>
  <java:alsb-soa-codec-debug>true</java:alsb-soa-codec-debug>
  <java:alsb-message-tracing-debug>true</java:alsb-message-tracing-debug>
  <java:alsb-result-caching-debug>true</java:alsb-result-caching-debug>
  <java:alsb-console-debug>false</java:alsb-console-debug>
  <java:alsb-bpel-debug>false</java:alsb-bpel-debug> <!-- Split-Join processing details -->
  <java:alsb-jca-framework-adapter-debug>true</java:alsb-jca-framework-adapter-debug>
</java:sb-debug-logger>
```

Example B–2 provides an example of the contents of the `configfwkdebug.xml` file.

Example B–2   *configfwkdebug.xml File*

```
<java:config-fwk-debug-logger xmlns:java="java:com.bea.wli.config.debug">
  <n1:Name xmlns:n1="java:weblogic.diagnostics.debug">configfwkdebug</n1:Name>
  <java:config-fwk-debug>false</java:config-fwk-debug>
  <java:config-fwk-transaction-debug>false</java:config-fwk-transaction-debug>
  <java:config-fwk-deployment-debug>false</java:config-fwk-deployment-debug>
  <java:config-fwk-component-debug>false</java:config-fwk-component-debug>
  <java:config-fwk-security-debug>false</java:config-fwk-security-debug>
</java:config-fwk-debug-logger>
```

Although debugging should be disabled during normal Oracle Service Bus operation, you may find it helpful to turn on certain debug flags while you are developing your solution and experimenting with it for the first time. For example, you may want to turn on the alert debugging flag when you are developing alerts and would like to investigate how the alert engine works.

Some of the available Oracle Service Bus debug flags are shown in Table B–1.

Table B–1   *Oracle Service Bus Debug Flags*

Debug Flag	Action
alsb-stages-transform-runtime-debug	Provides information on transformation related actions.
alsb-alert-manager-debug	Prints an evaluation of alerts.
alsb-jms-reporting-provider-debug	Provides information on the out of the box, JMS-based reporting provider.
alsb-management-debug	Provides information on user and group management in the console.
alsb-monitoring-debug	Provides information on the statistics system.
alsb-pipeline-debug	Provides information on errors that are generated within the pipeline.

Table B–1   (Cont.) Oracle Service Bus Debug Flags

Debug Flag	Action
alsb-service-repository-debug	Provides information on various service related configuration operations.
alsb-service-security-manager-debug	Provides information on access control.
alsb-transports-debug	Provides transport related debug information, including transport headers, which is printed per-message.
	Note: For the JCA transport, framework, and adapters, use alsb-jca-framework-adapter-debug.
alsb-jca-framework-adapter-debug	Provides JCA transport, framework, and adapters debug information.
alsb-wsdl-repository-debug	Provides information on WSDL related configuration operation.
alsb-wspolicy-repository-debug	Provides information on WS policy.
alsb-custom-resource-debug	Provides information on custom resources.
alsb-mqconnection-debug	Provides information on the MQ connection resource.
alsb-throttling-debug	Provides information on the throttling feature.
alsb-result-caching-debug	Provides information on business service result caching.

Table B–2 lists the available configuration framework debug flags.

Table B–2   Configuration Framework Debug Flags

Debug Flag	Action
config-fwk-debug	Provides information on general aspects of Oracle Service Bus configuration.
config-fwk-transaction-debug	Provides low level debug information about changes made to in-memory data structures and files. This debug flag also generates server startup recovery logs.
config-fwk-deployment-debug	Provides debug information on session creation, activation, and distribution of configuration in a cluster.
config-fwk-component-debug	Provides low level debug information about create, update, delete, and import operations.
config-fwk-security-debug	Provides debug information on encryption and decryption during importing and exporting.

All other debug flags are self explanatory.

For all flags, debug information is logged to the server log at *DOMAIN_ DIRECTORY*/servers/*servername*/logs/*servername*.log.

B.2 Debugging Proxy Services and Split-Joins in Eclipse

See "Using the Oracle Service Bus Debugger" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

C

Oracle Service Bus APIs

This chapter describes Oracle Service Bus APIs for resource update and customization, management, monitoring, and deployment.

Oracle Service Bus exposes APIs to allow customizing resources and to provide external access to monitoring data and deployment.

- Section C.1, "Resource Update and Customization"

- Section C.2, "Management and Monitoring"

- Section C.3, "Deployment"

Javadoc for the Oracle Service Bus APIs is provided in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

C.1 Resource Update and Customization

A number of APIs are exposed to allow customization of service definitions, WSDLs, schemas, XQueries, and other design-time resources through programmatic interfaces. The supporting APIs allow loading ZIP files containing resources, in addition to moving, renaming, cloning, or deleting resources, folders, and projects. A typical use case is one in which you have a prototypical proxy service from which you make a number of copies—each copy can be modified programmatically.

Numerous customization options can be applied during deployment. For example, environment variables allow you to preserve or tailor settings when moving from one environment to another.

The available APIs include:

- ProxyServiceConfigurationMBean—Enable and disable SLA alerts and pipeline alerts for proxy services

- BusinessServiceConfigurationMBean

  - Enable and disable SLA alerts

  - Synchronize business services imported from UDDI registries

  - Detach business services from a UDDI registry

- ALSBConfigurationMBean Interface—APIs for managing resources in an Oracle Service Bus domain, including:

  - Query, export, and import resources

  - Obtain validation errors

  - Get and set environment values

- – Modify references inside resources to new references

- – Move, rename, clone, and delete resources

- ■ Customization Class

  - – Find and replace environment values

  - – Assign environment values

  - – Map references found in resources to other references

C.2 Management and Monitoring

The JMX Monitoring API in Oracle Service Bus provides external access to monitoring data. Java Management Extensions (JMX) technology was used for the implementation. Oracle Service Bus resources within a domain use JMX Managed Beans (MBeans) to expose their management functions. An MBean is a concrete Java class that is developed according to JMX specifications.

For information, see Appendix D, "JMX Monitoring API."

C.3 Deployment

You can use the Oracle Service Bus MBeans in Java programs and WLST scripts to automate promotion of Oracle Service Bus configurations from development environments through testing, staging, and finally to production environments.

Numerous customization options can be applied during deployment. For example, an extended list of environment variables lets you preserve or tailor settings when moving from one environment to another.

For information, see "Using the Deployment APIs" in the *Oracle Fusion Middleware Deployment Guide for Oracle Service Bus*.

D

JMX Monitoring API

This chapter describes the Java Management Extensions (JMX) Monitoring API in Oracle Service Bus (OSB), which provides external access to OSB monitoring data.

The primary purpose of the JMX Monitoring API is to provide efficient, lower-level APIs supporting bulk operations. It does this using JMX as a transport. This API is not a high-level API compatible with JMX-based tools. However, if you are developing client software, you may want to develop high-level JMX APIs that support JMX-based tooling.

D.1 Description

The JMX monitoring API makes use of JMX as a transport only. It exposes a public MBean to provide all the required operations to get monitoring data (statistical information) for any monitored service and its components. It also exposes a set of public POJO objects required to carry out operations provided by the MBean. There is no need for third-party client software to know the intricacies of the hierarchy inherent in the statistical information stored in the Oracle Service Bus monitoring system.

Using these APIs, customers can integrate their monitoring/management systems with Oracle Service Bus to do the following:

- Identify services enabled for monitoring.

- Get detailed statistical information for a specific service, for its components, or for both.

- Reset statistics accumulated since the last reset.

D.2 Concepts

The public JMX APIs are modeled by a single instance of `ServiceDomainMBean`, which has operations to check for monitored services and retrieve data from them. A public set of POJOs provide additional objects and methods that, along with `ServiceDomainMbean`, provide a complete API for monitoring statistics.

The following sections provide brief descriptions of the POJOs and MBean. The Java API Reference provides detailed descriptions. There is also a detailed description of statistics that are reported for resources.

Be sure to read the important notes at the end of this chapter.

D.2.1 Public POJO Objects

The following POJO objects are exposed as part this API.

D.2.1.1 ResourceType

This object represents all types of resources that are enabled for service monitoring. There are four `enum` constants representing types: `SERVICE`, `FLOW_COMPONENT`, `URI`, and `WEBSERVICE_OPERATION`.

See `com.bea.wli.monitoring.ResourceType` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.1.2 ServiceResourceStatistic

This object represents all business and proxy service resource types and the statistics associated with them. There are methods to get statistics for all resources or for a specified one.

See `com.bea.wli.monitoring.ServiceResourceStatistic` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.1.3 ResourceStatistic

This object represents a resource for which statistics collection is supported. There are methods to get the name of the resource, the type, and the statistics.

See `com.bea.wli.monitoring.ResourceStatistic` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.1.4 StatisticValue

This object represents a statistic value for a resource. The monitoring system currently supports the following types of statistic values, both nested classes:

- CountStatistic

- IntervalStatistic

- StatusStatistic

`StatisticValue` is an abstract class so that concrete objects representing count and interval statistic values can be derived from it. It includes `getName()` and `getType()` methods.

See `com.bea.wli.monitoring.StatisticValue` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.1.5 StatisticType

This object represents predefined types of statistics. There are three `enum` types: `STATUS`, `COUNT`, and `INTERVAL`.

See `com.bea.wli.monitoring.StatisticValue` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.2 ServiceDomainMBean

This is the only MBean exposed as part of the public JMX API. It provides methods to find monitored service and get and reset statistics.

See `com.bea.wli.monitoring.ServiceDomainMBean` in the *Oracle Fusion Middleware Java API Reference for Oracle Service Bus*.

D.2.3 Statistics Details

The following sections provide detailed information about statistics reported for each resource type.

D.2.3.1 Statistics details for Resource Type - SERVICE

A service is an inbound or outbound endpoint that is configured within Oracle Service Bus. It may have an associated WSDL, security settings, and so on.

The following statistics are reported for this resource type.

Table D–1   SERVICE Statistics

Statistic Name	Type
message-count	count
error-count	count
failover-count	count
wss-error	count
response-time	interval
validation-errors	count
failure-rate	count
success-rate	count
sla-severity-warning	count
sla-severity-major	count
sla-severity-minor	count
sla-severity-normal	count
sla-severity-fatal	count
sla-severity-critical	count
sla-severity-all	count
pipeline-severity-warning	count
pipeline-severity-major	count
pipeline-severity-minor	count
pipeline-severity-normal	count
pipeline-severity-fatal	count
pipeline-severity-critical	count
pipeline-severity-all	count
throttling-time	interval
uri-offline-count	count

Table D–1   (Cont.) SERVICE Statistics

Statistic Name	Type
hit-count	count

The statistics `sla-severity-warning`, `sla-severity-major`, `sla-severity-minor`, `sla-severity-normal`, `sla-severity-fatal`, `sla-severity-critical`, and `sla-severity-all` are collected for both proxy services and business services.

The statistics `pipeline-severity-warning`, `pipeline-severity-major`, `pipeline-severity-minor`, `pipeline-severity-normal`, `pipeline-severity-fatal`, `pipeline-severity-critical`, and `pipeline-severity-all` are collected only for proxy services.

> **Note:**   Statistic "wss-error" provides Web Service security violations counts. It is applicable to both business and proxy services.
>
> Statistic "validation errors" is only applicable to proxy service; it is not returned for a business service.
>
> Statistic "failover-count" is only applicable to business service; it is not returned for a proxy service.
>
> When the statistics of a Managed Server are retrieved from a cluster domain using the ServiceDomainMBean the statistics for proxy services will not contain `sla-severity-normal`, `sla-severity-minor`, `sla-severity-major`, `sla-severity-warning`, `sla-severity-critical`, `sla-severity-fatal`, `sla-severity-all`.

D.2.3.2 Statistics details for Resource Type–FLOW_COMPONENT

Statistics are collected for the following two types of components that can be present in the flow definition of a proxy service.

- Pipeline-pair node
- Route node

Pipelines are one-way processing paths consisting of stages that are executed sequentially against the current message. Stages are used to perform activities such as transformation, logging and publishing.

There are three categories of pipelines: request, response, and error.

The pipeline-pair node ties together a single request and a single response pipeline into one top-level element.

A routing node consists of a set of routes. A route identifies a target service and includes some additional configuration options that determines how the message will be packaged and sent to that service. A routing node will result in at most one route being selected as part of request processing.

The following statistics are reported for this resource type.

Table D–2   FLOW_COMPONENT Statistics

Statistic Name	Type
elapse-time	interval
message-count	count

Table D–2   (Cont.) FLOW_COMPONENT Statistics

Statistic Name	Type
error-count	count

> **Note:**   Statistics for pipeline and route nodes are returned as statistics for flow components. `enum` value `ResourceType.FLOW_COMPONENT` represents both pipeline and route nodes. Thus there is no way for a client to check if the returned flow component is a pipeline or route node. The name of the flow component, however, may suggest the type.

D.2.3.3 Statistics details for Resource Type – WEBSERVICE_OPERATION

This resource type provides statistical information pertaining to WSDL operations. Statistics are reported for each defined operation.

The following statistics are reported.

Table D–3   WEBSERVICE_OPERATION Statistics

Statistic Name	Type
elapsed-time	interval
message-count	count
error-count	count

D.2.3.4 Statistics details for Resource Type – URI

This resource type provides statistical information pertaining to endpoint URI for a business service. Statistics are reported for each defined Endpoint URI. The following statistics are reported.

Table D–4   Statistics for Endpoint URI

Statistic Name	Type
message-count	count
error-count	count
response-time	interval
status	status

D.2.4 Caveats

Be aware of the following:

- A client program will not know about newly added services that have monitoring turned on, or services modified to turn on monitoring, unless it periodically checks for such changes.

- Reset operations should not be performed too frequently. Oracle recommends that reset intervals be greater than 15 minutes.

- If statistics are reset while proxy or business services are running, the following `TransactionConflictException` can occur. This is most likely to occur if statistics are reset at system startup.

```
<BEA-382016> <Failed to instantiate router for service...>
```

■ Oracle strongly discourages using this API in a concurrent manner with more than one thread or process. This is because a reset performed in one thread or process is not visible to another threads or processes. This caveat also applies to resets performed from the Monitoring Dashboard of the Oracle Service Bus Administration Console, as such resets are not visible to this API.

D.2.5 Performance

Performance should be better than or equivalent to that observed in the Monitoring Dashboard of the Oracle Service Bus Administration Console.

D.3 API Usage Example

The sample program in this section demonstrates how to use the JMX Monitoring API.

The following steps describe how statistics can be retrieved for a proxy service that is enabled for monitoring.

1. Get `ServiceDomainMBean` from the MBean Server.

2. Get the references for monitored proxy services using the `getMonitoredProxyServiceRefs` operation of the `ServiceDomainMBean`.

3. Identify the references of the desired proxy service from the retrieved references.

4. Get `ServiceResourceStatistics` using the `getProxyServiceStatistics` operation of the `ServiceDomainMBean` of the desired proxy service.

5. Get all `ResourceStatistic` objects using the operations of `ServiceResourceStatistic`.

6. For each retrieved `ResourceStatistic` object, get `StatisticValue` objects using the `getStatistics` operation and print statistical information.

7. Repeat process as necessary.

D.3.1 Sample Program

The following sample program explains how to:

1. Find by type services enabled for monitoring.

2. Set a resource-type filter.

3. Retrieve by type statistics for one or more services.

4. Extract statistics from the `ServiceResourceStatistics` object.

5. Save retrieved statistics in the proper format.

6. Reset statistics for one or more services.

To run this program, include the following JAR files in the classpath:

■ weblogic.jar

■ sb-kernel-api.jar

■ com.bea.common.configfwk_*version*.jar

■ com.bea.core.management.core_*version*.jar

■ com.bea.core.management.jmx_*version*.jar

You may need to reset the default values for `SERVER_NAME`, `HOSTNAME`, `PORT`, `USERNAME`, or `PASSWORD` in the code below for your environment.

> **Note:** If you need to get the Status statistics for each end point of a business service, set the `SERVER_NAME` attribute in a cluster environment. If you are running on a single node, Status statistics are returned even if you do not set this property.

For performance reasons, avoid extracting and resetting statistics for a large number of services too often. See Section D.2.4, "Caveats." See Example D–1 for a sample program.

Example D–1   Sample Program to Retrieve Statistics for a Proxy Service that is Enabled for Monitoring

```
package tests.monitoring;

import com.bea.wli.config.Ref;
import com.bea.wli.monitoring.*;
import weblogic.management.jmx.MBeanServerInvocationHandler;

import javax.management.MBeanServerConnection;
import javax.management.MalformedObjectNameException;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;
import javax.naming.Context;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.lang.reflect.InvocationHandler;
import java.lang.reflect.Method;
import java.lang.reflect.Proxy;
import java.net.MalformedURLException;
import java.util.*;
import java.text.SimpleDateFormat;

/**
   * This class provides sample code showing how to use
   * ServiceDomainMBean.
   * It shows how to:
   * 1. Find business and proxy services enabled for monitoring.
   * 2. Get statistics for one or more business and proxy services.
   * 3. Perform reset operation on one or more business and proxy services.
   * 4. Handle exceptions.
   * It uses a timer to retrieve statistics, save them in a file, and
   * perform resets in a recursive manner.
*/

public class ServiceStatisticsRetriever {
     private ServiceDomainMBean serviceDomainMbean = null;
     private String serverName = null;

     /**
       * Retrieve statistics for all business services being monitored in the
      * domain and reset statistics for the same.
      * @throws Exception
```

```
    */

void getAndResetStatsForAllMonitoredBizServices() throws Exception {
    Ref[] serviceRefs =
        serviceDomainMbean.getMonitoredBusinessServiceRefs();

    // Create a bitwise map for desired resource types.
    int typeFlag = 0;
    typeFlag = typeFlag | ResourceType.SERVICE.value();
    typeFlag = typeFlag | ResourceType.WEBSERVICE_OPERATION.value();
    typeFlag = typeFlag | ResourceType.URI.value();

    HashMap<Ref, ServiceResourceStatistic> resourcesMap = null;
    HashMap<Ref, ServiceResourceStatistic> resourcesMapOnSingle
        Server = null;
    // Get cluster-level statistics.
    try {
        // Get statistics.
        System.out.println("Now trying to get statistics for -" +
            serviceRefs.length + " business services...");
        resourcesMap =
            serviceDomainMbean.getBusinessServiceStatistics
                (serviceRefs, typeFlag, serverName);
        // Reset statistics.
        long resetRequestTime =
            serviceDomainMbean.resetStatistics (serviceRefs);

        // Save retrieved statistics.
        String fileName = "BizStatistics_" +
            new SimpleDateFormat("yyyy_MM_dd_HH_mm").
                format(new Date(System. currentTimeMillis()))
                    + ".txt";
        saveStatisticsToFile(resourcesMap, resetRequestTime,
            fileName);
    }
    catch (IllegalArgumentException iae) {
        System.out.println("===============================\n");
        System.out.println("Encountered IllegalArgumentException...
            Details:");
        System.out.println(iae.getMessage());
        System.out.println("Check if proxy ref was passed OR
            flowComp " +
            "resource was passed OR bitmap is invalid..." +
            "\nIf so correct it and try again!!!");
        System.out.println("=================================\n");
        throw iae;
    }
    catch (DomainMonitoringDisabledException dmde) {
        /** Statistics not available as monitoring is turned off at
            domain level.
         */
        System.out.println("=================================\n");
        System.out.println("Statistics not available as
            monitoring " +
        "is turned off at domain level.");
        System.out.println("============================\n");
        throw dmde;
    }
    catch (MonitoringException me) {
        // Internal problem... May be aggregation server is
```

```
                                crashed...
                System.out.println("===============================\n");
                System.out.println("ERROR: Statistics is not available...
                    " +
                    "Check if aggregation server is crashed...");
                System.out.println("===============================\n");
                throw me;
        }
    }
    /**
     * Retrieve statistics for all proxy services being monitored in the
     * domain and reset statistics for the same.
     * @throws Exception
     */
    void getAndResetStatsForAllMonitoredProxyServices() throws Exception {
        Ref[] serviceRefs =
            serviceDomainMbean.getMonitoredProxyServiceRefs();
        // Create a bitwise map for desired resource types.
        int typeFlag = 0;
        typeFlag = typeFlag | ResourceType.SERVICE.value();
        typeFlag = typeFlag | ResourceType.FLOW_COMPONENT.value();
        typeFlag = typeFlag | ResourceType.WEBSERVICE_OPERATION.value();

        HashMap<Ref, ServiceResourceStatistic> resourcesMap = null;

        // Get cluster-level statistics.
        try {
            // Get statistics.
            System.out.println("Now trying to get statistics for -" +
                serviceRefs.length + " proxy services...");
            resourcesMap = serviceDomainMbean.getProxyServiceStatistics
                (serviceRefs,typeFlag, null);

            // Reset statistics.
            long resetRequestTime =
                serviceDomainMbean.resetStatistics(serviceRefs);

            // Save retrieved statistics.
            String fileName = "ProxyStatistics_" +
                new SimpleDateFormat("yyyy_MM_dd_HH_mm").
                    format(new Date(System.currentTimeMillis())) +
                        ".txt";
            saveStatisticsToFile(resourcesMap, resetRequestTime,
                fileName);
        }
        catch (IllegalArgumentException iae) {
            System.out.println("===============================\n");
            System.out.println("Encountered IllegalArgumentException...
                Details:");
            System.out.println(iae.getMessage());
                System.out.println("Check if business ref was passed OR bitmap
                    is " + "invalid...\nIf so correct it and try again!!!");
            System.out.println("===============================\n");
                throw iae;
        }


        catch (DomainMonitoringDisabledException dmde) {
            /** Statistics not available as monitoring is turned off at the
             * domain level.
```

```
                        */
                         System.out.println("===================================\n");
                        System.out.println("Statistics not available as monitoring
                             " +
                             "is turned off at domain level.");
                         System.out.println("===================================\n");
                        throw dmde;
                }
                catch (MonitoringException me) {
                        // Internal problem ... May be aggregation server is
                             crashed ...
                         System.out.println("===================================\n");
                          System.out.println("ERROR: Statistics is not available... " +
                             "Check if aggregation server is crashed...");
                         System.out.println("===================================\n");
                        throw me;
                }
        }

        /**
         * Saves statistics of all services from the specified map.
         * @param statsMap Map containing statistics for one or more services
         * of the same type; i.e., business or proxy.
         * @param resetReqTime Reset request time. This information will be
         * written at the end of the file, provided it is not zero.
         * @param fileName Statistics will be saved in a file with this name.
         * @throws Exception
         */
        private void saveStatisticsToFile(
                HashMap<Ref, ServiceResourceStatistic> statsMap,
                     long resetReqTime, String fileName) throws Exception {
                if (statsMap == null) {
                        System.out.println("\nService statistics map is null...
                             Nothing to save.\n");
                }
                if (statsMap.size() == 0) {
                        System.out.println("\nService statistics map is empty...
                             Nothing to save.\n");
                }
                FileWriter out = new FileWriter(new File(fileName));

                out.write("**********************************************\n");
                out.write("This file contains statistics for " + statsMap.size()
                     + " services.\n");
                out.write("**********************************************\n");

                Set<Map.Entry<Ref, ServiceResourceStatistic>> set =
                     statsMap.entrySet();

                System.out.println(new StringBuffer().append("\nWriting stats to
                     the file - ").append(fileName).append("\n").toString());

                // Print statistical information of each service
                for (Map.Entry<Ref, ServiceResourceStatistic> mapEntry : set) {
                        out.write(new StringBuffer().
                                  append("\n\n======= Pirnting statistics for service ").
                              append(mapEntry.getKey().getFullName()).
                              append("=======\n").toString());

                              ServiceResourceStatistic serviceStats = mapEntry.getValue();
```

```
out.write(new StringBuffer().
    append("Statistic collection time is - ").
    append(new Date(serviceStats.getCollectionTimestamp
        ())).
    append("\n").toString());

ResourceStatistic[] resStatsArray = null;
try {
    resStatsArray = serviceStats.getAllResourceStatistics
        ();
}
catch (MonitoringNotEnabledException mnee) {
    // Statistics not available
    out.write("WARNING: Monitoring is not enabled for  " +
        "this service... Do someting...");
    out.write("======================================\n");
    continue;
}

catch (InvalidServiceRefException isre) {
    // Invalid service
    out.write("ERROR: Invlaid Ref. May be this service is
        " +
        "deleted. Do something...");
    out.write("======================================\n");
    continue;
}

catch (MonitoringException me) {
    // Statistics not available
    out.write("ERROR: Failed to get statistics for this
        service... " + "Details: " + me.getMessage());
    me.printStackTrace();
    out.write("======================================\n");
    continue;
}
for (ResourceStatistic resStats : resStatsArray) {
    // Print resource information
    out.write("\nResource name: " + resStats.getName());
    out.write("\tResource type: " +
        resStats.getResourceType().toString());

    // Now get and print statistics for this resource
        StatisticValue[] statValues = resStats.getStatistics();
    for (StatisticValue value : statValues) {
        out.write("\n\t\tStatistic Name - " +
            value.getName ());
        out.write("\n\t\tStatistic Type - " +
            value.getType().toString());

        // Determine statistics type
          if ( value.getType() == StatisticType.INTERVAL ) {
            StatisticValue.IntervalStatistic is =
             (StatisticValue.IntervalStatistic)value;

            // Print interval statistics values
            out.write("\n\t\t\t\tCount Value
                - " + is.getCount());
            out.write("\n\t\t\t\tMin Value - " +
                is.getMin());
```

```
                                        out.write("\n\t\t\t\tMax Value - " +
                                            is.getMax());
                                        out.write("\n\t\t\t\tSum Value - " +
                                            is.getSum());
                                        out.write("\n\t\t\t\tAve Value - " +
                                            is.getAverage());
                            }
                            else if ( value.getType() == StatisticType.
                                COUNT ) {
                                StatisticValue.CountStatistic cs =
                                    (StatisticValue.CountStatistic)
                                        value;

                                // Print count statistics value
                                out.write("\n\t\t\t\tCount Value - " +
                                    cs.getCount());
                            }
                            else if ( value.getType() == StatisticType.STATUS
                                ){
                                StatisticValue.StatusStatistic ss =
                                    (StatisticValue.StatusStatistic)value;
                                // Print count statistics value
                                out.write("\n\t\t\t\t Initial Status - " +
                                    ss.getInitialStatus());
                                out.write("\n\t\t\t\t Current Status - " +
                                    ss.getCurrentStatus());

                            }
                    }
                }
                out.write("\n==========================================\n");
        }
        if (resetReqTime > 0) {
            // Save reset request time.
            out.write("\n****************************************\n");
              out.write("Statistics for all these services are RESET.\n");
            out.write("RESET request time is " +
                new SimpleDateFormat("MM/dd/yyyy HH:mm:ss").
                    format(new Date(resetReqTime)));
            out.write("\n****************************************\n");
        }

        // Flush and close file.
        out.flush();
        out.close();
    }

    /**
     * Init method.
     *
     * @param props Properties required for initialization.
     * @throws Exception
     */
    private void init(HashMap props) throws Exception {
        Properties properties = new Properties();
        properties.putAll(props);
        getServiceDomainMBean(properties.getProperty("HOSTNAME"),
            Integer.parseInt(properties.getProperty("PORT", "7001")),
            properties.getProperty("USERNAME"),
            properties.getProperty("PASSWORD"));
```

```
            serverName = properties.getProperty("SERVER_NAME");
        }


        /**
          * Gets an instance of ServiceDomainMBean from the weblogic server.
          *
          * @param host
          * @param port
          * @param username
          * @param password
          * @throws Exception
          */
        private void getServiceDomainMBean(String host, int port, String
                    username, String password) throws Exception {
            InvocationHandler handler =
                new ServiceDomainMBeanInvocationHandler(host, port,
                        username,password);
            Object proxy = Proxy.newProxyInstance(
                ServiceDomainMBean.class.getClassLoader(),
                new Class[]{ServiceDomainMBean.class}, handler);
            serviceDomainMbean = (ServiceDomainMBean) proxy;
        }

        /**
         * Invocation handler class for ServiceDomainMBean class.
         */
        public static class ServiceDomainMBeanInvocationHandler
                    implements InvocationHandler {
            private String jndiURL =
                "weblogic.management.mbeanservers.domainruntime";
            private String mbeanName = ServiceDomainMBean.NAME;
            private String type = ServiceDomainMBean.TYPE;

            private String protocol = "t3";
            private String hostname = "localhost";
            private int port = 7001;
            private String jndiRoot = "/jndi/";

            private String username = "weblogic";
            private String password = "weblogic";

            private JMXConnector conn = null;
            private Object actualMBean = null;

            public ServiceDomainMBeanInvocationHandler(String hostName, int
                    port, String userName, String password) {
                this.hostname = hostName;
                this.port = port;
                this.username = userName;
                this.password = password;
            }

            /**
              * Gets JMX connection
              * @return JMX connection
              * @throws IOException
              * @throws MalformedURLException
              */
            public JMXConnector initConnection()
```

```
                                throws IOException, MalformedURLException {
                        JMXServiceURL serviceURL = new JMXServiceURL(protocol,
                               hostname, port, jndiRoot + jndiURL);
                        Hashtable<String, String> h = new Hashtable<String,
                               String>();

                        if (username != null)
                               h.put(Context.SECURITY_PRINCIPAL, username);
                        if (password != null)
                               h.put(Context.SECURITY_CREDENTIALS, password);

                        h.put(JMXConnectorFactory.PROTOCOL_PROVIDER_PACKAGES,
                               "weblogic.management.remote");
                        return JMXConnectorFactory.connect(serviceURL, h);
                }

                /**
                  * Invokes specified method with specified params on specified
                  * object.
                  * @param proxy
                  * @param method
                  * @param args
                  * @return
                  * @throws Throwable
                  */
                public Object invoke(Object proxy, Method method, Object[] args)
                        throws Throwable {
                    try {
                        if (conn == null) {
                            conn = initConnection();
                            }
                        if (actualMBean == null) {
                            actualMBean =
                                  findServiceDomain(conn.getMBeanServer
                                  Connection(),mbeanName, type, null);
                         }
                        Object returnValue = method.invoke(actualMBean, args);

                        return returnValue;
                    }
                    catch (Exception e) {
                        throw e;
                    }
                }
                /**
                 * Finds the specified MBean object
                 *
                 * @param connection - A connection to the MBeanServer.
                 * @param mbeanName - The name of the MBean instance.
                 * @param mbeanType - The type of the MBean.
                 * @param parent - The name of the parent Service. Can be NULL.
                  * @return Object - The MBean or null if the MBean was not found.
                 */



                public Object findServiceDomain(MBeanServerConnection connection,
                     String mbeanName,
                     String mbeanType,
```

```
            String parent) {
            ServiceDomainMBean serviceDomainbean = null;
            try {
            ObjectName on =
                new ObjectName(ServiceDomainMBean.OBJECT_NAME);
            serviceDomainbean = (ServiceDomainMBean)
                MBeanServerInvocationHandler.
                    newProxyInstance(connection, on);
            }
            catch (MalformedObjectNameException e) {
                e.printStackTrace();
                return null;
            }
            return serviceDomainbean;
        }
    }
    /**
     * Timer task to keep retrieving and resetting service statistics.
     */
    static class GetAndResetStatisticsTask extends TimerTask {
        private ServiceStatisticsRetriever collector;

        public GetAndResetStatisticsTask(ServiceStatisticsRetriever col
            ){collector = col;
        }

        public void run() {
                System.out.println("\n********************************");
             System.out.println("Retrieving statistics for all monitored
                " + "business services.");




            try {
                    collector.getAndResetStatsForAllMonitoredBizServices();
                System.out.println("Successfully retrieved and reset
                    statistics for " +
                    "all monitored \n business services at " +
                    new SimpleDateFormat("MM/dd/yyyy HH:mm:ss").
                        format(new Date(System.currentTimeMillis())));
            } catch (Exception e) {
                System.out.println("Failed to retrieve and reset
                    statistics for all " + "monitored business
                    service...");
                e.printStackTrace();
            }
            System.out.println("********************************\n");

            System.out.println("\n********************************");
            System.out.println("Retrieving statistics for all
                monitored proxy services.");
            try {
                collector.getAndResetStatsForAllMonitoredProxy
                    Services();
                System.out.println("Successfully retrieved and reset
                    statistics " +
                    "for all monitored \nproxy services at " +
                    new SimpleDateFormat("MM/dd/yyyy HH:mm:ss").
```

```
                                   format(new Date(System.currentTimeMillis()))));
           } catch (Exception e) {
               System.out.println("Failed to retrieve and reset
                   statistics " + "for all monitored proxy service
                           ...");
               e.printStackTrace();
           }
           System.out.println("*********************************\n");
      }
 }



 /**
   * The main method to start the timer task to extract, save, and reset
   * statistics for all monitored business and proxy services. It uses
   * the following system properties.
   * 1. hostname - Hostname of admin server
   * 2. port - Listening port of admin server
   * 3. username - Login username
   * 4. password - Login password
   * 5. period - Frequency in hours. This will be used by the timer
   * to determine the time gap between two executions.
   *
   * @param args Not used.
   */
 public static void main(String[] args) {
     try {
          Properties p = System.getProperties();

          HashMap map = new HashMap();

          map.put("HOSTNAME", p.getProperty("hostname","localhost"));
          map.put("PORT", p.getProperty("port", "7001"));
          map.put("USERNAME", p.getProperty("username", "weblogic"));
          map.put("PASSWORD", p.getProperty("password", "weblogic"));
          //set a server name if you want to get the uri status
               statistics
               in a cluster
          map.put("SERVER_NAME", p.getProperty("server_name","AdminServer"));

          ServiceStatisticsRetriever collector =
               new ServiceStatisticsRetriever();
          String periodStr = p.getProperty("period", "1");
          int periodInHour = Integer.parseInt(periodStr);
          long periodInMilliSec = periodInHour * 60 * 60 * 1000;




          collector.init(map);

          // Start timer.
          Timer timer = new Timer();
          timer.scheduleAtFixedRate(
               new GetAndResetStatisticsTask(collector),
               0, periodInMilliSec);
     }
     catch (Exception e) {
```

```
                    e.printStackTrace();
            }
        }
    }
```

E

SNMP Components

This chapter provides guidelines for creating and using Simple Network Management Protocol (SNMP) resources in Oracle Service Bus.

SNMP is an application-layer protocol that allows the exchange of information on the management of a resource across a network. SNMP lets you to monitor a resource and, if required, take some action based on the data obtained from the resource.

For detailed information on SNMP, see the *Oracle Fusion Middleware SNMP Management Guide for Oracle WebLogic Server*.

E.1 Creating and Using SNMP Agents

This section provides guidelines on creating and targeting SNMP agents in an existing Oracle WebLogic Domain to trap SNMP messages generated by Oracle Service Bus.

For instructions on creating and targeting SNMP agents, see the following topics in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*:

- Create SNMP agents
- Target SNMP agents
- Create trap destinations

E.1.1 Guidelines for Creating and Targeting SNMP Agents for Oracle Service Bus

You can create SNMP agents that are either domain-scoped or server-scoped. Domain-scoped agents, which are targeted to the domain rather than to an individual server, are for backward compatibility and are being deprecated. Oracle recommends that you create server-scoped agents.

When creating and targeting an SNMP server-scoped agent for Oracle Service Bus, use the following guidelines:

Targeting the Agent

When targeting an SNMP agent to Oracle Service Bus, target only the Oracle Service Bus Admin Server. Only agents targeted to the Admin Server receive alerts from Oracle Service Bus. Agents targeted to Managed Servers do not receive SNMP messages.

Creating a Trap Destination

Enter the following settings:

- **Name**: alsbDestination-0
- **Community**: weblogic

- **Host** and **Port**: Set the values to point to the host and port where the SNMP manager is listening for these alerts, such as localhost and 163.

To Start Listening for Traps

1. In a command window, change directories to *WL_ORACLE_HOME*/server/bin, and run the following command:

   ```
   setWLSEnv.cmd(.sh)
   ```

2. Run the following Java command to start the Oracle WebLogic Server SNMP command line utility, which listens for traps and prints them on the server console (using 163 as the listen port for traps):

   ```
   java weblogic.diagnostics.snmp.cmdline.Manager SnmpTrapMonitor -p 163
   ```

   From then on, the traps generated within Oracle Service Bus should reach the running command line utility listening for traps.

   Note: On Solaris, port numbers 0 to 1023 are reserved for root login. If you want to use ports 161 and 163 (as used in this procedure), you may have to start the server and command line utility using root login. To avoid this problem, and to avoid using the root login, specify port numbers above 1023 for both the SNMP agent and the SNMP manager command.

F

Monitoring Statistics in Oracle Service Bus

This chapter describes the service statistics that Oracle Service Bus captures.

In Oracle Service Bus, you can collect many types of statistics when you monitor business services or proxy services. The name of statistics are linked to component of the service for which they are collected using '.'.

Table F–1 lists the different statistics available in Oracle Service Bus:

Table F–1   Different Statistics in Oracle Service Bus

Name	Type of Associated Service	Type
Transport.error-count	Business Service and Proxy Service	count
Transport.message-count	Business Service and Proxy Service	count
Transport.response-time	Business Service and Proxy Service	interval
Transport.failover-count	Business Service	count
Security.WebService Security.wss-error	Business Service and Proxy Service	count
Alert.sla-all	Business Service and Proxy Service	count
Alert.sla-normal	Business Service and Proxy Service	count
Alert.sla-warning	Business Service and Proxy Service	count
Alert.sla-minor	Business Service and Proxy Service	count
Alert.sla-major	Business Service and Proxy Service	count
Alert.sla-critical	Business Service and Proxy Service	count
Alert.sla-fatal	Business Service and Proxy Service	count
Alert.pipeline-all	Proxy Service	count
Alert.pipeline-warning	Proxy Service	count
Alert.pipeline-minor	Proxy Service	count
Alert.pipeline-major	Proxy Service	count

Table F–1   (Cont.) Different Statistics in Oracle Service Bus

Name	Type of Associated Service	Type
Alert.pipeline-critical	Proxy Service	count
Alert.pipeline-fatal	Proxy Service	count
Router.error-count	Proxy Service	count
Router.message-count	Proxy Service	count
Router.validation-errors	Proxy Service	count
Router.Pipeline.pipeline_name.error-count	Proxy Service	count
Router.Pipeline.pipeline_name.message-count	Proxy Service	count
Router.Pipeline.pipeline_name.elapsed-time	Proxy Service	count
Router.Route Node.route_node_name.error-count	Proxy Service	count
Router.Route Node.route_node_name.message-count	Proxy Service	count
Router.Operation Node.route_node_name.elapsed-time	Proxy Service	count
Router.Operation.operation_name.error-count	Proxy Service	count
Router.Operation Node.operation_name.message-count	Proxy Service	count
Router.Operation.operation_name.elapsed-time	Proxy Service	count
Transport.url.url_name.status	Business Service	status
Transport.url.url_name.error-count	Business Service	count
Transport.url.url_name.message-count	Business Service	count
Transport.url.url_name.response-time	Business Service	interval
Transport.uri-offline_count-status	Business Service	status
Transport.throttling-time	Business Service	interval
Transport.cache-hit-count	Business Service	count

For more information about statistics, see Appendix D, "JMX Monitoring API."

Notes: Only the name of the statistics are displayed in the Oracle Service Bus Administration Console.

Statistics `Router.error-count`, `Router.message-count`, and `Router.elapsed-time` are not displayed. Instead, `Transport.error-count`, `Transport.message-count`, and `Transport.elapsed-time` are displayed.

`Success Ratio` and `Failure Ratio` are also displayed in the Administration Console. `Success ratio` is the percentage ratio of, successful messages to total number of messages. `Failure ratio` is the percentage ratio of failed messages to total number of messages.

You cannot obtain any statistic of the type status for a cluster using the JMX Monitoring APIs.

G

Auditing your Oracle Service Bus System

This chapter provides general guidelines on auditing your Oracle Service Bus environment.

In addition to monitoring the services in your Oracle Service Bus system, you can audit your system to determine the history of configuration changes to the system, log the status of messages as they flow through the Oracle Service Bus pipeline at runtime, and log any security violations for messages in the pipeline.

G.1 Auditing the Configuration Changes

When you perform configurational changes in the Oracle Service Bus Administration Console, a track record of the changes is generated and a record of all the configurational changes is maintained. Only the previous image of the object is maintained. You can view or access the history of configurational changes and the list of resources that have been changed during the session only through the Administration Console. However, to access all the information on configuration you have to activate the session.

G.2 Creating an Audit Trail for a Message Flow

Auditing the entire message flow pipeline during is time consuming. However, you can use the reporting action to perform selective auditing of the message flow pipeline during runtime. You insert the reporting action at required points in the message flow pipeline and extract the required information. The extracted information may be then stored in a database or sent to the reporting stream to write the auditing report.

G.3 Auditing Security Violations

When a message is sent to the proxy service and there is a breach in the transport level authentication or the security of the Web Services, Oracle WebLogic Server generates an audit trail. You must configure the Oracle WebLogic Server to generate this audit trail. Using this you can audit all security violations that occur in the message flow pipeline. It also generates an audit trail whenever it authenticates a user. For more information about security auditing, see "Configuring the Oracle WebLogic Security Framework–Main Steps" in the *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

H

XQuery-SQL Mapping Reference

This chapter provides information about the native RDBMS Data Type support and XQuery mappings that the Oracle XQuery engine generates or supports.

This chapter includes the following sections:

- Core RDBMS Data Type Mapping:
  - Section H.1, "IBM DB2/NT 8"
  - Section H.2, "Microsoft SQL Server"
  - Section H.3, "Oracle8i, 8.1.x"
  - Section H.4, "Oracle 9i, Oracle Database 10g"
  - Section H.5, "Sybase 12.5.2 (and higher)"
- Section H.6, "Base (Generic) RDBMS Data Type Mapping"

For information about using these mappings in Oracle Service Bus XQueries, see Section 37.9, "Accessing Databases Using XQuery."

For complete information about database and JDBC drivers support in Oracle Service Bus, see *Oracle Fusion Middleware Supported System Configurations* at:

http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html

H.1 IBM DB2/NT 8

This section identifies the data type mappings that the XQuery engine generates or supports for IBM DB2/NT 8.

Table H–1   IBM DB2 Data Type Mappings

DB2 Data Type	XQuery Type
BIGINT	xs:long
BLOB	xs:hexBinary
CHAR	xs:string
CHAR() FOR BIT DATA	xs:hexBinary
CLOB[1]	xs:string
DATE	xs:date
DOUBLE	xs:double
DECIMAL(p,s)[2] (NUMERIC)	xs:decimal (if s > 0), xs:integer (if s = 0)

Table H–1   (Cont.) IBM DB2 Data Type Mappings

DB2 Data Type	XQuery Type
INTEGER	xs:int
LONG VARCHAR[1]	xs:string
LONG VARCHAR FOR BIT DATA	xs:hexBinary
REAL	xs:float
SMALLINT	xs:short
TIME[3]	xs:time[4]
TIMESTAMP[5]	xs:dateTime[4]
VARCHAR	xs:string[4]
VARCHAR() FOR BIT DATA	xs:hexBinary

[1] Pushed down in project list only.

[2] Where p is precision (total number of digits, both to the right and left of decimal point) and s is scale (total number of digits to the right of decimal point).

[3] Accurate to 1 second.

[4] Values converted to local time zone (time zone information removed) due to TIME and TIMESTAMP limitations.

[5] Precision limited to milliseconds.

H.2 Microsoft SQL Server

This section identifies the data type mappings that the XQuery engine generates or supports for Microsoft SQL Server.

Table H–2   SQL Server 2000 Data Type Mapping

SQL Data Type	XQuery Type
BIGINT	xs:long
BINARY	xs:hexBinary
BIT	xs:boolean
CHAR	xs:string
DATETIME[1]	xs:dateTime[2]
DECIMAL(p,s)[3] (NUMERIC)	xs:decimal (if s > 0), xs:integer (if s = 0)
FLOAT	xs:double
IMAGE	xs:hexBinary
INTEGER	xs:int
MONEY	xs:decimal
NCHAR	xs:string
NTEXT[4]	xs:string
NVARCHAR	xs:string
REAL	xs:float
SMALLDATETIME[5]	xs:dateTime

Table H–2   (Cont.) SQL Server 2000 Data Type Mapping

SQL Data Type	XQuery Type
SMALLINT	xs:short
SMALLMONEY	xs:decimal
SQL_VARIANT	xs:string
TEXT[4]	xs:string
TIMESTAMP	xs:hexBinary
TINYINT	xs:short
VARBINARY	xs:hexBinary
VARCHAR	xs:string
UNIQUIDENTIFIER	xs:string

[1] Fractional-second-precision up to 3 digits (milliseconds). No time zone.

[2] Values converted to local time zone (time zone information removed) and fractional seconds truncated to milliseconds due to DATETIME limitations.

[3] Where *p* is precision (total number of digits, both to the right and left of decimal point) and *s* is scale (total number of digits to the right of decimal point).

[4] Pushed down in project list only.

[5] Accuracy of 1 minute.

H.3 Oracle8*i*, 8.1.x

This section identifies the data types that the XQuery engine generates or supports for Oracle 8.1.x (Oracle 8i).

Table H–3   Oracle 8.1.x Data Type Mapping

Oracle 8 Data Type	XQuery Type
BFILE	not supported
BLOB	xs:hexBinary
CHAR	xs:string
CLOB[1]	xs:string
DATE[2]	xs:dateTime
FLOAT	xs:double
LONG[1]	xs:string
LONG RAW	xs:hexBinary
NCHAR	xs:string
NCLOB[1]	xs:string
NUMBER	xs:double
NUMBER(p,s)[3]	xs:decimal (if s > 0), xs:integer (if s <=0)
NVARCHAR2	xs:string
RAW	xs:hexBinary
ROWID	xs:string
UROWID	xs:string

[1] Pushed down in project list only.

[2] Does not support fractional seconds.

[3] Where *p* is precision (total number of digits, both to the right and left of decimal point) and *s* is scale (total number of digits to the right of decimal point).

H.4 Oracle 9*i*, Oracle Database 10*g*

This section identifies the data type and other mappings that the XQuery engine generates or supports for Oracle 9*i* and Oracle Database 10*g*. Oracle treats empty strings as NULLs, which deviates from XQuery semantics and may lead to unexpected results for expressions that are pushed down.

Table H–4   Oracle 9i, Oracle Database 10g Data Type Mapping

Oracle Data Type	XQuery Type
BFILE	not supported
BLOB	xs:hexBinary
CHAR	xs:string
CLOB[1]	xs:string
DATE	xs:dateTime[2]
FLOAT	xs:double
INTERVAL DAY TO SECOND	xdt:dayTimeDuration
INTERVAL YEAR TO MONTH	xdt:yearMonthDuration
LONG[1]	xs:string
LONG RAW	xs:hexBinary
NCHAR	xs:string
NCLOB[1]	xs:string
NUMBER	xs:double
NUMBER(p,s)	xs:decimal (if s > 0), xs:integer (if s <=0)
NVARCHAR2	xs:string
RAW	xs:hexBinary
ROWID	xs:string
TIMESTAMP	xs:dateTime[3]
TIMESTAMP WITH LOCAL TIMEZONE	xs:dateTime
TIMESTAMP WITH TIMEZONE	xs:dateTime
VARCHAR2	xs:string
UROWID	xs:string

[1] Pushed down in project list only.

[2] When SDO stores xs:dateTime value in Oracle DATE type, it is converted to local time zone and fractional seconds are truncated due to DATE limitations.

[3] XQuery engine maps XQuery xs:dateTime to either TIMESTAMP or TIMESTAMP WITH TIMEZONE data type, depending on presence of time zone information. Storing xs:dateTime using SDO may result in loss of precision for fractional seconds, depending on the SQL type definition.

H.5 Sybase 12.5.2 (and higher)

This section identifies the data types that the XQuery engine generates or supports for Sybase 12.5.2 (and higher).

Note: Sybase deviates from XQuery semantics (which ignores empty strings) and treats empty strings as a single-space string.

Table H–5   Sybase 12.5.2 Data Type Mapping

Sybase Data Type	XQuery Type
BINARY	xs:hexBinary
BIT	xs:boolean
CHAR	xs:string
DATE	xs:date
DATETIME[1]	xs:dateTime[2]
DECIMAL(p,s)[3] (NUMERIC)	xs:decimal (if s > 0), xs:integer (if s == 0)
DOUBLE PRECISION	xs:double
FLOAT	xs:double
IMAGE	xs:hexBinary
INT (INTEGER)	xs:int
MONEY	xs:decimal
NCHAR	xs:string
NVARCHAR	xs:string
REAL	xs:float
SMALLDATETIME[4]	xs:dateTime
SMALLINT	xs:short
SMALLMONEY	xs:decimal
SYSNAME	xs:string
TEXT[5]	xs:string
TIME	xs:time
TINYINT	xs:short
VARBINARY	xs:hexBinary
VARCHAR	xs:string

[1] Supports fractional seconds up to 3 digits (milliseconds) precision; no time zone information.

[2] When SDO stores xs:dateTime value in Oracle DATE type, it is converted to local time zone and fractional seconds are truncated due to DATE limitations.

[3] Where p is precision (total number of digits, both to the right and left of decimal point) and s is scale (total number of digits to the right of decimal point).

[4] Accurate to 1 minute.

[5] Expressions returning text are pushed down in the project list only.

H.6 Base (Generic) RDBMS Data Type Mapping

When mapping SQL to XQuery data types, the XQuery engine first checks the JDBC typecode. If the typecode has a corresponding XQuery type, the XQuery engine uses the matching native type name. If no matching typecode or type name is available, the column is ignored. Table H–6 shows this mapping.

Table H–6   RDBMS Data Type Mapping

JDBC Data Type	Typecode	XQuery Data Type
BIGINT	-5	xs:long
BINARY	-2	xs:string
BIT	-7	xs:boolean
BLOB	2004	xs:hexBinary
BOOLEAN	16	xs:boolean
CHAR	1	xs:string
CLOB[1]	2005	xs:string
DATE	91	xs:date[2]
DECIMAL (p,s)[3]	3	xs:decimal (if s > 0), xs:integer (if s =0)
DOUBLE	8	xs:double
FLOAT	6	xs:double
INTEGER	4	xs:int
LONGVARBINARY	-4	xs:hexBinary
LONGVARCHAR[1]	-1	xs:string
NUMERIC (p,s)[3]	2	xs:decimal (if s > 0), xs:integer (if s =0)
REAL	7	xs:float
SMALLINT	5	xs:short
TIME[4]	92	xs:time[4]
TIMESTAMP[4]	93	xs:dateTime[2]
TINYINT	-6	xs:short
VARBINARY	-3	xs:hexBinary
VARCHAR	12	xs:string
OTHER vendor-specific JDBC type codes	1111	Oracle Service Bus uses native data type name to map to an appropriate XQuery data type.

[1] Pushed down in project list only.
[2] Values converted to local time zone (time zone information removed) due to DATE limitations.
[3] Where p is precision (total number of digits, both to the right and left of decimal point) and s is scale (total number of digits to the right of decimal point).
[4] Precision of underlying RDBMS determines the precision of TIME data type and how much truncation, if any, will occur in translating xs:time to TIME.

I

Interoperability with WSRP

This chapter describes how Oracle Service Bus provides Service Level Agreement (SLA) monitoring in applications that use WSRP.

Web Services for Remote Portlets (WSRP) is a mechanism used to generate markup fragments on a remote system for display in a local portal application.

This chapter includes the following sections:

- Section I.1, "WSRP Producers and Consumers"
- Section I.2, "WSRP Architecture"
- Section I.3, "WSRP Design Concepts"

I.1 WSRP Producers and Consumers

WSRP involves two integral components:

- **WSRP producer** - (referred to as *producer* in this document) is a remote application, implements standards-based Web Services using the SOAP specification over HTTP. You can create a producer using WebLogic Portal or third-party implementations of WSRP.

- **WSRP consumer** (referred to as a *consumer* in this document) is a portal application. Typically, the consumer application references WSDL of the producer when the portal is designed, and the consumer directly accesses the producer.

I.2 WSRP Architecture

This section describes the architecture of WSRP and shows how to enhance the architecture by adding Oracle Service Bus.

Figure I–1 shows the WSRP SOAP request and response flow between a producer application and a consumer application.

Figure I–1   Basic Request/Response Flow Between Producer and Consumer Applications
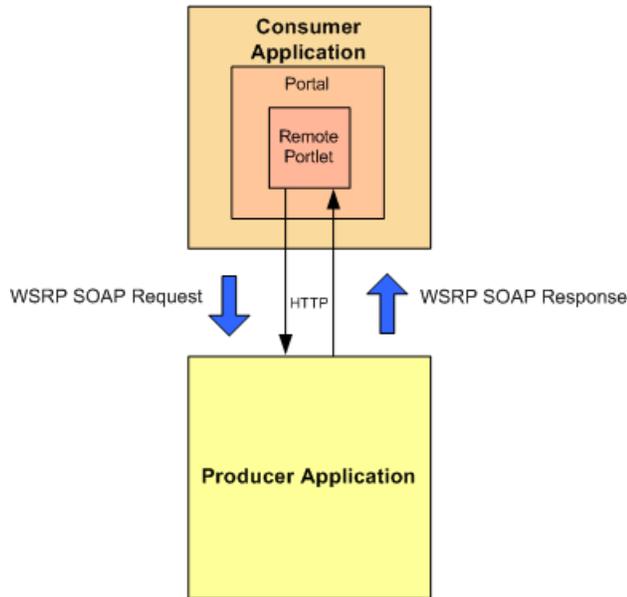
I.2.1 Enhanced Architecture with Oracle Service Bus

Figure I–2 shows how to use Oracle Service Bus as an intermediary between the producer and the consumer to provide Service Level Agreement (SLA) monitoring. You can use Oracle Service Bus in this.

Figure I–2   Enhanced WSRP Request / Response Flow Through Oracle Service Bus
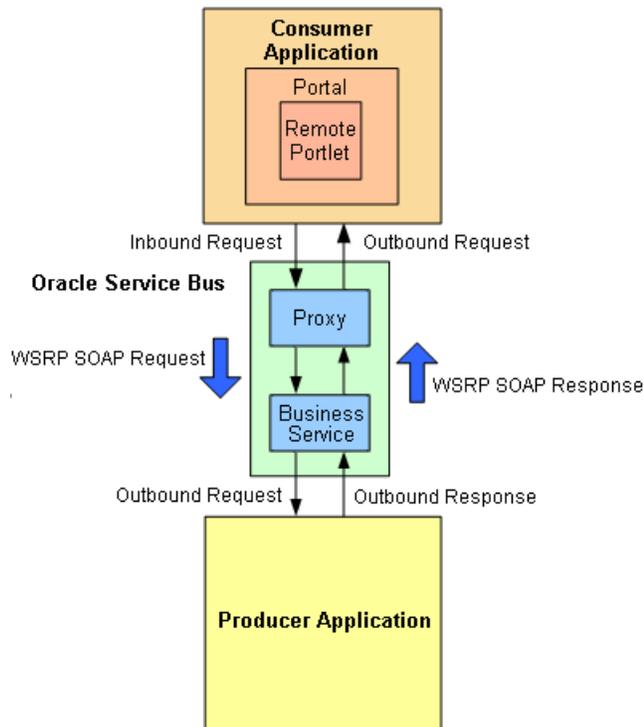
The WSRP SOAP request/response flow occurs in the following sequence:

1. **Inbound Request**: The consumer calls the proxy service in Oracle Service Bus.

2. **Outbound Request**: The proxy service routes the request, which is a message containing the SOAP body and transport headers, to the producer

3. **Outbound Response**: The producer sends a response to Oracle Service Bus.

4. **Inbound Response**: The proxy service sends a response to the consumer. The response is a message that contains the SOAP body and transport headers.

The remainder of this section describes how to configure Oracle Service Bus to send requests for WSRP services through proxy services. It describes services that a producer provides, along with other attributes of WSRP that must be used to configure Oracle Service Bus. This section also discusses how to monitor producers with increasing degrees of detail. Finally, it discusses load balancing and failover with WSRP.

I.3 WSRP Design Concepts

This section describes the following WSRP design concepts:

- Section I.3.1, "WSRP WSDLs"

- Section I.3.2, "WSRP Messages"

I.3.1 WSRP WSDLs

Table I–1 describes various kinds of services offered by WSDLs. WSDLs are referred to as *Producers*.

Table I–1   Producer Services

Service	Description
Service Description	*Required service*. Describes the producer and the portlets that the producer makes available to consumers.
Markup	*Required service*. Manages user interaction with a remote portlet and returns the HTML markup used to render the portlet.
Registration	*Optional service*. Required for complex producers. Allows consumers to register themselves with the producer.
Management	*Optional service*. Provided by complex producers for managing portlet customization and portlet preferences.
Markup Extension	Provided by Oracle WebLogic Portal producers and replaces the Markup service. Markup Extension allows more efficient message handling by using multipart MIME messages for transmitting HTML markup content.

Each producer implements a minimum of two services, such as Service Description and Markup A simple producer offers just these two services. A complex producer, however, provides two additional services, such as Registration and Management. In addition, WebLogic Portal producers implement an extension service, such as Markup Extension that replaces the standard Markup service.

These services are described using a standard WSDL format. The producer supplies a single URL for retrieving its WSDL, which describes all the services that are provided by that producer. The end points for each service indicate whether the consumer should use transport-level security (HTTPs) or abstain from communication with the producer.

I.3.2 WSRP Messages

WSRP uses SOAP over HTTP for all messages exchanged between producers and consumers. In addition to using the standard message formats in the SOAP body, WSRP requires that consumers set at least a `SOAPAction` header, the cookie headers, and the usual HTTP headers, such as `Content-Type`. Producers return a session cookie, and any application-specific cookies, in the HTTP transport header of the response. The consumer must return the session cookie in subsequent request messages.

I.4 Configuring Oracle Service Bus for WSRP

The Oracle Service Bus Administration Console is used to configure Oracle Service Bus. For more information about creating WSRP-enabled portals by using Oracle WebLogic Portal, see the *Oracle Fusion Middleware Federated Portals Guide for Oracle WebLogic Portal*.

Configuring Oracle Service Bus for WSRP involves the following tasks:

- Implementing a service that consumers can invoke to obtain an appropriate WSDL for a particular producer.

- Implementing the details of conveying a consumer's request to the producer and returning the response to the consumer.

This chapter describes the following tasks:

- Section I.4.1, "Getting the Producer WSDL"

- Section I.4.2, "Routing Messages Between Consumer and Producer"

- Section I.4.3, "Monitoring WSRP Applications"

- Section I.4.4, "Load Balancing and Failover"

I.4.1 Getting the Producer WSDL

As a common practice, consumers directly contact a producer to obtain its WSDL. However, if Oracle Service Bus is used as a proxy service, then all access to the producer occurs through Oracle Service Bus. Therefore, a proxy service is implemented for consumers. The proxy service calls the producer's real URL to obtain the producer WSDL. The proxy service transforms the results as follows:

- Rewrite the endpoint address for the producer to refer to the Service Bus IP address and port

- Change the endpoint URI to refer to the Oracle Service Bus proxy service that reflects the required monitoring granularity as described in Section I.4.3, "Monitoring WSRP Applications."

- Change the endpoint protocol and port if transport security is used between the consumer and the Oracle Service Bus proxy service

The developer who creates a producer can specify whether the producer requires SSL (`"secure=true"`). In addition, the Oracle Service Bus administrator can change the security requirement to the consumer through Oracle Service Bus configuration. For example, if a producer does not require SSL, the Oracle Service Bus administrator can require consumers to use SSL by doing the following:

- Changing the WSDL to specify HTTP(s)

- Configuring the proxy services for WSRP to use HTTP(s)

When configured in this way, Oracle Service Bus automatically bridges the secure messages from the consumer to the non-secure messages used by the producer.

I.4.2 Routing Messages Between Consumer and Producer

After retrieving a copy of the WSDL, the consumer uses the WSDL definitions to formulate service requests and sends them to the producer through Oracle Service Bus. The WSRP request/response process involves the following steps:

1. The consumer sends a message to the Oracle Service Bus proxy service corresponding to the producer service.

2. The proxy service executes a simple message flow that routes the message (unchanged) to the actual producer service.

3. The producer formulates and sends a response to the consumer through Oracle Service Bus.

4. The consumer receives the response (unchanged) from the producer.

WSRP web services expose portlets, and those can rely on HTTP cookies and sessions. You must configure Oracle Service Bus to propagate HTTP transport headers, such as `SOAPAction` and cookies. However, by default, Oracle Service Bus does not pass transport headers from the proxy service to the business service because the proxy service may not use the same transport as the business service. Therefore, you must configure the message flow to copy the request headers from the inbound request to the outbound request. Similarly, you must copy the response headers from the business service back to the proxy service's response to the consumer.

Although it is possible to copy all transport headers between the proxy service and the business service, it is necessary to be more selective to avoid errors. You must copy the Set-Cookie and Cookie headers. The final message must own some headers, such as `Content-Length` because Oracle Service Bus is the entity that assembles the final message to send. For example, if the message flow copies the `Content-Length` header from the proxy service to the business service, it can result in an error because the length of the message can change during processing. Therefore, Oracle Service Bus must own this header.

I.4.3 Monitoring WSRP Applications

Monitoring a WSRP application tracks the usage of a producer's individual services and operations. The message flow for WSRP services introduces very little overhead, and the mapping between proxy services and producers, and between business services and producers, is simple to configure. Therefore, to satisfy SLA requirements, it is sufficient to monitor only the proxy services.

For more information about monitoring WSRP Applications, see Chapter 46, "Monitoring Oracle Service Bus at Runtime."

I.4.4 Load Balancing and Failover

Oracle Service Bus allows business services to define multiple endpoints that provide the same web service. When multiple endpoints are defined, Oracle Service Bus can automatically distribute load balance requests across endpoints, and it can automatically failover requests when an endpoint is inaccessible.

I.4.4.1 WSRP Limitations Without Session Stickiness

Portlets are a means of exposing a user interface to an application. Therefore, portlets typically have session data associated with them. To preserve session data, requests to the portlet must be directed to the same server or cluster that serviced the original request. This requirement makes load balancing through Oracle Service Bus inappropriate unless the business process is configured for HTTP session stickiness. Multiple endpoints in a business service usually target different servers or clusters. Without session stickiness configured there is no way to preserve the session because there is no communication among servers that are in separate clusters. Therefore, if multiple endpoints are defined for a WSRP business service and session stickiness is not enabled, you must set the load-balancing algorithm to `"none"`.

You can use multiple endpoints to provide redundancy in certain circumstances in the event that one of the endpoints is not available. The WSRP service is still available on a secondary endpoint. However, any session data that existed at the time the first endpoint failed will not be available on other endpoints.

This failover configuration is an option only for simple producers (see Section I.3.1, "WSRP WSDLs"), not for complex produces. Complex producers require that their consumers register with the producer before sending service requests. The producer returns a registration handle that the consumer must include with each request to that producer. If a business service defines multiple endpoints, each endpoint provides and requires its own registration handle.

However, Oracle Service Bus is stateless across requests—it does not maintain a mapping of the correct handle to send to a particular endpoint. In fact, it sends the registration request to a single endpoint, so the consumer is registered with only one producer. If that producer is not available, then Oracle Service Bus routes a service request to another endpoint defined for that business service. Since the consumer is not registered with that new producer, the request fails with an `"InvalidRegistration"` fault.

The management of registration handles requires an application outside of Oracle Service Bus to maintain this state data. Therefore, the registration requirement avoids defining multiple endpoints for complex producers. As simple producers do not support the registration service, a failover configuration that defines multiple endpoints in the business service is possible although session data is lost on failover.

I.4.4.2 Using WSRP with HTTP Session Stickiness

For load balancing with HTTP business services that have multiple endpoints, you can configure the service to use session stickiness, also known as session affinity. When using sticky sessions, all messages are processed by the server that processed the initial request in the session. Session stickiness can be configured at runtime without needing to restart the server.

When an initial request is sent for a session, it is sent to the first URI endpoint based on the load-balancing algorithm. The session ID is associated with the URI endpoint that serves this initial request, and the session is mapped to that URI index in a URI table. Subsequent requests in the same session go to that URI endpoint instead of the URI endpoint that would have been chosen based on a load-balancing algorithm.

Session stickiness behaves as follows, depending on the business service configuration:

- Messages are retried on the same URI endpoint based on the retry configuration of the business process. If the retry count is exhausted, the URI endpoint is marked as offline and Service Bus throws an exception.

- If a business service URI endpoint is dynamically configured from a message flow, the sessions will not be sticky even if session stickiness is enabled.

- If the URI endpoint is offline, session stickiness is no longer maintained.

- If the service URI table is modified mid-session and the index is out of the bounds of the URI table, a `TransportException` is thrown with an error code of `TransportManager.TRANSPORT_STATUS_APPLICATION_ERROR`.

When configuring business processes for session stickiness, be sure to transfer the cookie headers from inbound to outbound for requests and from outbound to inbound for responses.

I.5 WSRP Interoperability Example

This section describes a WSRP 2.0 interoperability example.

This section discusses the following topics:

- Section I.5.1, "Example Prerequisites"
- Section I.5.2, "Example Projects and Folders"
- Section I.5.3, "Monitoring Example"

I.5.1 Example Prerequisites

The WSRP interoperability example assumes the following components and configuration:

- WebLogic Platform 10.3
- WebLogic Portal 10.3
- Oracle Service Bus 10gR3
- Sample Platform domain configured at `platform:7001`
- Oracle Service Bus domain configured at `alsb:7001`
- Sample Portal application consumer
- Sample producer

I.5.2 Example Projects and Folders

The structure of the sample is divided into two projects—one containing common resources, and the other containing resources for the sample producer.

Table I–2   Projects in the WSRP Interoperability Examples

Folder	Description
wsrp	Contains common resources that are not specific to any producer.
operationExample	Full example supports the most fine-grained monitoring. The folder contains resources specified by the producer. See Section I.5.3, "Monitoring Example."

I.5.3 Monitoring Example

The monitoring configuration example (in the `operationExample folder`) involves configuring Oracle Service Bus to monitor all services and operations of a producer.

The monitoring configuration uses both business services and proxy services that are based on the WSDLs defined by the WSRP standard. The section discusses the following topics:

I.5.3.1 Step 1: Define WSDL Resources

Import all the WSRP WSDL definition files, along with the XML schema files on which the definitions depend. All the files are available as part of the sample code associated with this example, but the standard resource locations are listed in Table I–3.

Table I–3   WSDL Resource Definitions

Resource Name	Type	Location
xml-2.0	XML Schema	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL/xml.xsd
wsrp-2.0-types	XML Schema	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL/wsrp-2.0-types.xsd
wsrp-2.0-interfaces	WSDL	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL/wsrp-2.0-interfaces.wsdl
wsrp-2.0-bindings	WSDL	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL/wsrp-2.0-bindings.wsdl
wsrp-2.0-wsdl	WSDL	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL

Producers generated by BEA Portal extend the standard WSDLs by defining an additional port that allows messages to be sent using MIME attachments. You must define these extension resources if the producer WSDL references them. In this example, an optional task is to create a resource for the WSDL used by the producer. After creating these WSDL and XML Schema resources, edit the references in each resource to resolve the dependencies on other resources.

I.5.3.2 Step 2: Create Business Services

This monitoring example uses the WSDL bindings for each port type implemented by the producer. You must create a separate business service resource for each business service because a business service can be associated with only one WSDL port or binding. A simple producer implements only the required Markup and Service Description interfaces, while a complex producer also implements the Management and Registration interfaces. The services are created identically except for the service name and types, see Table I–4.

Table I–4   Business Service Configuration

Service Name	Service Type
base	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPBaseService"
desc	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPServiceDescriptionService"

Table I–4   (Cont.) Business Service Configuration

Service Name	Service Type
mgmt	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port=" WSRPPortletManagementService
reg	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port=" WSRPRegistrationService"

For each service, the required attributes are listed in Table I–5.

Table I–5   Service Attributes for Business Services

Name	Value	Comments
Protocol	HTTP	Or HTTP(s) if the producer was created with secure= `true`.
Load Balancing Algorithm	none	Must be `none`, or session data will be lost across requests if multiple end points are defined.

Endpoint URI	Service Description:
	
	`http://host*:port+/producer/producer/wsrp-2.0/serviceDescription`
	
	Markup:
	
	`http://host*:port+/producer/producer/wsrp-2.0.0/markup`
	
	Registration:
	
	`http://host*:port+/producer/producer/wsrp-2.0/registration`
	
	Portlet Management:
	
	`http://host*:port+/producer/producer/wsrp-2.0/portletManagement`
		Multiple endpoints must be defined for WSRP producers.

I.5.3.3 Step 3: Create Proxy Services

Proxy services in this monitoring example are configured as follows:

- The proxy services must be based on the same WSDL because the business services are based on a WSDL.

- One proxy service is created for each business service, but each proxy service must have a different URI.

- The configuration must specify which operation is being invoked.

To create a proxy service:

1. Create the proxy service for the base WSRP service.

    As in the earlier example, create the proxy service using the existing `operationExample/base` business service as the model. This model creates the proxy service based on the same WSDL binding as the business service, and it creates a message flow with an unconditional route action to the business service. For the endpoint URI, you can use any URI, such as the producer name with the port type abbreviation appended to it (for example, `/operationExampleBase`).

2. Edit the message flow to add the same transformations required to copy the request transport headers and response transport headers between the consumer and producer.

WSRP relies on data conveyed in the transport headers to function properly. In particular, producers return session cookies to consumers in the response headers if they expect consumers to supply session cookies in subsequent requests. Similarly, producers expect consumers to provide the requested operation in the `SOAPAction` request header.

By default, Oracle Service Bus does not copy transport headers from the inbound request to the outbound request, or from the outbound response to the inbound response. The message flow must propagate the required headers both in and out of the business service.

> **Note:** To retrieve all the headers from the transport, select **Yes** in the **Get All Headers** field of the **Transport Configuration** page of the Oracle Service Bus proxy. For more information, see Section 20.2.3, "Transport Configuration Page."
>
> Add Transport headers to the **Request & Response** actions of the **Route Node** in the Message Flow and enable the **Pass all headers through pipeline option**. For more information, see Section 21.7, "Adding Transport Header Actions." Oracle Service Bus automatically ensures that content-length is not copied.

3. Alternatively, while configuring the routing to a business service using the Oracle Service Bus Administration Console, select the **Use inbound operation for outbound** check box when you are editing a route node to avoid low level XQuery manipulation, as in Figure I–3.

Figure I–3   Passing an Operation from Inbound to Outbound

With this transformation, the operation for the business service is dynamically set to the same value as was specified for the proxy service. Oracle Service Bus counts and monitors all operations of the service.

I.5.3.4 Alternative Methods to Create Proxy Services

The proxy services for the other business services can be created by repeating these steps although you can use a shortcut to avoid recreating all of the transformations manually.

For example, to create the proxy service for the Service Description service:

1. Create a new proxy service using the existing `operationExample/base` proxy service that is created as the model. Following this example, use `/operationExampleDesc` for the endpoint URI.

2. On the Summary Page, click the **Edit** link for General Configuration. The WSDL binding is created using the Base port, so modify the binding to refer to the `WSRPServiceDescriptionService` port.

3. Edit the message flow. The route action refers to the base business service. Modify the route action to the `desc` service.

> **Note:** Use the Transport Header action to minimize low level Xquery manipulation, and simplify the configuration of a proxy service. See Section 21.7, "Adding Transport Header Actions" for details.

For each service, the proxy service configurations are listed in Table I–6.

Table I–6   Proxy Service Configuration

Service Name	Service Type
proxyBase	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPBaseService"
proxyDesc	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPServiceDescriptionService"
proxyMgmt	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPPortletManagementService"
proxyReg	WSDL port: operationExample-2.0/wsrp-2.0-wsdl, port="WSRPRegistrationService"

For each proxy service, the required attributes are listed in Table I–7.

Table I–7   Required Attributes

Name	Protocol
Protocol	HTTP
Get All Headers	Yes
Endpoint URI	For WebLogic Platform 10.0/9.2 the URLs are as follows: ■ proxyBase: `/operationExampleBase-2.0` ■ proxyDesc: `/operationExampleDesc-2.0` ■ proxyReg: `/operationExampleReg-2.0` ■ proxyMgmt: `/operationExampleMgmt-2.0`

I.5.3.5 Step 4: Retrieve the WSDL from the Producer

Create a service that retrieves the WSDL specific to WSRP 2.0 from the producer and transform it to hide the actual producer endpoints. In this example, the proxies for each producer have a different URI. In addition, the section describes how to create the resources to retrieve the producer WSDL.

I.5.3.5.1 Step 1: Create the Business Service to Retrieve the WSDL Specific to WSRP 2.0 Create a business service to obtain the WSDL from the producer. This resource is specific to the producer, so you must create the resource in the operationExample project. Table I–8 describes the properties of the business service.

Table I–8   Business Service Configuration Properties

Name	Value	Comments
Service Name	wsdlSvc 2.0	Any name is allowed.
Service Type	Any XML Service	Consumers usually retrieve the WSDL from the producer using an HTTP GET request. Only XML services support GET.
Protocol	HTTP	HTTP
Load Balancing Algorithm	None	None preferred.
Endpoint URI	http://platform:7001/producer/producer/wsrp-2.0/markup?WSDL	Although you can specify multiple endpoints for retrieving the WSDL, it has no additional use or benefit.
HTTP Request Method	GET	

I.5.3.5.2 Step 2: Create an XQuery Expression to Construct URLs You must transform all endpoint addresses in the producer's WSDL to reflect the Oracle Service Bus server address and the proxy service URI values. You must create an XQuery expression to simplify the construction of the endpoint locations because each producer WSDL can have four or more ports defined. The XQuery expression accepts the following three string variables as input and concatenates them together to form a SOAP address element:

- **base URL** for the Oracle Service Bus server

- **name** to identify the producer

- **extension** used to differentiate ports for a producer

Table I–9 provides the query definition in the `wsrp` project.

Table I–9   XQuery Definition in the wsrp Project

Name	Value
Resource Name	`wsrp/addr`

XQuery	`declare variable $baseURL external;`
	`declare variable $name external;`
	`declare variable $svc external;`
	`declare namespace`
	`soap="http://schemas.xmlsoap.org/wsdl/soap/";`
	`<soap:address location="{concat($baseURL, $name, $svc)}"/>`

I.5.3.5.3 Step 3: Create a No-Op Proxy Service Create a service that does nothing. To create this service, define a new proxy service in the `wsrp` project folder with the resource name `nullSvc`. Accept all of the defaults for this service. Configuring this proxy service creates a message flow for the service of an echo node.

I.5.3.5.4 Step 4: Create a Common Proxy Service to Retrieve the WSDL Specific to WSRP 2.0
Create a proxy service used by consumers to get WSDLs from producers. This proxy service is appropriate for any producer configuration modeled on this sample. The example described in this section is only a suggestion–a different approach is necessary based on the specific requirements of a given implementation. You must create this proxy service in the `wsrp` project folder because the proxy service is not specific to a single producer.

- The approach used in this step requires the administrator to assign each producer a name that is included in part of the URL to retrieve the WSDL.

- The message flow for the proxy service extracts the name from the URL, uses it to locate the business service specific to that producer, obtain the WSDL, and then transforms the WSDL to rewrite the endpoints to Oracle Service Bus.

- The proxy service endpoint URI is configured as `/getWSDL`, and the URL that consumers use to obtain a WSDL is as follows:

  ```
  http://alsb:7001/getWSDL/<producerName>
  ```

  where `<producerName>` is the name assigned to the producer by the administrator. In this example, the producer is `operationExample`.

Table I–10 describes the configuration properties for the proxy service `getWSDL2.0`:

Table I–10   Proxy Service Configuration Properties

Property Name	Value
Service Name	getWSDL2.0
	Any name is allowed.
Service Type	Any XML Service
Protocol	HTTP
Endpoint URI	/getWSDL2.0

The message flow for this proxy service consists of a pipeline pair and a route node. The request side of the pipeline pair consists of a single stage whose job is to extract the producer name from the URL and assign it to a context variable. The action is:

```
Assign $inbound/ctx:transport/ctx:request/http:relative-URI to variable
producerName
```

The response side of the message flow is a stage where all the transformations are performed. Before executing the Replace Actions to transform the WSDL, assign the base URL of the Oracle Service Bus server to a context variable to avoid specifying it on every transformation:

```
Assign "http://alsb:7001/" to variable nonSecureBaseURL
```

Edit the stage of the Response Pipeline to modify each Replace Action to make the transformation match the Endpoint URI given to the proxies created earlier. In this example, the proxies were created using the producer name with an abbreviated service type appended to it. The `addr` XQuery resource created earlier accepts an extension argument to construct the URI location. Simply change that argument to the

proper value, as listed in Table I–11.

Table I–11   Extension Settings to Construct the URI Location

If @binding is	svc arg of addr is
WSRP_v2_Markup_Binding_SOAP	"Base"
WSRP_v2_ServiceDescription_Binding_SOAP	"Desc"
WSRP_v2_PortletManagement_Binding_SOAP	"Mgmt"
WSRP_v2_Registration_Binding_SOAP	"Reg"
WLP_WSRP_v2_Markup_Ext_Binding_SOAP	"Ext"

You must map `name:` to `$producerName` and `BaseURL` to `$nonSecureBaseURL` similar to the `svg arg` mapping in the use table: table num_xref, Section I–1, "Basic Request/Response Flow Between Producer and Consumer Applications."

The five Replace Actions are defined, as in the following code listing. The value of name is replaced with the binding names from the table.

```
Replace
./*[local-name()="definitions"]/*[local-name()="service"]/*[local-name()="port"][e
nds-with(attribute::binding,"name")]/*[local-name()="address"
Replace entire node
name
WSRP_v2_Markup_Binding_SOAP
WSRP_v2_ServiceDescription_Binding_SOAP
WSRP_v2_PortletManagement_Binding_SOAP
WSRP_v2_Registration_Binding_SOAP
```

For the first Replace Action, you must add the User Namespace definitions listed in Table I–12:

Table I–12   User Namespace Definitions on Replace Action

Prefix	Namespace
wsdl	http://schemas.xmlsoap.org/wsdl/
soap	http://schemas.xmlsoap.org/wsdl/soap/

The route node of this message flow consists of a routing table that selects the case based on `$producerName`. For each known producer, add cases so that each case routes to the correct business service to retrieve the WSDL if the name matches. This example uses the following directive:

```
= "operationExample" Route to wsdlSvc
```

1. Add a Default Case that routes to the no-op service to handle cases in which an unknown producer name is specified:

   ```
   Default Route to nullSvc
   ```

2. In this example, return an HTTP 404 status code by adding these response actions to the default case:

   ```
   Insert <http:http-response-code>404</http:http-response-code> as last child of
   ./ctx:transport/ctx:response in variable inbound
   Reply With Failure
   ```

3. Edit the Routing Table in the route node to make the cases correspond to the producers known to the system.

I.5.3.5.5   Step 5: Define the Message Flow of getWSDL2.0 Proxy Service to Modify WSDL to Use Oracle Service Bus URLs   The message flow for `getWSDL2.0` proxy service consists of a pipeline pair and a route node. To define the message flow for this proxy service, do the following:

1. Create a Pipeline Pair

2. Edit the request side of pipeline pair

   The request side of the pipeline pair consists of a single stage whose job is to extract the producer name from the URL and assign it to a context variable. To achieve this, add the Assign action as follows:

   Assign `$inbound/ctx:transport/ctx:request/http:relative`-URI to variable producerName

3. Edit the response side of pipeline pair

   The response side of the message flow is a stage where all the transformations are performed.

   ■ Create an Assign action to assign the base URL of the Oracle Service Bus server to context variable to avoid specifying it on every transformation:

   Assign `http://alsb:7001/` to variable nonSecureBaseURL

   ■ Create Replace actions to change the URLs in the response WSDL to make them point to the corresponding proxy services created earlier. Add the Replace action as follows:

   – Add Replace action.

   – In the Xpath, specify

   ```
   ./*[local-name()="definitions"]/*[local-name()="service"]/*[local-name(
   )="port"][ends-with(attribute::binding,"WSRP_v2_Markup_Binding_
   SOAP")]/*[local-name()="address"][starts-with(attribute::location,"http
   :")]
   ```

   – In text box, specify variable body.

   – In the Expression, Select the Xquery resource addr (which was added earlier as a part of wsrp 1.0 configuration) and set svc to `Base-2.0`, baseURL to $nonSecureBaseURL, and name to $producerName.

   – Select the option **Replace entire node**.

   – Create the remaining Replace actions, as described in the following table.

If attribute binding in Xpath is	svc argument of addr Xquery is
WSRP_v2_Markup_Binding_SOAP	"Base-2.0"
WSRP_v2_ServiceDescription_Binding_SOAP	"Desc-2.0"
WSRP_v2_PortletManagement_Binding_SOAP	"Mgmt-2.0"
WSRP_v2_Registration_Binding_SOAP	"Reg-2.0"

4. Add the route node to the pipeline pair. The route node of this message flow consists of a routing table that selects the case based on $producerName. For each known producer, add cases so that each case routes to the correct business service

to retrieve the WSDL if the name matches. This example uses the following directive:

```
= "operationExample" Route to wsdlSvc-2.0
```

a. Add a **Default Case** that routes to the no-op service to handle cases in which an unknown producer name is specified:

```
Default Route to nullSvc
```

b. In this example, return an HTTP 404 status code by adding these response actions to the default case:

c. Insert `<http:http-response-code>404</http:http-response-code>` as last child of **./ctx:transport/ctx:response** in variable inbound

```
Reply With Failure
```

d. Edit the **Routing Table** in the route node to make the cases correspond to the producers known to the system.

I.5.3.5.6   Step 6: Change the Message Flow of Proxy Service getWSDL to Enable WSRP 2.0 WSDL Retrieval Using Proxy Service getWSDL2.0   In order to consume the producer, the consumer must register the producer using its WSDL. By default, the WSDL uses WSRP version 1.0. To make consumer use WSRP version 2.0, producer WSDL with WSRP version 2.0 must be made available to the consumer. Generally, the default producer WSDL (one which uses WSRP 1.0) contains a link to the WSDL with WSRP version 2.0. This link enables consumer to use WSDL of WSRP version 2.0.

The getWSDL proxy service provides the WSDL of the producer with WSRP 1.0 version. This WSDL contains an URL of the WSDL of the same producer with version WSRP 2.0. This is the direct URL of the producer's WSDL. Instead of using the direct URL, you must the access the WSDL through Oracle Service Bus using proxy service `getWSDL2.0`. This can be achieved in following way.

1. Create an Xquery resource to construct the URL of WSDL, which uses WSRP 2.0

Resource Name: `import`

Xquery:

```
declare variable $baseURL external;
declare variable $name external;
declare variable $svc external;
<import location="{concat($baseURL, $svc, $name )}"
namespace="urn:oasis:names:tc:wsrp:v2:wsdl"
xmlns="http://schemas.xmlsoap.org/wsdl/" />
```

2. Add the Replace action in the response side of message flow of `getWSDL proxy service`. This replace action replaces the WSDL URL in the response to the `getWSDL2.0` proxy service endpoint URI. This can be done as follows:

- Add Replace action.

- In Xpath, specify

```
./*[local-name()="definitions"]/*[local-name()="import"][ends-with(attribut
e::location,"/producer/wsrp-2.0/markup?WSDL")]
```

- In the text box, specify variable body.

- In **Expression**, select the Xquery resource import and set svc to `getWSDL2.0`, baseURL to `$nonSecureBaseURL`, and name to `$producerName`.

- Select the option **Replace entire node**.

I.5.3.6 Step 5: Verify the Configuration

After completing the configuration, verify it as follows:

1. Retrieve the WSDL from a regular browser window by entering the following URL:

    http://alsb:7001/getWSDL/operationExample

2. Verify that all of the end point WSRP end point URLs (except for the BEA extension service) have been changed to correctly refer to the proxy service values on the Oracle Service Bus server.

3. Create a remote portlet in a Portal consumer application, specifying this URL as the address of the WSDL for the producer.

    Use either Eclipse or the Portal Administration Tools to create the remote portlet. Except for entering a different URL to retrieve the WSDL, the steps to create this portlet are the same as those used to create the portlet that is not proxied by Oracle Service Bus.

4. After the consumer portal is complete, run the application.

5. Enable monitoring on the Oracle Service Bus components that you have chosen.

6. Use the Oracle Service Bus Administration Console to navigate to message counts and performance statistics on all WSRP services and operations handled by the producer.

J

Diagnosing Problems with Oracle Service Bus

This appendix describes how to identify Service Bus problems and take the proper corrective actions with the assistance of the WebLogic Diagnostic Framework (WLDF) and the Oracle Fusion Middleware Diagnostic Framework (DFW).

This appendix includes the following sections:

- Section J.1, "About Diagnostics for Service Bus"
- Section J.2, "About the WebLogic Diagnostic Framework"
- Section J.3, "About the Automatic Diagnostic Repository"
- Section J.4, "Supported Oracle Service Bus Diagnostic Dumps"
- Section J.5, "Generating the Diagnostic Dump Using RDA"

J.1 About Diagnostics for Service Bus

Service Bus leverages Oracle's Diagnostic Framework (DFW) and WebLogic Diagnostic Framework (WLDF) to help you diagnose problems in Oracle Service Bus. WLDF lets you to monitor diagnostic scenarios using watches and notifications. DFW lets you gather diagnostic scenarios specific to Service Bus into data dumps that are formatted for viewing and analyzing.

WebLogic and SOA Suite both provide several predefined diagnostic dumps to help you with diagnostics. In addition, Service Bus supports the following diagnostic dumps:

- Derived Resource Caches
- JMS Request/Response Correlation Table
- MQ Request/Response Correlation Table

For information about the diagnostic frameworks, watches, and notifications, see "Diagnosing Problems" in the *Oracle Fusion Middleware Administrator's Guide*. For information about using the diagnostic frameworks with SOA Suite (including generated dumps, setting up watches and notifications, and predefined diagnostic dumps), see "Diagnosing Problems with SOA Composite Applications" in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

J.2 About the WebLogic Diagnostic Framework

WLDF is a monitoring and diagnostics framework included with Oracle WebLogic Server that defines and implements a set of services that run within server processes

and participate in the standard server life cycle. With WLDF, you can capture diagnostic data and set watches and notifications when certain conditions occur. Use watches and notifications to collect diagnostic data to identify problems. This data enables you to isolate and diagnose faults when they occur.

For more information about WLDF, see *Oracle Fusion Middleware Configuring and Using the Diagnostics Framework for Oracle WebLogic Server*.

J.3 About the Automatic Diagnostic Repository

The Automatic Diagnostic Repository (ADR) is a file-based hierarchical repository for diagnostic data, such as traces and dumps. The Oracle Fusion Middleware components store all incident data in the ADR, and each Oracle WebLogic Server stores diagnostic data in subdirectories of its own home directory within the ADR. For more information about the ADR, see "Diagnosing Problems with SOA Composite Applications" in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

J.4 Supported Oracle Service Bus Diagnostic Dumps

In addition to the diagnostic dumps available with Oracle WebLogic Server and Oracle SOA Suite, Service Bus supports the creation of the diagnostic dumps in Table J–1.

Table J–1   Supported Oracle Service Bus Diagnostic Dumps

Dump	Description
OSB.derived-caches	A collection of statistics about all Service Bus derived resource caches on the server
OSB.jms-async-table	Service Bus JMS request/response correlation table
OSB.mq-async-table	Service Bus MQ request/response correlation table

J.4.1 Listing the Available Diagnostic Dumps

The Diagnostic Framework outputs and records the diagnostic dumps. You can list details about all the supported diagnostic dumps with the WLST `listDumps` and `describeDump` commands.

> **Note:** You must start `WLST` from `MW_HOME`/oracle_ common/common/bin. Otherwise, the ODF functions are missing.

To list the supported dumps:

1. Connect to the server on which the SOA Infrastructure is installed.

```
wls:/offline> connect('user_name', 'password',
't3://myhost:7001')
Connecting to t3://myhost:8001 with userid user_name ...
Successfully connected to managed Server 'soa_server1' that belongs to
domain 'soainfra'.
```

2. List the supported Diagnostic Framework dumps.

```
wls:/soainfra/serverConfig> listDumps()
odl.activeLogConfig
jvm.classhistogram
dms.ecidctx
```

```
wls.image
odl.logs
dms.metrics
odl.quicktrace
http.requests
jvm.threads

Use the command describeDump(name=<dumpName>) for help on a specific dump.
```

3. List the supported Oracle Service Bus dumps.

```
wls:/soainfra/serverConfig> listDumps(appName='OSB')
OSB.derived-caches
OSB.jms-async-table
OSB.mq-async-table

Use the command describeDump(name='<dumpName>') for help on a specific dump.
```

For more information about listDumps and describeDump, see "Diagnostic Framework Custom WLST Commands" in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

For more information about Diagnostic Framework dumps, see "Diagnosing Problems" in *Oracle Fusion Middleware Administrator's Guide*.

J.4.2 Derived Resource Caches Diagnostic Dumps (OSB.derived-caches)

Table J–2 provides details about Service Bus derived resource caches diagnostic dumps. The information captured includes the name of each cache type, statistical information for each cache, and information about each cached entry.

Table J–2   JMS Correction Table Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
OSB.derived-caches	None	For each derived resource cache managed in the OSB runtime, the following information is provided:
		■ Derived resource cache type
		■ Product version
		■ Total number of configured cache entries
		■ Cache entries in use
		■ Total hits to entries in the cache server since the server was last started
		■ Total misses while trying to access cached information since the server was last started
		■ Hit ratio of the cache sine the server was last started
		For each cache entry, the following information is provided:
		■ Ref that is being cached
		■ Create date and time
		■ Amount of time spent computing the cache entry. This is the time taken to create the cached information in milliseconds.

Table J–3 lists each Service Bus cache included in the diagnostic information.

Table J–3   Oracle Service Bus Derived Resource Caches

Cache	Description
Archive ClassLoader	Dependency-aware archive class loaders.
Archive Summary	Archive summaries.
CodecFactory	Codec factories.
EffectiveWSDL	EffectiveWSDL objects that are derived from the service or WSDL resources of business or proxy services.
Flow_Info	Split-join objects.
LightweightEffectiveWSDL	EffectiveWSDL objects that are derived from the service or WSDL resources of business or proxy services.
MflExecutor	MFL executors.
RouterRuntime	Compiled router runtimes for proxy services.
RuntimeEffectiveWSDL	Session valid effectiveWSDL objects derived from the service or WSDL resources of business or proxy services.
RuntimeEffectiveWSPolicy	WS policies for business or proxy services.
SchemaTypeSystem	Type system information for MFL, XS, and WSDL documents.
ServiceAlertsStatisticInfo	Service alert statistics for business or proxy services.
ServiceInfo	Compiled service information for business or proxy services and for WSDL documents.
Wsdl_Info	WSDL information objects.
WsPolicyMetadata	Complied WS-Policy metadata.
XMLSchema_Info	XML schema information for XML schema objects.
XqueryExecutors	XQuery executors.
XsltExecutor	XSLT executors.
alsb.transports.ejb.bindingtype	EJB binding information for EJB business services.
alsb.transports.jejb.business.bindingtype	JEJB binding information for JEJB business services.
alsb.transports.jejb.proxy.bindingtype	JEJB binding information for JEJB proxy services.

J.4.2.1 WLST Command Dump Description and Execution

1. Enter the following WLST command line syntax to display a dump description of `OSB.derived-caches`:

```
wls:/soainfra/serverConfig>
describeDump(name='OSB.derived-caches',appName='OSB')
```

The following information appears:

```
Name: OSB.derived-caches
Description: Dump of the caches used for OSB derived resources.
Mandatory Arguments:
Optional Arguments:
```

2. Enter the following WLST command line syntax to execute a dump of `OSB.derived-caches`:

```
wls:/soainfra/serverConfig> executeDump(name='OSB.derived-caches',
appName='OSB')
```

Information similar to the following appears (parts of this dump have been truncated for readability).

```
<derivedCaches xmlns="http://www.bea.com/wli/config/xmltypes">
 <version>11.1.1.7</version>
  <derivedCache cacheType="RuntimeEffectiveWSDL">
    <configuredEntries>2147483647</configuredEntries>
    <cacheEntriesInUse>0</cacheEntriesInUse>
    <totalHits>0</totalHits>
    <totalMisses>0</totalMisses>
    <hitRatio>0.0</hitRatio>
    <cacheEntries/>
  </derivedCache>
 ...
 <derivedCache cacheType="ServiceAlertsStatisticInfo">
    <configuredEntries>2147483647</configuredEntries>
    <cacheEntriesInUse>9</cacheEntriesInUse>
    <totalHits>0</totalHits>
    <totalMisses>51</totalMisses>
    <hitRatio>0.0</hitRatio>
    <cacheEntries>
        <cacheEntry>
            <ref>services/bs_dq_uri4.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.737-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/bs_dq_nopooling.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.736-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/bs_dq_uri1.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.738-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/proxy_dq_uri.ProxyService</ref>
            <creationTime>2012-03-22T23:44:53.736-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/bs_dq_conn_pooling.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.736-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/bs_dq_conn_nopooling.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.737-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
        </cacheEntry>
        <cacheEntry>
            <ref>services/bs_dq_uri2.BusinessService</ref>
            <creationTime>2012-03-22T23:44:53.737-07:00</creationTime>
            <computeTimeMSecs>0</computeTimeMSecs>
```

```
            </cacheEntry>
            <cacheEntry>
                <ref>services/bs_dq_pooling.BusinessService</ref>
                <creationTime>2012-03-22T23:44:53.736-07:00</creationTime>
                <computeTimeMSecs>0</computeTimeMSecs>
            </cacheEntry>
            <cacheEntry>
                <ref>services/bs_dq_uri3.BusinessService</ref>
                <creationTime>2012-03-22T23:44:53.737-07:00</creationTime>
                <computeTimeMSecs>0</computeTimeMSecs>
            </cacheEntry>
        </cacheEntries>
    </derivedCache>
    ...
</derivedCaches>
```

J.4.3 JMS Correlation Table Diagnostic Dumps (OSB.jms-async-table)

Table J–4 provides details about Service Bus JMS request/response correlation table diagnostic dumps. The information captured includes the correlation ID, expiration, and destination for each message.

Table J–4   JMS Correlation Table Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
OSB.jms-async-table	None	In addition to the Service Bus version, the following information is provided for each pending message in each service reference: ■ Correlation ID (could be the actual correlation ID or a message ID) ■ Expiration date and time ■ Message destination

J.4.3.1 WLST Command Dump Description and Execution

1. Enter the following WLST command line syntax to display a dump description of OSB.jms-async-table:

    ```
    wls:/soainfra/serverConfig>
    describeDump(name='OSB.jms-async-table',appName='OSB')
    ```

    The following information appears:

    ```
    Name: OSB.jms-async-table
    Description: Dump of the OSB JMS Async Table of pending messages
    Mandatory Arguments:
    Optional Arguments:
    ```

2. Enter the following WLST command line syntax to execute a dump of OSB.jms-async-table:

    ```
    wls:/soainfra/serverConfig> executeDump(name='OSB.jms-async-table',
    appName='OSB')
    ```

    Information similar to the following appears:

    ```
    <transportDiagnosticsContents xmlns="http://www.bea.com/wli/sb/transportdiags">
    ```

```
<version>11.1.1.7</version>
<transportDiagnostics transportType="jms">
  <correlationTable>
    <services>
      <service>
        <ref>default/testJmsResponseRollback_out</ref>
        <message>
          <correlationMsgId responsePattern="JMSCorrelationID">
             ID:42454153155cc06b7f5ab312000001363d5bd59effff8d4
          </correlationMsgId>
        <expirationTime>2012-03-22T19:53:43.621-07:00</expirationTime>
        <msgDestination>testJmsResponseRollback_outRequest</msgDestination>
        </message>
      </service>
    </services>
  </correlationTable>
 </transportDiagnostics>
</transportDiagnosticsContents>
```

J.4.4 MQ Correlation Table Diagnostic Dumps (OSB.mq-async-table)

Table J–5 provides details about Service Bus MQ request/response correlation table diagnostic dumps. The information captured includes the correlation ID, expiration, and destination for each message.

Table J–5   MQ Correlation Table Diagnostic Dumps

Dump Name	Dump Parameters/Dump Mode	Information Captured
OSB.mq-async-table	None	In addition to the Service Bus version, the following information is provided for each pending message in each service reference: ■ Correlation ID (could be the actual correlation ID or a message ID) ■ Expiration date and time ■ Message destination

J.4.4.1 WLST Command Dump Description and Execution

1. Enter the following WLST command line syntax to display a dump description of OSB.mq-async-table:

    ```
    wls:/soainfra/serverConfig>
    describeDump(name='OSB.mq-async-table',appName='OSB')
    ```

    The following information appears:

    ```
    Name: OSB.mq-async-table
    Description: Dump of the OSB MQ Async Table of pending messages
    Mandatory Arguments:
    Optional Arguments:
    ```

2. Enter the following WLST command line syntax to execute a dump of OSB.mq-async-table:

    ```
    wls:/soainfra/serverConfig> executeDump(name='OSB.mq-async-table',
    appName='OSB')
    ```

    Information similar to the following appears:

```
<transportDiagnosticsContents xmlns="http://www.bea.com/wli/sb/transportdiags">
 <version>11.1.1.7</version>
 <transportDiagnostics transportType="mq">
   <correlationTable>
     <services>
       <service>
         <ref>services/mq_Biz_cached</ref>
         <message>
           <correlationMsgId responsePattern="MQCorrelationID">
             000000000000000000000000000000000000000000000000
           </correlationMsgId>
           <expirationTime>2012-03-22T23:48:09.085-07:00</expirationTime>
           <msgDestination>rc_req</msgDestination>
         </message>
       </service>
     </services>
   </correlationTable>
 </transportDiagnostics>
</transportDiagnosticsContents>
```

J.5 Generating the Diagnostic Dump Using RDA

In addition to generating Service Bus diagnostic dumps using WSLT, you can also use Oracle Remote Diagnostic Agent (RDA). Before performing the following steps, make sure RDA is installed on your system.

> **Tip:** For more information about using RDA for Service Bus, refer to the knowledge base article, "*How to Collect Analysis Information using RDA for Oracle Service Bus (OSB),*" on Oracle support. The ID for this document is 1352549.1.

To generate a diagnostic dump using RDA

1. From a command line, run the following command:

    For Windows:

    ```
    rda.cmd -vSCRP OSB
    ```

    For UNIX or LINUX:

    ```
    rda.sh -vSCRP OSB
    ```

2. Enter information as prompted on the command line. When asked whether you want RDA to collect Oracle Service Bus information, accept the default (Y).

    > **Note:** For more information about running the RDA tool and the options, see the readme file in the RDA directory in your Fusion Middleware home directory.

3. You can display the results in your web browser. Access the file from the output directory you specified.

    The name of the file is *prefix*__start.htm, where *prefix* is the prefix you specified.