

Oracle® Service Bus

Oracle BPEL Process Manager Transport User Guide
10g Release 3 (10.3)

October 2008

ORACLE®

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

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

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

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

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

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

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

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

Contents

Using Oracle BPEL Process Manager with Oracle Service Bus

Before You Begin	1
Overview	2
SOAP Support.	2
Transaction Propagation	3
SSL Support	3
Environment Variables	3
Simple Use Cases (Synchronous)	3
Synchronous: Invoking Processes in Oracle BPEL Process Manager	4
Creating and Configuring the Services	4
Synchronous: Calling External Services from Oracle BPEL Process Manager.	5
Creating and Configuring the Services	5
Associating Messages with the Correct Conversation	6
Advanced Use Cases (Asynchronous)	6
Asynchronous: Invoking Processes in Oracle BPEL Process Manager	6
Creating and Configuring the Services	7
Asynchronous: Calling Service Providers from Oracle BPEL Process Manager.	8
Creating and Configuring the Services	9
Transport Configuration Reference	10
Endpoint URI	11
bpel-10g Transport Configuration	12
Security	14

Using SSL from Oracle Service Bus to Oracle Servers	15
Error Handling	15
Application Errors	15
Connection Errors.	15
Other Errors	16
WS-Addressing Reference.	16
ReplyTo	16
Calling a BPEL Process Asynchronously Through Oracle Service Bus	16
BPEL Processes Calling External Services Through Oracle Service Bus	16
MessageID / RelatesTo.	17
XML Examples	17
Conversation ID Examples.	17
Port and Message Definitions	18
WS-Addressing that Sets the Conversation ID	19
Message Payload Data that Sets the Conversation ID	20
Transformation Examples	23
Asynchronous BPEL to BPEL Through Oracle Service Bus Example	25
Port and Message Definitions	25
BP1 to P1 – Initiate operation	26
P1/B1 to BP2.	26
BP2 to P2 – onResult operation	27
P2/B2 to BP1 – onResult operation.	28

Using Oracle BPEL Process Manager with Oracle Service Bus

This document describes how to bring Oracle BPEL Process Manager into your service oriented architecture (SOA) environment using Oracle Service Bus. It contains the following sections:

- [Before You Begin](#)
- [Overview](#)
- [Simple Use Cases \(Synchronous\)](#)
- [Advanced Use Cases \(Asynchronous\)](#)
- [Transport Configuration Reference](#)
- [Security](#)
- [Error Handling](#)
- [WS-Addressing Reference](#)
- [XML Examples](#)

Before You Begin

Before you begin building services in Oracle Service Bus that work with Oracle BPEL Process Manager, be sure to review the [Product Support Information](#) to make sure you are using the supported configurations for BPEL interoperability.

Overview

Oracle Service Bus provides support for communication with Oracle BPEL Process Manager, letting you include BPEL processes in your service oriented architecture (SOA). With Oracle Service Bus's native transport for Oracle BPEL Process Manager (BPEL transport), you can expose BPEL processes as Web services in the service bus layer, letting other services invoke BPEL processes.

Likewise, Oracle BPEL Process Manager can call services in the service bus layer for use in its own processes.

While this guide focuses on built-in integration between Oracle Service Bus and Oracle BPEL Process Manager—integration enabled at the Oracle Service Bus API level, you are not limited to only the solutions provided in this guide. You may also use other standard communication protocols provided with Oracle Service Bus, such as HTTP, JMS, and File.

For detailed information on Oracle BPEL Process Manager, go to <http://www.oracle.com/technology/products/ias/bpel/index.html>.

SOAP Support

Communication between Oracle BPEL Process Manager and Oracle Service Bus is done over SOAP only.

Oracle Service Bus and Oracle BPEL Process Manager do not provide full support for SOAP RPC encoding. The BPEL transport accepts SOAP RPC encoding bindings, but some encoding mechanisms, such as multiRef, may cause run time failures.

The BPEL transport supports the following features:

- SOAP 1.1. SOAP 1.2 is supported only from Oracle Service Bus to Oracle BPEL Process Manager using synchronous communication.
- SOAP headers

The BPEL transport has the following restrictions:

- No attachments
- No WS-Security or WS-RM

Transaction Propagation

Oracle BPEL Process Manager supports transaction propagation through its API, and the BPEL transport is transactional to support transaction propagation when Oracle BPEL Process Manager is deployed on WebLogic Server. For example, if a process begins in a service outside of Oracle BPEL Process Manager, Oracle Service Bus can propagate the transaction to Oracle BPEL Process Manager through the BPEL transport to complete the transaction.

Transaction propagation is also supported from BPEL Process Manager to Oracle Service Bus through an SB transport-based proxy service.

Note: Transaction propagation is not yet supported for Oracle servers OC4J and Oracle AS and not yet certified on IBM WebSphere.

SSL Support

Because calls from Oracle Service Bus to Oracle BPEL Process Manager are made using RMI, the BPEL transport supports security at the call level through one-way SSL. For more information, see the [Endpoint URI](#) section.

Environment Variables

The BPEL transport declares the following environment values, which can be maintained when moving an Oracle Service Bus configuration among different deployment environments:

Environment Variable	Description
Dispatch Policy	The WebLogic Server Work Manager instance used for the service endpoint's dispatch policy.
Service Account	For JNDI context security; the service account used to access the Oracle BPEL Process Manager delivery service.
Endpoint URI	The URI of the service.

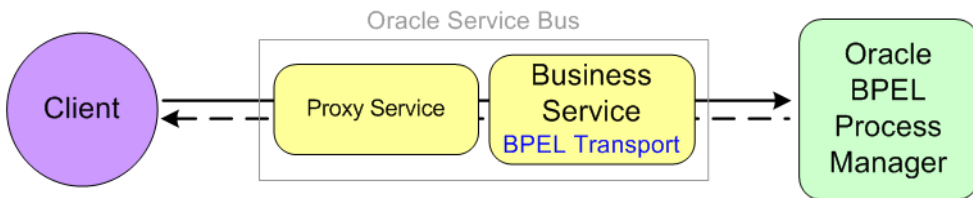
Simple Use Cases (Synchronous)

This section describes the simple, most common use cases for communicating to and from Oracle BPEL Process Manager through Oracle Service Bus: synchronous one-way or request/response communication.

Synchronous: Invoking Processes in Oracle BPEL Process Manager

Figure 1 illustrates a synchronous communication pattern between a client and Oracle BPEL Process Manager through Oracle Service Bus.

Figure 1 Invoking Oracle BPEL processes synchronously through Oracle Service Bus



Creating and Configuring the Services

Use the following guidelines to invoke Oracle BPEL Process Manager processes from a client:

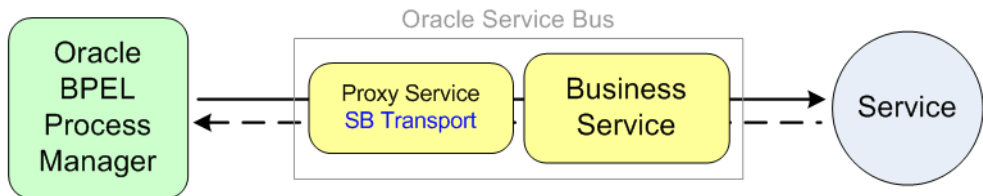
- Create a Business Service in Oracle Service Bus that represents the BPEL process you want to invoke.
 - Create a WSDL-based business service. Generate the WSDL from Oracle BPEL Process Manager.
 - Select the **bpel-10g** transport in the business service configuration.
 - Set the Endpoint URI as described in [Table 2](#).
 - Configure the remainder of the business service. See the [Transport Configuration Reference](#).
- Create a Proxy Service in Oracle Service Bus that invokes the business service. See one of the following topics for proxy configuration details:
 - Oracle Service Bus Console Help – [Proxy Services: Creating and Managing](#)
 - Workshop for WebLogic Help – [Working with Proxy Services](#)

To ensure that messages are associated with the correct conversation, see [Associating Messages with the Correct Conversation](#).

Synchronous: Calling External Services from Oracle BPEL Process Manager

Figure 2 illustrates a synchronous communication pattern between Oracle BPEL Process Manager and a service provider through Oracle Service Bus.

Figure 2 Oracle BPEL processes invoking a service synchronously through Oracle Service Bus



Creating and Configuring the Services

Use the following guidelines to invoke an external service from Oracle BPEL Process Manager:

- Create a Business Service in Oracle Service Bus that represents the external service you want to invoke. See one of the following topics for proxy configuration details:
 - Oracle Service Bus Console Help – [Business Services: Creating and Managing](#)
 - Workshop for WebLogic Help – [Working with Business Services](#)
- Create a Proxy Service in Oracle Service Bus that invokes the business service.
 - You must create the proxy with a SOAP WSDL to invoke the business service. When defining your proxy service, for the Service Type select **WSDL Web Service**, and select the desired port or binding.
 - Select the **sb** transport in the proxy service configuration.
 - To invoke the proxy service from Oracle BPEL Process Manager, export the proxy service's effective WSDL and import it into your Oracle BPEL Process Manager development environment. Invoke the proxy service from Oracle BPEL Process Manager as you normally would.

See one of the following topics for proxy configuration details:

- Oracle Service Bus Console Help – [Proxy Services: Creating and Managing](#)
- Workshop for WebLogic Help – [Working with Proxy Services](#)
- [SB Transport User Guide](#)

To ensure that messages are associated with the correct conversation, see [Associating Messages with the Correct Conversation](#).

Associating Messages with the Correct Conversation

When using stateful services, the messages sent synchronously between Oracle Service Bus and Oracle BPEL Process Manager are known as a conversation. Oracle BPEL Process Manager supports the following mechanisms for ensuring that messages are correctly associated with each other as part of a conversation. These mechanisms are independent of each other, and you may choose to use both to ensure correct association.

- **BPEL Correlation** – BPEL correlation is part of the BPEL specification. When a WSDL-based business service in Oracle Service Bus sends a message to a BPEL process, the BPEL engine examines the message to find the target BPEL process instance.
- **Opaque Correlation using WS-Addressing** – When a conversation is initiated by a client through Oracle Service Bus to a BPEL process, the BPEL engine looks in the WS-Addressing SOAP header for the “messageID” value to use as the ID for the new conversation. The conversation ID is carried through the conversation as the “RelatesTo” value.

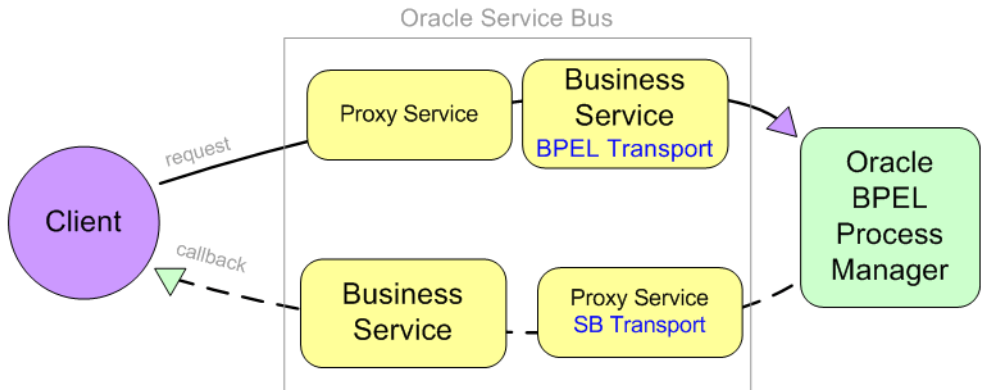
For more information on WS-Addressing, see the [MessageID / RelatesTo](#) section in the WS-Addressing reference. For an example of conversation ID setting, see the [Conversation ID Examples](#).

Advanced Use Cases (Asynchronous)

This section describes more advanced use cases for communicating to and from Oracle BPEL Process Manager through Oracle Service Bus using asynchronous communication.

Asynchronous: Invoking Processes in Oracle BPEL Process Manager

[Figure 3](#) illustrates an asynchronous communication pattern between Oracle BPEL Process Manager and a service provider through Oracle Service Bus.

Figure 3 Invoking Oracle BPEL processes asynchronously through Oracle Service Bus

In an asynchronous message exchange, a callback is sent on a different connection than the request.

Creating and Configuring the Services

Use the following guidelines to invoke Oracle BPEL Process Manager processes asynchronously from a client through Oracle Service Bus:

- Create two proxy services in Oracle Service Bus: one that invokes the business service and another that handles the callback.

Request Proxy Service

- Since the callback will be sent on a different connection in asynchronous communication, you must establish the callback address in the request proxy. This callback address will be passed to the callback proxy and callback business services so that the message is sent back to the correct client.

As part of the business service configuration, you select a Callback Proxy. At run time, the BPEL transport uses this proxy as the callback proxy.

For approaches to setting a callback address if you do not select a callback proxy in the business service, see the [WS-Addressing Reference](#) and the [Asynchronous BPEL to BPEL Through Oracle Service Bus Example](#).

Callback Proxy Service

- Configure the proxy to use a Service Type of WSDL SOAP or Any SOAP Service and the SB or HTTP transport. Use the SB transport if you want transaction propagation from Oracle BPEL Process Manager to Oracle Service Bus.

If you select this proxy service as the business service's callback proxy, the BPEL transport provides the correct callback URI at run time.

See one of the following topics for proxy configuration details:

- Oracle Service Bus Console Help – [Proxy Services: Creating and Managing](#)
- Workshop for WebLogic Help – [Working with Proxy Services](#)
- Create two business services in Oracle Service Bus: one that makes the request to the Oracle BPEL Process Manager process you want to interact with and another that handles the callback.

Request Business Service

- Create a WSDL-based business service. Generate the WSDL from Oracle BPEL Process Manager. Select a Service Type of **WSDL Web Service**, and select the appropriate binding or port in the WSDL.
- Select the **bpel-10g** transport in the business service configuration.
- Set the role to **Asynchronous Client**.
- Set the Endpoint URI, described in [Table 1](#).
- Use the Callback Proxy field on the bpel-10g transport configuration page to select the callback proxy you created.

Callback Business Service

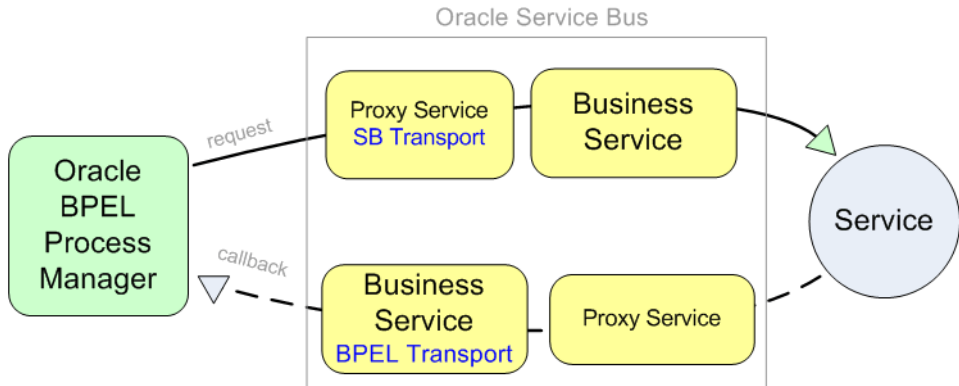
Configure the business process you need to handle the callback.

See the [Transport Configuration Reference](#). See also the [Business Service Transport Configuration page](#) reference for configuration information not covered in this document.

Asynchronous: Calling Service Providers from Oracle BPEL Process Manager

This section describes the steps and configuration needed for Oracle BPEL Process Manager to make service calls through Oracle Service Bus.

[Figure 4](#) illustrates an asynchronous communication pattern between Oracle BPEL Process Manager and a service provider through Oracle Service Bus.

Figure 4 Oracle BPEL processes invoking a service asynchronously through Oracle Service Bus

In an asynchronous message exchange, a callback is sent on a different connection than the request.

Creating and Configuring the Services

Use the following guidelines to invoke an external service asynchronously from Oracle BPEL Process Manager through Oracle Service Bus.

- Create two proxy services in Oracle Service Bus: one for the request that invokes the business service and another that handles the callback.

Request Proxy Service

- Configure the proxy service to use the **sb** transport.
- Since the callback will be sent on a different connection in asynchronous communication, you must establish a callback address so that the message is sent back to the correct client.

For information on setting a callback address, see the [ReplyTo](#) section of the WS-Addressing reference and the [Asynchronous BPEL to BPEL Through Oracle Service Bus Example](#).

Callback Proxy Service

- Configure the proxy service to pass the callback address to the business service. The callback URI is provided in the request. Use URI rewriting to extract the callback URI and forward it to the business service.

See one of the following topics for proxy configuration details:

- Oracle Service Bus Console Help – [Proxy Services: Creating and Managing](#)
- Workshop for WebLogic Help – [Working with Proxy Services](#)
- [SB Transport User Guide](#)
- Create two business services in Oracle Service Bus: a request business service that invokes the external service and a callback business service.

Request Business Service

- Configure the business service to invoke the external service.

Callback Business Service

- The callback business service receives the callback address from the callback proxy. The URI rewriting performed by the callback proxy service determines which BPEL process to send the response to.

Create a WSDL-based business service. Generate the WSDL from Oracle BPEL Process Manager. Select a Service Type of **WSDL Web Service**, and select the appropriate binding or port in the WSDL

- Select the **bpel-10g** transport in the business service configuration.
- Set the Endpoint URI to **bpel://callback**, as described in [Table 1](#). The callback URI is provided by the callback proxy service.

Note: If the callback address is always known, for example when the client and BPEL service are linked together because of a trading partner agreement, you can provide the exact callback address in the callback business service instead of using `bpel://callback`.

- Set the role to **Service Callback** on the `bpel-10g` transport configuration page, as described in [Table 2](#).

Configure the remainder of the business service. See the [Transport Configuration Reference](#). See also the [Business Service Transport Configuration page](#) reference for configuration information not covered in this document.

Transport Configuration Reference

This section provides descriptions for BPEL transport-specific configuration options. For descriptions of other business service configuration options, see one of the following topics:

- Oracle Service Bus Console Help – [Business Services: Creating and Managing](#)

- Workshop for WebLogic Help – [Working with Business Services](#)

Endpoint URI

[Table 1](#) describes the URI format for the BPEL transport, which you configure on the main Transport page of the business service in either Workshop for WebLogic or in the Oracle Service Bus console.

Table 1 Specifying an Endpoint URI

Transport used as	Endpoint URI format
Synchronous Client or Asynchronous Client role	<p>For the following endpoint URI format, optional elements are shown in square brackets. Descriptions follow.</p> <p>protocol://host[:port][/protocol-path]/domain/process[/version[/partnerlink/role]]</p> <ul style="list-style-type: none"> • protocol Use one of the following RMI / JNDI protocols: <ul style="list-style-type: none"> – iiop / iiops – For generic, server-agnostic use. – t3 / t3s – For use with WebLogic Server. – http / https – For tunneling and use with WebLogic Server. – ormi / ormis – For stand-alone deployment on OC4J (Oracle Container for JEE). – opmn / opmns – For use with Oracle Application Server. • port – Optional, for the ormi and opmn protocols only, if the server is configured to use the default port. • protocol-path – For use only with the opmn / opmns protocol. The protocol-path is the server instance in a cluster. • version – Optional. The version of the process to invoke. • partnerlink/role – Optional. Include for a full path description when you specify version. <p>Cluster example</p> <p>You can also use the following URI format for targeting specific nodes in a cluster:</p> <p>protocol://host1:port1,host2:port2/<remainder_of_URI></p>

Table 1 Specifying an Endpoint URI

Transport used as	Endpoint URI format
Service Callback	<p><code>bpel://callback</code></p> <p>If the callback address is always known, for example when the client and BPEL service are linked together because of a trading partner agreement, you can provide the exact callback address for the Endpoint URI instead of using <code>bpel://callback</code>.</p> <p>See the description of the Service Callback role in Table 2.</p>

bpel-10g Transport Configuration

[Table 2](#) describes the options available on the BPEL transport configuration page in either Workshop for WebLogic or in the Oracle Service Bus console.

Table 2 BPEL transport configuration

Property	Description
Role	<p>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:</p> <ul style="list-style-type: none"> 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 (described in Table 1) and/or 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 the guidelines in Advanced Use Cases (Asynchronous). Service Callback Use this role if the business service is designed to be a service callback to Oracle BPEL Process Manager (where Oracle BPEL Process Manager is calling an service provider asynchronously through Oracle Service Bus). A Service Callback business service does not support load balancing or failover. For instructions on using Service Callback, see “Service Callback” in Table 1 and Asynchronous: Calling Service Providers from Oracle BPEL Process Manager.
Callback Proxy	<p>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 WSDL SOAP or Any SOAP) that will be used to route callbacks to the Oracle Service Bus client that made the request. For more information, see the guidelines in Advanced Use Cases (Asynchronous).</p>
Service Account	<p>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.</p> <p>There is no restriction on the type of service account that can be configured, such as static or pass-through, but the run time must be able to access a user name and password.</p>

Table 2 BPEL transport configuration (Continued)

Property	Description
Suspend Transaction	<p>Selecting Suspend Transaction makes the business service non-transactional even if the business service is invoked by a transaction.</p> <p>If you do not select Suspend Transaction:</p> <ul style="list-style-type: none"> • 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. <p>The BPEL transport ignores the Suspend Transaction option in the following situations:</p> <ul style="list-style-type: none"> • 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. <p>For a description of transaction propagation, see Transaction Propagation.</p>
Dispatch Policy	<p>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.</p> <p>For information about Work Managers, see the following WebLogic Server Administration Console Online Help topics:</p> <ul style="list-style-type: none"> • “Using Work Managers to Optimize Scheduled Work” at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/config_wls/self_tuned.html • “Create Global Work Managers” at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/ConsoleHelp/taskhelp/work/CreateGlobalWorkManager.html

Security

Security in Oracle BPEL Process Manager is handled at the EJB level.

The BPEL transport supports a JNDI “Service Account” option used for the creation of the JNDI context and the EJB lookup, as described in [Table 2](#). The BPEL transport gets the user name and password. If the service account is not configured, an anonymous subject is assumed.

The BPEL transport also supports security at the call level by letting you indicate one-way SSL in the protocol portion of the URI from Oracle Service Bus to Oracle BPEL Process Manager. For more information, see the [Endpoint URI](#) section.

Using SSL from Oracle Service Bus to Oracle Servers

To use SSL from Oracle Service Bus to OC4J and Oracle AS servers (using the ormis and opmns protocols), you must configure SSL on your Oracle Service Bus server by adding the following properties to your domain's `setDomainEnv` script:

- Djavax.net.ssl.trustStorePassword=passphrase
- Djavax.net.ssl.trustStore=<file path to a keystore of trust certificate>

Error Handling

The BPEL transport handles Oracle BPEL Process Manager exceptions in different ways.

Application Errors

If a BPEL process replies with a fault, the BPEL transport intercepts the fault message at the API and translates it into a SOAP fault.

The transport's automated application error-handling functionality lets you decide whether or not to automatically retry application errors—in this case BPEL faults—when they occur. For example, you may determine that application errors will always fail until the problem is fixed. Use the **Retry Application Errors** option on the transport configuration page to turn application retries on and off.

Connection Errors

Oracle BPEL Process Manager can throw the following non-fault exceptions, which the BPEL transport automatically categorizes as `TransportException` connection errors:

- `NamingException`
- `RemoteException`

The transport's automated connection error-handling functionality lets you determine if and how often connection errors are retried. Use the **Retry Count** and the **Retry Iteration Interval** options on the transport configuration page to control the number of retries and the number of

seconds to wait between retries on connection errors. Setting Retry Iteration Interval to zero (0) means connection errors are not retried.

Other Errors

Other non-application and non-connection exceptions are re-thrown as a generic `TransportException` error.

WS-Addressing Reference

This section describes specific WS-Addressing properties that the BPEL transport uses to communicate with Oracle BPEL Process Manager. This section also describes ways to provide callback addresses in asynchronous communication, as described in the [Advanced Use Cases \(Asynchronous\)](#) guidelines.

See the [XML Examples](#) for WS-Addressing examples.

ReplyTo

In asynchronous communication, a service callback is sent on a different connection than the request. As a service developer, you must supply the correct callback address in an asynchronous exchange so that the callback is sent to the correct client.

Calling a BPEL Process Asynchronously Through Oracle Service Bus

The BPEL transport provides a built-in mechanism for providing the correct callback address. When you create a BPEL business service in Oracle Service Bus, you can select a Callback Proxy to handle the callback, and the BPEL transport automatically sets the correct callback address. You do not need to manually use “ReplyTo.”

For more information, see [Asynchronous: Invoking Processes in Oracle BPEL Process Manager](#)

BPEL Processes Calling External Services Through Oracle Service Bus

When calling an external service from Oracle BPEL Process Manager through Oracle Service Bus, you must manually set a callback address. To do this using WS-Addressing, in the request proxy service set the callback address as the “ReplyTo” value. The BPEL transport in the business service uses that URI as the callback address.

For more information, see [Asynchronous: Calling Service Providers from Oracle BPEL Process Manager](#).

MessageID / RelatesTo

“MessageID” and “RelatesTo” are used to store the conversation ID in conversations between Oracle Service Bus and Oracle BPEL Process Manager, making sure related messages remain in the same conversation.

The BPEL transport does not let you specify whether a given operation is a start or continue operation. Instead, the BPEL transport looks for the “MessageID” and “RelatesTo” properties and sets them accordingly.

The following describes how the BPEL transport uses “MessageID” and “RelatesTo” in synchronous and asynchronous conversations:

- **Synchronous conversation:** In the initial request, the “MessageID” determines the conversation ID. In the remaining communication, the BPEL transport provides the conversation ID as the RelatesTo value.

If there is no value assigned to “MessageID” or “RelatesTo,” the transport assumes either no conversation is occurring or that Oracle BPEL Process Manager is handling the correlation.

- **Asynchronous callbacks** - In the initial request, the “MessageID” determines the conversation ID. In the remaining communication, the BPEL transport provides the conversation ID as the “RelatesTo” value in the callback.

If there is no value assigned to “MessageID” or “RelatesTo,” the transport assumes either no conversation is occurring or that Oracle BPEL Process Manager is handling the correlation.

For more implementation on establishing a conversation ID to make sure messages participate in the correct conversation, see [Associating Messages with the Correct Conversation](#) and the [Conversation ID Examples](#).

XML Examples

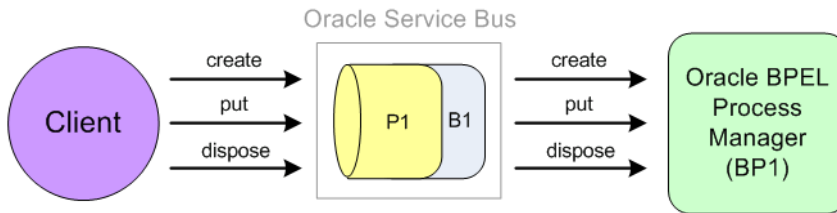
Following are examples of XML messaging issues between Oracle Service Bus and Oracle BPEL Process Manager.

Conversation ID Examples

This section provides different examples of establishing a conversation ID among messages in a conversation between Oracle Service Bus and Oracle BPEL Process Manager.

In [Figure 5](#), a client is synchronously invoking a process in Oracle BPEL Process Manager. The business service (B1) uses the BPEL transport to invoke a process. The proxy service (P1) handles any necessary conversation ID mapping.

Figure 5 Operations in a synchronous exchange through Oracle Service Bus



Port and Message Definitions

The examples in this section use the following port and message definitions defined in the WSDL.

```

<wsdl:types>
  <xsd:schema
    targetNamespace="http://www.sample.org/spec/samples/types"
    elementFormDefault="qualified">
    <xsd:complexType name="ValueHolder">
      <xsd:all>
        <xsd:any minOccurs="1"/>
      </xsd:all>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>

<message name="create"/>

<message name="putRequest">
  <part name="key" type="xsd:string"/>
  <part name="value" type="types:ValueHolder"/>
</message>

<message name="putResponse">
  <part name="value" type="types:ValueHolder"/>
</message>

...
<message name="dispose"/>

```

```

<portType name="ServiceMap">
  <operation name="create">
    <input message="tns:create"/>
  </operation>
  <operation name="put">
    <input message="tns:putRequest"/>
    <output message="tns:putResponse"/>
  </operation>
  ...
  <operation name="dispose">
    <input message="tns:dispose"/>
  </operation>
</portType>

```

WS-Addressing that Sets the Conversation ID

This example shows how WS-Addressing is used to set the conversation ID among messages in a conversation.

Figure 5 shows communication pattern.

Create Operation

```

<soap:Envelope>
  <soap:Header>
    <wsa03:MessageID>uuid:123456789</wsa03:MessageID>
  </soap:Header>
  <soap:Body>
    <create/>
  </soap:Body>
</soap:Envelope>

```

Put Operation

```

<soap:Envelope>
  <soap:Header>
    <wsa03:MessageID>uuid:111111111</wsa03:MessageID>
    <wsa03:RelatesTo>uuid:123456789</wsa03:RelatesTo>
  </soap:Header>
  <soap:Body>

```

```
<put>
  <key>key</key>
  <value>
    <PO/>
  </value>
</put>
</soap:Body>
</soap:Envelope>

<soap:Envelope>
  <soap:Body>
    <putResponse>
      <value/>
    </putResponse>
  </soap:Body>
</soap:Envelope>
```

The <put> operation also has a MessageID, but it is ignored because the RelatesTo has a value that provides the conversation ID.

Message Payload Data that Sets the Conversation ID

This example shows how message payload data can be used to set the conversation ID among messages in a conversation.

In these examples, the proxy service maps the ID to the MessageID / RelatesTo SOAP headers.

[Figure 5](#) shows communication pattern.

Create Operation

Client to proxy service

```
<soap:Envelope>
  <soap:Body>
    <create/>
  </soap:Body>
</soap:Envelope>

<soap:Envelope>
  <soap:Body>
    <createResponse>
      <mapID>uuid:123456789</mapID>
```



```

        </createResponse>
    </soap:Body>
</soap:Envelope>

```

Proxy service to BPEL process (via a business service)

```

<soap:Envelope>
  <soap:Header>
    <wsa03:MessageID>uuid:123456789</wsa03:MessageID>
  </soap:Header>
  <soap:Body>
    <create/>
  </soap:Body>
</soap:Envelope>

```

Not shown: The ID was generated in the request of the proxy service pipeline and inserted as a <wsa03:MessageID> before invoking the process. On the process side, the Create operation is one-way, so a SOAP response must be created before replying to the client. The response sends back the ID that was generated by the proxy service.

Put Operation

Client to proxy service

```

<soap:Envelope>
  <soap:Body>
    <put>
      <mapID>uuid:123456789</mapID>
      <key>key</key>
      <value>
        <PO/>
      </value>
    </put>
  </soap:Body>
</soap:Envelope>

<soap:Envelope>
  <soap:Body>
    <putResponse>
      <value/>
    </putResponse>

```

```
    </soap:Body>
</soap:Envelope>
```

Proxy service to BPEL process (via a business service)

```
<soap:Envelope>
  <soap:Header>
    <wsa03:RelatesTo>uuid:123456789</wsa03:RelatesTo>
  </soap:Header>
  <soap:Body>
    <put>
      <key>key</key>
      <value>
        <PO/>
      </value>
    </put>
  </soap:Body>
</soap:Envelope>

<soap:Envelope>
  <soap:Body>
    <putResponse>
      <value/>
    </putResponse>
  </soap:Body>
</soap:Envelope>
```

Dispose Operation

Client to proxy service

```
<soap:Envelope>
  <soap:Body>
    <dispose>
      <mapID>uuid:123456789</mapID>
    </dispose>
  </soap:Body>
</soap:Envelope>
```

Proxy service to BPEL process (via a business service)

```

<soap:Envelope>
  <soap:Header>
    <wsa03:RelatesTo>uuid:123456789</wsa03:RelatesTo>
  </soap:Header>
  <soap:Body>
    <dispose/>
  </soap:Body>
</soap:Envelope>

```

Transformation Examples

In these examples, the client uses a more recent version of the WS-Addressing spec (wsa04 prefix). The proxy service is responsible for transforming the SOAP headers to use the wsa03 prefix. The proxy service developer configures the transformation.

[Figure 5](#) shows communication pattern.

Create Operation

Client to proxy service

```

<soap:Envelope>
  <soap:Header>
    <wsa04:MessageID>uuid:123456789</wsa04:MessageID>
  </soap:Header>
  <soap:Body>
    <create/>
  </soap:Body>
</soap:Envelope>

```

Proxy service to BPEL process (via a business service)

```

<soap:Envelope>
  <soap:Header>
    <wsa03:MessageID>uuid:123456789</wsa03:MessageID>
  </soap:Header>
  <soap:Body>
    <create/>
  </soap:Body>
</soap:Envelope>

```

Put Operation

Client to proxy service

```
<soap:Envelope>
  <soap:Header>
    <wsa04:MessageID>uuid:111111111</wsa04:MessageID>
    <wsa04:RelatesTo>uuid:123456789</wsa04:RelatesTo>
  </soap:Header>
  <soap:Body>
    <put>
      <key>key</key>
      <value>
        <PO/>
      </value>
    </put>
  </soap:Body>
</soap:Envelope>

<soap:Envelope>
  <soap:Body>
    <putResponse>
      <value/>
    </putResponse>
  </soap:Body>
</soap:Envelope>
```

Proxy service to BPEL process (via a business service)

```
<soap:Envelope>
  <soap:Header>
    <wsa03:MessageID>uuid:111111111</wsa03:MessageID>
    <wsa03:RelatesTo>uuid:123456789</wsa03:RelatesTo>
  </soap:Header>
  <soap:Body>
    <put>
      <key>key</key>
      <value>
        <PO/>
      </value>
    </put>
  </soap:Body>
</soap:Envelope>
```

```

    </soap:Body>
  </soap:Envelope>

  <soap:Envelope>
    <soap:Body>
      <putResponse>
        <value/>
      </putResponse>
    </soap:Body>
  </soap:Envelope>

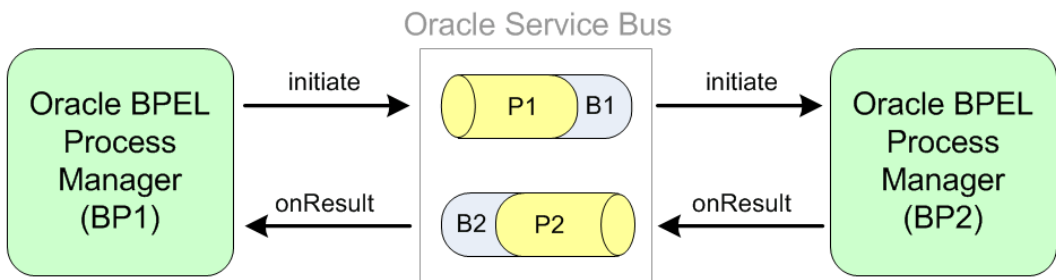
```

Asynchronous BPEL to BPEL Through Oracle Service Bus Example

The following example shows the SOAP headers involved in one BPEL process invoking another BPEL process asynchronously through Oracle Service Bus.

In [Figure 6](#), P1 and P2 are proxy services that pass messages (and perform transformations) to B1 and B2 business services, which are required to make calls to BPEL processes using the Oracle Service Bus BPEL transport.

Figure 6 One BPEL process invoking another BPEL process through Oracle Service Bus



Refer to [Figure 6](#) for the following SOAP header examples.

Port and Message Definitions

```

<message name="LoanServiceRequestMessage">
  <part name="payload" element="types:loanApplication"/>
</message>

```

```
<message name="LoanServiceResultMessage">
  <part name="payload" element="types:loanOffer"/>
</message>

<portType name="LoanService">
  <operation name="initiate">
    <input message="tns:LoanServiceRequestMessage"/>
  </operation>
</portType>

<portType name="LoanServiceCallback">
  <operation name="onResult">
    <input message="tns:LoanServiceResultMessage"/>
  </operation>
</portType>
```

BP1 to P1 – Initiate operation

```
<soap:Envelope>
  <soap:Header>
    <wsa03:ReplyTo>
      <wsa03:Address>
ormi://serverB:7001/default/AmericanLoanClient/1.0/service/LoanServiceRequester
      </wsa03:Address>
    </wsa03:ReplyTo>
    <MessageID>AmericanLoanClient~1.0/60007</MessageID>
  </soap:Header>
  <soap:Body >
    <loanApplication>
      ...
    </loanApplication>
  </soap:Body>
</soap:Envelope>
```

P1/B1 to BP2

```
<soap:Envelope>
  <soap:Header>
    <wsa03:ReplyTo>
      <wsa03:Address>http://serverB:7001/P2</wsa03:Address>
    </wsa03:ReplyTo>
  </soap:Header>
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>
```

```

        <wsa03:ReferenceProperties>
            <osb:Callback>
                <osb:Address>
ormi//localhost/default/AmericanLoanClient/1.0/service/LoanServiceRequester
                </osb:Address>
            </osb:Callback>
        </wsa03:ReferenceProperties>
    </wsa03:ReplyTo>
    <MessageID>AmericanLoanClient~1.0/60007</MessageID>
</soap:Header>
<soap:Body >
    <loanApplication>
        ...
    </loanApplication>
</soap:Body>
</soap:Envelope>

```

The ReplyTo callback address is set by B1, which gets the value from the Callback Proxy field in the BPEL transport configuration, as described in [Table 2](#). B1's callback proxy is P2.

You must wrap the original replyTo information and send it as reference properties so that it is echoed back in the onResult callback message (to follow).

Note: This sample uses osb:Callback and osb:Address for illustration purpose only. There is no standard or Oracle Service Bus standard elements defined for WS-Addressing support.

BP2 to P2 – onResult operation

```

<soap:Envelope>
    <soap:Header>
        <wsa03:RelatesTo>AmericanLoanClient~1.0/60007</wsa03:RelatesTo>
        <osb:Callback>
            <osb:Address>
ormi//localhost/default/AmericanLoanClient/1.0/service/LoanServiceRequester
            </osb:Address>
        </osb:Callback>
    </soap:Header>
    <soap:Body >
        <loanOffer>
            ...

```

```
        </loanOffer>
    </soap:Body>
</soap:Envelope>
```

The reference property `osb:Callback` is sent back as a SOAP header by the Oracle BPEL Process Manager engine.

P2/B2 to BP1 – onResult operation

```
<soap:Envelope>
  <soap:Header>
    <wsa03:RelatesTo>AmericanLoanClient~1.0/60007</wsa03:RelatesTo>
  </soap:Header>
  <soap:Body >
    <loanOffer>
      ...
    </loanOffer>
  </soap:Body>
</soap:Envelope>
```

Proxy service P2 removes the temporary `osb:Callback` header; but prior to deleting this header, the `replyTo` address value is copied to the `$outbound` variable so that the BPEL transport in business service B2 can send the callback message to the correct BPEL process.