

Oracle® Service Bus

Interoperability Solution for Tuxedo
10g Release 3 (10.3)

October 2008

ORACLE®

Oracle Service Bus Interoperability Solution for Tuxedo, 10g Release 3 (10.3)

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

Introduction to Tuxedo Transport

Capabilities of the Tuxedo transport	1-2
--	-----

Configuring WebLogic Tuxedo Connector for Tuxedo transport

Before You Begin	2-2
Configuring WebLogic Tuxedo Connector	2-2

Using Tuxedo Services from Oracle Service Bus

Configuring a New Business Service.	3-1
Add a New Project	3-2
Add a Business Service.	3-2
Load Balancing and Failover.	3-7
Handling Errors	3-7
Testing Your Configuration	3-8

Using Oracle Service Bus from Tuxedo

Adding and Configuring a Proxy Service	4-1
Add a New Project	4-2
Add a Proxy Service	4-2
Configure the Proxy Service.	4-4
Testing Your Configuration	4-6

Tuxedo Transport Buffer Transformation

Any XML Service Type.	5-2
----------------------------	-----

Messaging Service Type 5-3

Tuxedo Transport Transaction Processing

Inbound Services 6-2

Outbound Services 6-2

Introduction to Tuxedo Transport

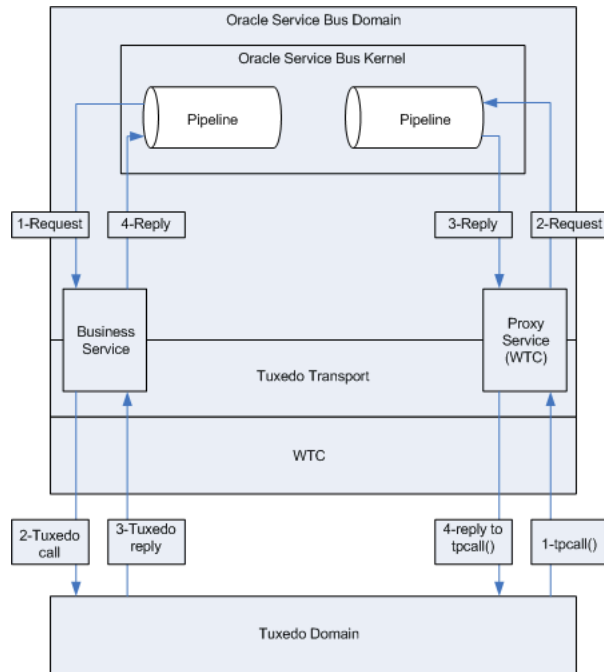
Oracle Service Bus and Oracle Tuxedo can interoperate to use the services that each product offers. The Tuxedo transport allows secure, guaranteed, high performance, bi-directional access to a Tuxedo domain from Oracle Service Bus. The Tuxedo transport allows Tuxedo domains to call services as well as have services called in a Tuxedo domain.

- When Oracle Service Bus uses services offered by Tuxedo, the Tuxedo transport facilitates access to those Tuxedo services. The term *outbound* refers to this business service scenario.
- When Tuxedo uses services offered by Oracle Service Bus, Tuxedo services can call Oracle Service Bus services as though they were another Tuxedo application. The term *inbound* refers to this proxy service scenario.

You configure the Tuxedo transport using the Oracle Service Bus Console. Specific parameters provide definitions for both proxy and business services. A basic WebLogic Tuxedo Connector (WTC) configuration with one local access point and one remote access point is required to enable configuration of the Tuxedo transport. Support for transactional and security contexts are available as well.

The following diagram summarizes this message handling processes.

Figure 1-1 WTC Message Handling



Capabilities of the Tuxedo transport

The following capabilities are available in the native Tuxedo transport in Oracle Service Bus.

- First-class tier transport

The native Tuxedo transport is fully integrated into the Oracle Service Bus configuration, management, and monitoring console. You can configure, manage, and monitor both Tuxedo proxy services and Tuxedo businesses services.

- Bi-directional access

Oracle Service Bus is an intermediary between SOAP, JMS, or other services and Tuxedo. The Tuxedo transport enables access to Tuxedo ATMI services as business services in Oracle Service Bus and allows Oracle Service Bus proxy services to be seen by Tuxedo as another ATMI service.

- Buffer transformation

You can transform XML messages to Tuxedo buffer types and Tuxedo buffer types to XML. All standard Tuxedo buffer types are supported; transformation is automatic and transparent. For more information, see [“Tuxedo Transport Buffer Transformation” on page 5-1](#).

- Transactional integrity

Tuxedo transport provides transactional integrity for inbound and outbound messages. You can call Tuxedo services in the context of a global transaction allowing Exactly Once quality of service (QoS). A Tuxedo application can start a transaction, call an Oracle Service Bus-based service and the XA transaction context is carried through to Oracle Service Bus, through the pipeline, and finally to the destination transport. For more information, see [“Tuxedo Transport Transaction Processing” on page 6-1](#).

- Security propagation

The security context established at the beginning of the message flow, from either a Tuxedo client or an Oracle Service Bus client, is propagated to the other system. This means that an incoming SOAP over HTTP request to Oracle Service Bus that requires authentication is authenticated by Oracle Service Bus. As with transactions, this support is fully bi-directional, so a client authenticated to Tuxedo can make requests to Oracle Service Bus services without requiring authentication a second time.

- Encrypted network links

You can encrypt the connections between Oracle Service Bus and Tuxedo through WTC configuration to ensure the security and privacy of communications between the two systems.

- Load balancing

A single network connection is the only requirement to connect Oracle Service Bus to a Tuxedo domain. However, it might be necessary to support multiple connections in case of a machine or network failure. You can make multiple connections to a single domain or multiple domains for purposes of load balancing.

See the following sections for detailed information on configurations for the following scenarios:

- [Configuring WebLogic Tuxedo Connector for Tuxedo transport](#)
- [Using Tuxedo Services from Oracle Service Bus](#)
- [Using Oracle Service Bus from Tuxedo](#)

You can use the following related documentation to learn more about this environment:

- For information on setting up your Oracle Service Bus environment, see [Using the Oracle Service Bus Console](#).
- For more information on setting up your Tuxedo environment, see [Setting Up a Tuxedo Application](#) in the Tuxedo documentation.
- For information about the WebLogic Tuxedo Connector, see [WebLogic Tuxedo Connector](#) in the WebLogic Server documentation.

Configuring WebLogic Tuxedo Connector for Tuxedo transport

Oracle Service Bus Tuxedo transport enables access to Tuxedo services using WebLogic Tuxedo Connector (WTC). To use the Tuxedo transport, you must configure a basic WTC server including one local access point and one remote access point.

The following sections describe how to configure WTC:

- [Before You Begin](#)
- [Configuring WebLogic Tuxedo Connector](#)

Before You Begin

Gather the following information about the Tuxedo application that Oracle Service Bus will use:

- ID of the Tuxedo local access point.
- Network address of the Tuxedo local access point.
- Name of the exported Tuxedo service.
- Whether the service needs XML-to-FML and FML-to-XML conversion or VIEW-to-XML or XML-to-VIEW conversion.

The example described in the following sections assumes the use of FML/FML32 buffer types.

- ID of the access point that the Tuxedo domain gateway will use to refer to this WebLogic Tuxedo Connector instance. This is the same as the remote access point ID.
- Network address that the Tuxedo domain gateway has defined for this WebLogic Tuxedo Connector local access point. This is the same as the remote network address.

Configuring WebLogic Tuxedo Connector

When you create or import Tuxedo business and proxy services in Oracle Service Bus, the service configuration includes WebLogic Tuxedo Connector configurations that appear as WebLogic Tuxedo Connector resources in the Oracle WebLogic Server Console. Oracle Service Bus needs to keep the WebLogic Tuxedo Connector resources it uses in sync. Modifications to those WebLogic Tuxedo Connector resources in the Oracle WebLogic Server Console can cause those resources to become out of sync with Oracle Service Bus, and re-import of those services into Oracle Service Bus results in service activation failure.

Use the following guidelines for using and configuring WebLogic Tuxedo Connector resources in Oracle Service Bus:

- Do not modify WebLogic Tuxedo Connector resources in the Oracle WebLogic Server Console that you use in Oracle Service Bus proxy and business services. Modify the WebLogic Tuxedo Connector configuration in your Oracle Service Bus service configurations.
- If the WebLogic Tuxedo Connector configurations do become out of sync between your Tuxedo services in Oracle Service Bus and the Oracle WebLogic Server Console, the easiest way to get back in sync is to delete the WebLogic Tuxedo Connector resources in

the Oracle WebLogic Server Console and re-configure or re-import the Tuxedo services in Oracle Service Bus.

For additional information about the WebLogic Tuxedo Connector, see [WebLogic Tuxedo Connector](#) in the WebLogic Server documentation.

To use Tuxedo services from Oracle Service Bus, follow the instructions in [“Configuring a New Business Service” on page 3-1](#). To use Oracle Service Bus services from Tuxedo, see [“Adding and Configuring a Proxy Service” on page 4-1](#).

Configuring WebLogic Tuxedo Connector for Tuxedo transport

Using Tuxedo Services from Oracle Service Bus

The following sections describe how to use Tuxedo services from Oracle Service Bus:

- [Configuring a New Business Service](#)
- [Load Balancing and Failover](#)
- [Handling Errors](#)
- [Testing Your Configuration](#)

Configuring a New Business Service

To use Tuxedo services from Oracle Service Bus, you must configure a new business service in the Oracle Service Bus Console. For more information about business services, see [Business Services: Creating and Managing](#) in *Using the Oracle Service Bus Console*.

Log in to the Oracle Service Bus Console. Perform the configuration steps in the order presented, using the instructions in the following sections.

- [Add a New Project](#)
- [Add a Business Service](#)

For detailed procedural information, see [Using the Oracle Service Bus Console](#).

Add a New Project

Follow these steps:

1. In the Change Center, click **Create** to create a new session or click **Edit** to enter an existing session.

You must be in a session to edit resources.

2. Select **Project Explorer**.
3. Enter a name for the new project and click **Add Project**.

A message at the top of the page indicates that the project was added successfully.

Add a Business Service

Follow these steps:

1. Click the name of the newly created project.
2. From the **Create Resource** drop-down list, select **Business Service**.
3. On the **Create a Business Service – General Configuration** page, enter the following values:

Service Name – The name of the service.

Service Type – Select **Any XML Service** (the default).

Note: Tuxedo transport only supports **Any XML Service** and **Messaging Service** service types.

4. Click **Next**.
5. On the **Create a Business Service – Transport Configuration** page, enter the following values:

Protocol – Select **tuxedo**.

Load Balancing Algorithm – Keep the default or select another algorithm.

Endpoint URI – Specify one or more endpoint URIs, using one of the following formats, then click **Add**.

- The URI format for outbound calls to Tuxedo services is `tuxedo:resourcename[/remotename]` where,
 - `resourcename` corresponds to a WTC Import service name; `resourcename` is required.
 - `remotename` corresponds to the service name exported by the remote Tuxedo domain; `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 the remote and resource name are the same, only one URI is allowed. This allows already defined WTC Import services to use WTC load balancing and failover.

Note: If you configure two identical URIs, an error displays notifying you that the service name already exists.

The Tuxedo transport uses the resource name and remote name from the URI to dynamically create a WTC Import service.

- The URI format for outbound calls to Tuxedo resources of type /Q is `tuxedo-queue:sendQspace/sendQname[/rcvQspace:]/rcvQname[/failureQname]` where,
 - `tuxedo-queue` indicates that /Q calls will be made.
 - `sendQspace` corresponds to the unique name of the queue space in the Tuxedo domain; `sendQspace` is required.
 - `sendQname` corresponds to the queue name in the queue space in which requests will be stored; `sendQname` is required.

The following two values are optional. If you specify neither one, the run time will return immediately and not expect a response. Also, the **Response Required?** option on the **Tuxedo Transport Configuration** page will be unavailable.

If you specify either value and do not select the **Response Required?** option, the values you specified will be ignored.

- `rcvQspace` corresponds to the unique name of the queue space in the Tuxedo domain from which replies will be received; `rcvQspace` is optional. If not specified, the `sendQspace` value is used.
- `rcvQname` corresponds to the name of the queue in the Tuxedo domain from which replies will be received; `rcvQname` is optional.

The last value is also optional. If you specify neither `rcvQspace` nor `rcvQname`, but specify `failureQname`, the URI format is

`tuxedo-queue:sendQspace/sendQname//failureQname`

- `failureQname` corresponds to the name of the queue in the Tuxedo domain where error messages will be stored; `failureQname` is optional.

Note: When a response is expected, it occurs in the same thread that sends the request.

6. Click **Next**.

7. On the **Tuxedo Transport Configuration** page, enter the following values:

Field Table Classes – Optional. 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 – Optional. Enter the name of the class or classes describing the VIEW/VIEW32 buffer received or sent. These are used for the VIEW/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.

Remote Access Point – From the drop-down list, select a remote access point that is associated with the WTC Import service. The drop-down list contains remote access points configured in WTC. You cannot create a business service without an 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.

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.

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 – This field appears only when you select **New** in the **Remote Access Point** field. From the drop-down 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.

Note: Access points are not deleted by the transport when the endpoints are removed, since they may be used by multiple endpoints. To remove access points, use the WebLogic Server Administration Console.

Request Buffer Type – From the drop-down list, select the type of buffer that the remote Tuxedo service will receive.

Request Buffer Subtype – This field is enabled if the Request Buffer Type value is VIEW or VIEW32. Enter the buffer subtype with which to associate the request buffer.

Response Required? – Select the check box to indicate a bidirectional call. If not selected, the underlying `tpcall` is invoked with `TPNOREPLY` flag, and a null response is posted asynchronously.

Suspend Transaction – Select the check box to suspend the transaction, if one exists. This is useful when the remote service does not support transactions.

When making calls to Tuxedo resources of the type `/Q`, use the **Suspend Transaction** option whether or not you expect a reply. A successful return from a one-way call means that a message has been successfully queued.

Note: Tuxedo transports to `/Q` mode endpoints are considered asynchronous transactional, if the **Suspend Transaction** option is not selected. This allows the framework to prevent a deadlock situation.

In `/Q` mode, when an endpoint expects a reply, multiple threads on multiple Managed Servers may reply using the same destination. For this reason, at run time, when a reply is expected, a unique correlation ID is sent along with the request.

The dequeue operation then waits for the message containing that correlation ID. Correlation IDs are composed in the same manner as those used by JMS transports in similar situations.

Dispatch Policy – From the drop-down list, select a WebLogic Server Work Manager, if available. The default Work Manager is used if no other one exists. The Work Manager asynchronously posts a null reply in the case of a one-way invocation.

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 – The ordering or grouping of elements when FML or FML32 buffers are transformed into XML. Select one of the following choices:

None: (default) The order of fields may not be respected.

Ordered: The fields will be presented with all their occurrences in the correct order.

Ordered and Grouped: If the fields are logically structured as records, the fields will be ordered by occurrence and also grouped by record.

Timeout – Specify the maximum amount of time (in seconds) that the transport run time 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 Tuxedo access point allows for a blocking call.

At run time, replies exceeding the time-out value are ignored and an error message with a TPETIME exception is returned.

Specify the timeout value only for request/response services. For /Q or one-way endpoints, the **Timeout** field is unavailable. If the outbound call is part of a transaction, the timeout value is ignored.

Note: The WTC BLOCKTIME value takes precedence if it is less than the time-out value. For example, if the endpoint time-out value is 30 seconds and the WTC BLOCKTIME is 20 seconds, calls will time out in 20 seconds.

8. Click **Finish**.
9. On the **Summary** page, click **Save**.

Load Balancing and Failover

When specifying a business service and defining the endpoint URIs, entering a remote name that is different from the resource name allows you to use the Oracle Service Bus load balancing and failover capabilities. In this case, you can define multiple service names and associate them to a service that is replicated across multiple remote domains. The resource name must be unique, but remote names do not have the same restriction.

Handling Errors

You can configure tuxedo-transport business services to handle application and communication errors as follows:

- Application errors—you can specify whether or not to retry business service endpoint URIs when application errors occur. See Retry Application Errors in [Creating and Configuring Business Services - Transport Configuration page](#) in *Using the Oracle Service Bus Console*.
- Communication errors—you can configure business service URIs to be taken offline when communication errors occur. See Enable business service endpoint URIs to be taken offline in [Configuring Operational Settings for Business Services](#) and [Viewing Business Services Endpoint URIs Metrics](#) in *Using the Oracle Service Bus Console*.

[Table 3-1](#) describes the following Tuxedo exceptions and the type of Oracle Service Bus error they denote.

Table 3-1 Tuxedo Exceptions


Exception	Description
TPESVCFAIL	The service failed—an application error
TPENOENT	The requested entity does not exist—a communication error
TPEPERM	A permissions error has occurred—a communication error

For more information on Tuxedo exceptions, see the [Javadocs](#).

Testing Your Configuration

Once you have configured Oracle Service Bus to work with Tuxedo, you can test the configuration using the test console in the Oracle Service Bus Console.

The following list of tasks summarizes the process of testing outbound usage of Tuxedo by Oracle Service Bus.

1. Build and start the Tuxedo servers.
2. Set up a Tuxedo service to call the Oracle Service Bus proxy service.
3. In the Oracle Service Bus Console, click **Activate** under Change Center, to enable the test console.
4. In the **Project Explorer**, click the  **Launch Test Console** icon associated with the business service you want to test.
5. Enter a payload in the test console. For more information, see [Testing Business Services](#) in *Using the Oracle Service Bus Console*.
6. Click **Execute**.

A response page displays the results of the service request.

Using Oracle Service Bus from Tuxedo

The following sections describe how to use Oracle Service Bus services from Tuxedo:

- [Adding and Configuring a Proxy Service](#)
- [Testing Your Configuration](#)

Adding and Configuring a Proxy Service

To use Oracle Service Bus services from Tuxedo, you must configure a new proxy service using the Oracle Service Bus Console. For more information about proxy services, see [Proxy Services: Creating and Managing](#) in the *Using the Oracle Service Bus Console*.

Log in to the Oracle Service Bus Console and perform these steps in the order presented.

To complete this configuration, you will perform the tasks described in the following sections:

- [Add a New Project](#)
- [Add a Proxy Service](#)
- [Configure the Proxy Service](#)

For detailed procedural information, see [Using the Oracle Service Bus Console](#).

Add a New Project

Follow these steps:

1. In the Change Center, click **Create** to create a new session or click **Edit** to enter an existing session.

You must be in a session to edit resources.

2. Select **Project Explorer**.
3. Enter a name for the new project and click **Add Project**.

A message at the top of the page indicates that the project was added successfully.

Add a Proxy Service

Follow these steps:

1. Click the name of the newly created project.
2. From the **Create Resource** drop-down list, select **Proxy Service**.
3. On the **Create a Proxy Service – General Configuration** page, enter the following values:

Service Name – The name of the service

Service Type – Select **Any XML Service** (the default).

Note: Tuxedo Transport only supports **Any XML Service** and **Messaging Service** service types.

4. Click **Next**.
5. On the **Create a Proxy Service – Transport Configuration** page, enter the following required values:

Protocol – Select **tuxedo**.

Endpoint URI - Enter a service name that corresponds to the endpoint URI on the Tuxedo server where the service was deployed.

6. Click **Next**.

7. On the **Tuxedo Transport Configuration** page, enter the following values:

Field Table Classes – Optional. 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 – Optional. Enter the name of the class or classes describing the VIEW/VIEW32 buffer received or sent. These are used for the VIEW/VIEW32-to-XML conversion routines to map field names to element names. This is a space-separated list of fully qualified class names.

Note: X_C_TYPE and X_COMMON Tuxedo buffer types are handled in the same manner as VIEW/VIEW32 buffers.

Note: If an incoming request contains VIEW buffers, then specify the corresponding VIEW classes 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 – From the drop-down list, select a local access point that is associated with the WTC Export service. The drop-down list contains local access points configured in WTC. You cannot create a proxy service without 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.

Remote Access Point – This field appears only when you select **New** in the **Local Access Point** field. From the drop-down 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.

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?** drop-down list. The password will be randomly generated using a temporary variable to avoid security issues.

Note: Access points and users are not deleted by the transport when the endpoints are removed, since they may be used by multiple endpoints. To remove access points, use the WebLogic Server Administration Console.

Reply Buffer Type – From the drop-down list, select the type of buffer that the remote Tuxedo client will receive. This field is enabled if the **Response Required** field is selected.

Reply Buffer Subtype – This field is enabled if the Request Buffer Type value is VIEW or VIEW32. Enter the buffer subtype with which to associate the reply buffer.

Response Required? – Select the check box if this service is expected to send a response. The default status is selected. However, if the service type is **Messaging Service** and the response type is **None** then it is not selected; in this case, the field is not enabled.

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 – The ordering or grouping of elements when FML or FML32 buffers are transformed into XML. Select one of the following choices:

None: (default) The order of fields may not be respected.

Ordered: The fields will be presented with all their occurrences in the correct order.

Ordered and Grouped: If the fields are logically structured as records, the fields will be ordered by occurrence and also grouped by record.

8. Click **Finish**.

9. On the **Summary** page, click **Save**.

Configure the Proxy Service


Oracle Service Bus message flows define the implementation of proxy services. Message flows can include zero or more pipeline pairs: Request and Response Pipelines for the proxy service (or for the operations on the service); and Error Handlers that can be defined for Stages, Pipelines, and proxy services. Pipelines can include one or more Stages, which in turn include actions.


For more information about creating, editing, and viewing message flows, see [Proxy Services: Message Flows](#) in the *Using the Oracle Service Bus Console* and [Modeling Message Flow in Oracle Service Bus](#) in the *Oracle Service Bus User Guide*.



The following sections provide one example of how to change the routing behavior of a proxy service and edit the message flow by:

- Adding a Route Node
- Configuring an action to route the proxy service to a business service

Follow these steps:


1. In the Oracle Service Bus Console, select **Resource Browser > Proxy Services**.
2. Click the  **Edit Message Flow** icon in the row of the proxy service you created in [“Add a Proxy Service”](#) on page 4-2.

The **Edit Message Flow** page displays the default message flow configuration which consists of a single  **Proxy Service** icon that signifies the starting node for the service. This is the minimum configuration of a message flow. The behavior of the message flow is sequential.

3. Click the  **Proxy Service** icon and select **Add Route**.
4. Click the  **Route Node** icon, then click **Edit Name and Description**. Change the name and description, as desired, then click **Save**.

In the message flow, the name of the node changes to display the Route Node name.

5. Click the named  **Route Node** icon and select **Edit Route**.

On the **Edit Stage Configuration** page, the only object displayed is the  **Add an Action** icon.

A Stage is an element of a pipeline and it is a container for actions defined in a pipeline. Actions are the elements of a pipeline stage that define the handling of messages as they flow through a proxy service.

6. Click the  **Add an Action** icon, then select **Add an Action > Communication > Routing**.

The **Edit Stage Configuration** page changes to display the contents of the action. The contents of the action are defined by the type of node you created—a Route Node.

7. Click **Service**.

The Service Browser displays the proxy services and business services that you created.

8. Select the business service that you want to expose to Tuxedo.

9. Click **Submit**.

The display updates to show routing to the business service.

The configuration is completed and ready to test.

Testing Your Configuration

Once you have configured Tuxedo to work with Oracle Service Bus, you can perform a test to verify that it is working correctly. If you are using XML-to-FML32 and FML32-to-XML conversions, you can test this configuration using the `ud32` Tuxedo client program that is included with Tuxedo. (If you are using FML conversions, you can use the `ud` client.) `ud32` reads input consisting of the text representation of FML buffers. For more information on the `ud` and `ud32` commands, see the [Tuxedo Command Reference](#).

If you are not using XML-to-FML and FML-to-XML conversions, you must develop a test client program in Tuxedo to test this configuration. To find information on this task, see the [Oracle Tuxedo documentation](#).

Tuxedo Transport Buffer Transformation

Oracle Service Bus and Tuxedo can interoperate to use the services that each product offers, which can include buffer transformation. The Oracle Service Bus service type and the Tuxedo buffer type determine how transformation occurs.

Tuxedo transport supports **Any XML Service** and **Messaging Service** service types in Oracle Service Bus.

- **Any XML Services (Non SOAP)**—The messages to XML-based services are XML, but can be of any Tuxedo buffer type allowed by the service configuration.
- **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, and untyped binary.

For more information, see “Selecting a Service Type” in [Modeling Message Flow in Oracle Service Bus](#) in the *Oracle Service Bus User Guide*.

The following sections explain how the Tuxedo transport handles buffer transformation.

- [Any XML Service Type](#)
- [Messaging Service Type](#)

Any XML Service Type

[Table 5-1](#) shows the behavior of the Tuxedo transport depending on the Tuxedo buffer type when the service type is **Any XML Service**.

Note: For **Any XML Service**, the payload must be a well-formed XML document.

Table 5-1 Buffer Transformation for Any XML Service

Tuxedo Buffer Type	Tuxedo Transport Behavior
STRING	Passes the buffer as is.
CARRAY X_OCTET	Passes the buffer as is.
FML/FML32	Converts the buffer to and from XML using the configured field classes.
XML	Passes the buffer as is. Note: The Tuxedo application <i>should not</i> send NULL bytes when the Tuxedo buffer is XML.
VIEW/VIEW32 X_C_TYPE X_COMMON	Converts the buffer to and from XML using the configured view classes.
MBSTRING	Passes the buffer as is. Note: Encoding is determined by the <code>encoding</code> element of the <code>MetaData</code> of the message sent or received.

Messaging Service Type

Table 5-2 shows the behavior of the Tuxedo transport depending on the Tuxedo buffer type when the service type is **Messaging Service**.

Note: If **None** is specified in the subtype, the Tuxedo transport should not receive any buffer.

Table 5-2 Buffer Transformation for Messaging Service

Tuxedo Buffer Type	Messaging Type			
	Binary	Text	MFL	XML ¹
STRING	Passed as is	Passed as is	Passed as is ²	XML
CARRAY	Passed as is	Not supported	Passed as is ²	XML
FML/FML32	Passed as is	Not supported	Not supported	XML
XML	Passed as is	Passed as is	Not supported	XML
VIEW/VIEW32	Passed as is	Not supported	Not supported	XML
MBSTRING	Passed as is	Passed as is	Passed as is ²	XML

1. The Tuxedo transport handles the buffer manipulation the same way as if the service was Any XML service type.

2. Assuming the buffer is in suitable format.

For unsupported mappings, the Tuxedo transport returns an error.

Tuxedo Transport Buffer Transformation

Tuxedo Transport Transaction Processing

Oracle Service Bus and Tuxedo can interoperate to use the services that each product offers, which often includes transaction processing. Tuxedo transport takes advantage of transactions or starts transactions in Oracle Service Bus.

Note: The exception to this transaction support is when the inbound transport is Tuxedo with a transactional message and the outbound is request/response XA-JMS. In this case, Oracle Service Bus detects this exception and it results in a TPESYSTEM error.

The Tuxedo transport transactional behavior is driven by the Quality of Service (QoS) setting available at the message context level. For more information, see “Quality of Service” in [Modeling Message Flow in Oracle Service Bus](#) in the *Oracle Service Bus User Guide*.

The following sections explain how the Tuxedo transport handles transactions.

- [Inbound Services](#)
- [Outbound Services](#)

Inbound Services

When a transactional context is received, the message going into the pipeline sets the QoS to **Exactly Once**, otherwise QoS is set to **Best Effort**.

When a `TransportException` is caught before the reply is sent back to the client, the request aborts by throwing a `TPESYSTEM` exception and a transaction rollback results.

Outbound Services

When the thread calling the business service has a transactional context, the Tuxedo transport behaves in the following manner:

- If QoS is set to **Exactly Once**, the Tuxedo transport automatically forwards the transactional context to the remote domain unless the endpoint is configured to suspend the transaction.
- If QoS is set to **Best Effort**, the Tuxedo transport suspends the transaction before making the call and resumes it after the call. This is equivalent to making an ATMI call with `TPNOTRAN` flag set.

When the thread calling the business service has no transaction associated, the Tuxedo call occurs nontransactionally, regardless of the QoS setting. In this case, it will correspond to a `tpcall()` or `tpacall()` with the `TPNOTRAN` flag set.